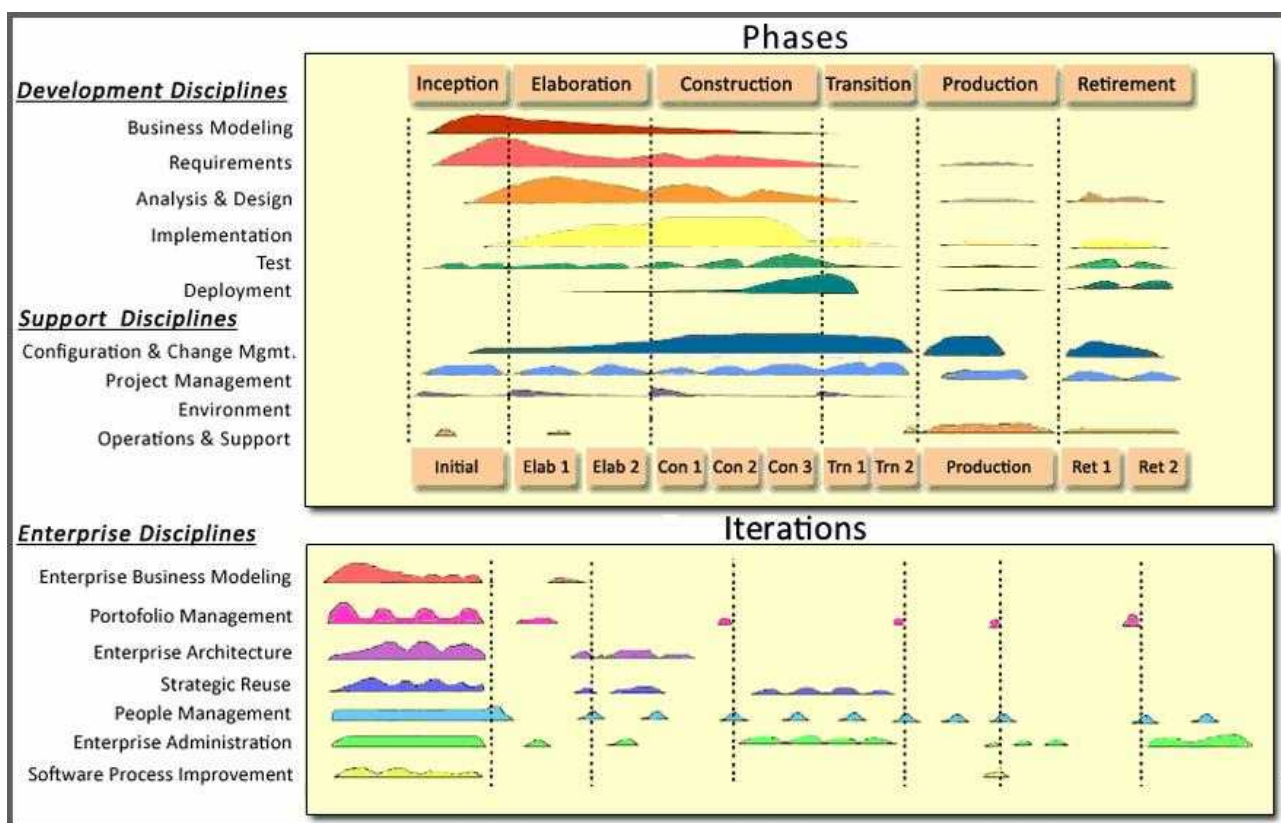




ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

Πτυχιακή εργασία

**ΕΝΟΠΟΙΗΜΕΝΗ ΔΙΕΡΓΑΣΙΑ (UNIFIED PROCESS)  
ΚΑΙ  
ΕΝΟΠΟΙΗΜΕΝΗ ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ (UML)**



**Παυλίδης Μιχάλης**  
A.M 99/1345

Επιβλέπων Καθηγητής: **Δεληγιάννης Ιγνάτιος**

**Θεσσαλονίκη 2007**

## ΠΕΡΙΕΧΟΜΕΝΑ

### ΠΡΟΛΟΓΟΣ

Α ΕΝΟΠΟΙΗΜΕΝΗ ΔΙΕΡΓΑΣΙΑ.....	5
1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	6
1.1 ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΠΟ ΤΙΣ ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ.....	6
1.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΩΣ ΚΕΝΤΡΙΚΟΣ ΑΞΟΝΑΣ.....	7
1.3 ΕΠΑΝΑΛΗΠΤΙΚΗ ΚΑΙ ΜΕ ΣΤΑΔΙΑΚΗ ΑΝΑΠΤΥΞΗ.....	8
2 ΚΥΚΛΟΣ ΖΩΗΣ ΤΗΣ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ.....	10
2.1 ΟΙ ΦΑΣΕΙΣ ΤΗΣ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ.....	14
2.2 Η ΕΠΑΝΑΛΗΨΗ ΓΕΝΙΚΑ.....	17
2.3 Η ΣΥΛΛΗΨΗ ΓΕΝΙΚΑ.....	19
2.4 Η ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΝΙΚΑ.....	21
2.5 Η ΚΑΤΑΣΚΕΥΗ ΓΕΝΙΚΑ.....	21
2.6 Η ΜΕΤΑΒΑΣΗ ΓΕΝΙΚΑ.....	22
3 Η ΓΕΝΙΚΗ ΕΠΑΝΑΛΗΨΗ.....	22
3.1 ΣΧΕΔΙΑΣΜΟΣ ΤΩΝ ΤΕΣΣΑΡΩΝ ΦΑΣΕΩΝ.....	24
3.2 ΣΧΕΔΙΑΣΜΟΣ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ.....	24
3.2.1 ΣΧΕΔΙΑΣΜΟΣ ΤΟΥ ΚΡΙΤΗΡΙΟΥ ΑΠΟΤΙΜΗΣΗΣ.....	25
3.2.2 ΟΙ ΚΙΝΔΥΝΟΙ.....	26
3.2.2.1 ΑΝΤΙΜΕΤΩΠΙΣΗ ΤΩΝ ΚΙΝΔΥΝΩΝ.....	27
3.2.2.2 ΟΙ ΚΙΝΔΥΝΟΙ ΕΠΗΡΕΑΖΟΥΝ ΤΟ ΠΛΑΝΟ ΤΗΣ ΕΠΑΝΑΛΗΨΗΣ.....	27
3.2.2.3 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΝΕΡΓΕΙΩΝ ΚΑΤΑ ΤΩΝ ΚΙΝΔΥΝΩΝ.....	28
3.2.2.4 ΣΕΙΡΑ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ ΤΩΝ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....	28
3.2.2.5 ΤΥΠΟΙ ΚΙΝΔΥΝΩΝ.....	29
3.2.3 ΠΟΡΟΙ ΠΟΥ ΧΡΕΙΑΖΟΝΤΑΙ.....	31
3.2.3.1 ΔΙΑΦΟΡΑ ΕΡΓΑ.....	31
3.2.3.2 ΕΝΑ ΤΥΠΙΚΟ ΕΡΓΟ.....	33
3.2.3.3 ΠΟΛΥΠΛΟΚΑ ΕΡΓΑ ΕΧΟΥΝ ΜΕΓΑΛΥΤΕΡΕΣ ΑΝΑΓΚΕΣ.....	33
3.2.3.4 ΤΟ ΚΑΙΝΟΥΡΓΙΟ ΠΡΟΙΟΝ ΑΠΑΙΤΕΙ ΕΜΠΕΙΡΙΑ.....	34
3.2.4 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΚΑΙ ΤΩΝ ΦΑΣΕΩΝ.....	35
3.2.5 Η ΕΠΟΜΕΝΗ ΕΠΑΝΑΛΗΨΗ.....	35
4 Η ΕΞΕΛΙΞΗ ΤΟΥ ΕΡΓΟΥ.....	36
4.1 ΣΥΛΛΗΨΗ.....	36
4.1.1 Η ΣΥΛΛΗΨΗ ΓΕΝΙΚΑ.....	37
4.1.2 ΠΡΙΝ ΤΗΝ ΕΝΑΡΞΗ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ.....	37
4.1.2.1 ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ.....	38
4.1.2.2 ΕΠΕΚΤΑΣΗ ΤΗΣ ΕΙΚΟΝΑΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	38
4.1.2.3 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ.....	39
4.1.3 ΟΙ ΕΡΓΑΣΙΕΣ ΤΗΣ ΕΠΑΝΑΛΗΨΗΣ ΣΤΗ ΦΑΣΗ ΤΗΣ ΣΥΛΛΗΨΗΣ.....	40
4.1.3.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΠΕΝΤΕ ΒΑΣΙΚΕΣ ΣΕΙΡΕΣ ΕΡΓΑΣΙΩΝ.....	40
4.1.3.2 ΕΥΡΕΣΗ ΚΡΙΣΙΜΩΝ ΚΙΝΔΥΝΩΝ.....	42
4.1.3.3 Η ΕΚΤΕΛΕΣΗ ΤΩΝ ΣΕΙΡΩΝ ΕΡΓΑΣΙΩΝ.....	42
4.1.3.3.1 ΚΑΤΑΓΡΑΦΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ.....	42

4.1.3.3.2 ΑΝΑΛΥΣΗ.....	44
4.1.3.3.3 ΣΧΕΔΙΑΣΗ.....	45
4.1.3.3.4 ΥΛΟΠΟΙΗΣΗ.....	45
4.1.3.3.5 ΔΟΚΙΜΗ.....	46
4.1.3.4 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΠΛΑΝΟΥ.....	46
4.1.3.5 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ.....	46
4.1.4 ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ.....	47
4.1.5 ΤΑ ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ.....	48
4.2 ΕΠΕΞΕΡΓΑΣΙΑ.....	49
4.2.1 Η ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΕΝΙΚΑ.....	49
4.2.2 ΑΡΧΙΚΑ ΣΤΗ ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ.....	50
4.2.3 ΣΧΕΔΙΑΣΗ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ.....	50
4.2.4 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ.....	50
4.2.5 Η ΕΠΑΝΑΛΗΨΗ ΣΤΗ ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ.....	51
4.2.6 ΕΚΤΕΛΕΣΗ ΤΩΝ ΣΕΙΡΩΝ ΕΡΓΑΣΙΩΝ.....	54
4.2.6.1 ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ.....	54
4.2.6.2 ΑΝΑΛΥΣΗ.....	56
4.2.6.3 ΣΧΕΔΙΑΣΗ.....	57
4.2.6.4 ΥΛΟΠΟΙΗΣΗ.....	59
4.2.6.5 ΔΟΚΙΜΗ.....	60
4.2.7 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΣΤΗ ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ.....	60
4.2.8 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΛΑΝΟΥ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	61
4.2.9 ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ.....	62
4.3 ΚΑΤΑΣΚΕΥΗ.....	62
4.3.1 ΑΡΧΙΚΑ ΣΤΗ ΦΑΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	63
4.3.2 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ.....	64
4.3.3 Η ΓΕΝΙΚΗ ΕΠΑΝΑΛΗΨΗ ΣΤΗ ΦΑΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	64
4.3.4 ΕΚΤΕΛΕΣΗ ΤΩΝ ΒΑΣΙΚΩΝ ΣΕΙΡΩΝ ΕΡΓΑΣΙΩΝ.....	66
4.3.4.1 ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ.....	66
4.3.4.2 ΑΝΑΛΥΣΗ.....	67
4.3.4.3 ΣΧΕΔΙΑΣΗ.....	68
4.3.4.4 ΥΛΟΠΟΙΗΣΗ.....	69
4.3.4.5 ΔΟΚΙΜΗ.....	69
4.3.5 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΚΑΙ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	70
4.3.6 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΛΑΝΟΥ ΤΗΣ ΜΕΤΑΒΑΣΗΣ.....	71
4.3.7 ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	71
4.4 ΜΕΤΑΒΑΣΗ.....	71
4.4.1 Η ΦΑΣΗ ΤΗΣ ΜΕΤΑΒΑΣΗΣ ΓΕΝΙΚΑ.....	72
4.4.2 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΛΑΝΟΥ ΤΗΣ ΜΕΤΑΒΑΣΗΣ.....	72
4.4.3 ΚΑΘΟΡΙΣΜΟΣ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ.....	73
4.4.4 ΟΙ ΒΑΣΙΚΕΣ ΣΕΙΡΕΣ ΕΡΓΑΣΙΩΝ ΣΤΗ ΦΑΣΗ ΤΗΣ ΜΕΤΑΒΑΣΗΣ.....	74
4.4.5 ΕΝΕΡΓΕΙΕΣ ΣΤΗ ΦΑΣΗ ΤΗΣ ΜΕΤΑΒΑΣΗΣ.....	76
4.4.6 ΑΠΟΤΙΜΗΣΗ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΜΕΤΑΒΑΣΗΣ.....	77

4.4.7 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΛΑΝΟΥ ΤΗΣ ΕΠΟΜΕΝΗΣ ΕΚΔΟΣΗΣ.....	78
4.4.8 ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΜΕΤΑΒΑΣΗΣ.....	78
<b>Β ΕΝΟΠΟΙΗΜΕΝΗ ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ.....</b>	<b>79</b>
1 ΓΕΝΙΚΑ.....	79
2 ΔΙΑΓΡΑΜΜΑΤΑ.....	80
2.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....	81
2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ.....	85
2.3 ΔΙΑΓΡΑΜΜΑΤΑ ΚΑΤΑΣΤΑΣΗΣ.....	95
2.4 ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ.....	98
2.5 ΔΙΑΓΡΑΜΜΑΤΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ.....	100
2.6 ΔΙΑΓΡΑΜΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ.....	103
<b>Γ ΠΑΡΑΔΕΙΓΜΑ.....</b>	<b>105</b>
1 ΣΥΛΛΗΨΗ.....	105
1.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ.....	105
1.2 ΔΕΥΤΕΡΗ ΕΠΑΝΑΛΗΨΗ.....	106
2 ΕΠΕΞΕΡΓΑΣΙΑ.....	110
2.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ.....	110
2.2 ΔΕΥΤΕΡΗ ΕΠΑΝΑΛΗΨΗ.....	111
2.3 ΤΡΙΤΗ ΕΠΑΝΑΛΗΨΗ.....	112
3 ΚΑΤΑΣΚΕΥΗ.....	113
3.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ.....	114
4 ΜΕΤΑΒΑΣΗ.....	114
4.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ.....	114
<b>Δ ΕΠΙΛΟΓΟΣ.....</b>	<b>115</b>
<b>ΓΛΩΣΣΑΡΙ.....</b>	<b>116</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>117</b>

## ΠΡΟΛΟΓΟΣ

Η τεχνολογία λογισμικού ως γνωστικό αντικείμενο έχει ως στόχο τη βέλτιστη ανάπτυξη συστημάτων λογισμικού και έχει ως βασικό χαρακτηριστικό ότι συνδυάζει αρχές και γνώσεις από άλλα γνωστικά αντικείμενα όπως της επιστήμης των υπολογιστών, της διοίκησης, της διαχείρισης έργων, της ψυχολογίας, και των μαθηματικών. Το εύρος των απαιτούμενων γνώσεων, σε συνδυασμό με την ποικιλία και την πολυπλοκότητα των σημερινών συστημάτων, κάνουν την ανάπτυξη του λογισμικού μια απαιτητική και σε πολλές περιπτώσεις μη προβλέψιμη διαδικασία.

Τα τελευταία χρόνια η κυρίαρχη προσέγγιση για την ανάπτυξη λογισμικού είναι η αντικειμενοστραφής προσέγγιση, σύμφωνα με την οποία η ανάπτυξη των συστημάτων βασίζεται στα αντικείμενα. Μερικά από τα πλεονεκτήματα της αντικειμενοστραφούς ανάπτυξης συστημάτων λογισμικού είναι η απλότητα στην ανάπτυξη, η καλύτερη ποιότητα και αξιοπιστία των παραγόμενων συστημάτων, και τέλος η δυνατότητα επαναχρησιμοποίησης των συστατικών αυτών.

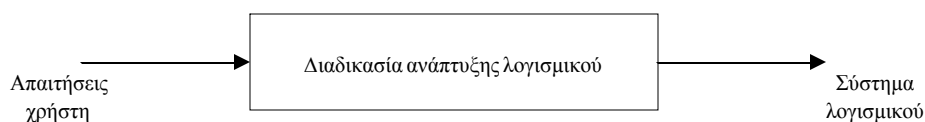
Στα πλαίσια αυτά, η ενοποιημένη διεργασία (unified process) έχει αναδυθεί ως μια δημοφιλής μέθοδος ανάπτυξης αντικειμενοστραφών συστημάτων λογισμικού, και η ενοποιημένη γλώσσα μοντελοποίησης (unified modeling language) ως την πλέον δημοφιλή και πρότυπη γλώσσα για την οπτικοποίηση, προσδιορισμό και τεκμηρίωση συστημάτων λογισμικού. Η ενοποιημένη διεργασία χρησιμοποιεί την ενοποιημένη γλώσσα μοντελοποίησης (UML) όταν ετοιμάζονται τα σχέδια ενός συστήματος λογισμικού. Συγκεκριμένα η UML αποτελεί αναπόσπαστο κομμάτι της ενοποιημένης διεργασίας αφού αναπτύχθηκαν παράλληλα. Σήμερα χιλιάδες εταιρίες χρησιμοποιούν τη μέθοδο της ενοποιημένης διεργασίας και την UML στην ανάπτυξη των συστημάτων τους. Λόγω των παραπάνω, η παρούσα εργασία έχει ως σκοπό να παρουσιάσει τη μέθοδο της ενοποιημένης διεργασίας και την ενοποιημένη γλώσσα μοντελοποίησης.

## Α ΕΝΟΠΟΙΗΜΕΝΗ ΔΙΕΡΓΑΣΙΑ

Σήμερα η τάση στο κόσμο του λογισμικού είναι προς μεγαλύτερα και πιο πολύπλοκα συστήματα. Αυτό οφείλεται εν μέρει στο γεγονός ότι οι υπολογιστές γίνονται πιο ισχυροί κάθε χρόνο με συνέπεια οι χρήστες να περιμένουν περισσότερα από αυτούς. Επίσης αυτή η τάση έχει ενισχυθεί και από την επέκταση της χρήσης του διαδικτύου για ανταλλαγή κάθε είδους πληροφορίας, όπως από απλό κείμενο έως πολυμέσα. Επίσης οι χρήστες συνέχεια ζητούν λογισμικό σύγχρονο που να ικανοποιεί τις όλο αυξανόμενες ανάγκες και απαιτήσεις τους. Αυτό με τη σειρά του οδηγεί σε ακόμα πιο πολύπλοκο λογισμικό. Επιπλέον οι χρήστες το θέλουν και γρήγορα αυτό το λογισμικό. Αυτό καθιστά το χρόνο σημαντικό παράγοντα στην αγορά λογισμικού. Για να επιτευχθούν οι παραπάνω στόχοι δεν είναι κάτι εύκολο. Έτσι όπως εξελίχθηκαν οι απαιτήσεις για δυνατό και πολύπλοκο λογισμικό έτσι έπρεπε να εξελιχθεί και ο τρόπος ή η μέθοδος που αναπτύσσεται το λογισμικό. Όμως αυτό δεν έγινε μέχρι πριν από λίγα χρόνια. Οι περισσότεροι ανέπτυσαν λογισμικό χρησιμοποιώντας μεθόδους που η ηλικία τους ξεπερνούσε τα 25 χρόνια. Αυτό ήταν ένα σημαντικό πρόβλημα. Αν δεν ανανεωνόταν η μέθοδος δεν θα μπορούσαμε να ικανοποιήσουμε τις ολοένα αυξανόμενες ανάγκες για πολύπλοκο λογισμικό. Η κοινότητα λογισμικού χρειάζεται μια μέθοδο εργασίας. Χρειάζεται μια διαδικασία που να ολοκληρώνει όλες τις πλευρές της ανάπτυξης λογισμικού. Χρειάζεται μια συνήθη προσέγγιση, μια διαδικασία δηλαδή που να:

- Παρέχει έναν οδηγό για τη σειρά των δραστηριοτήτων μιας ομάδας.
- Καθοδηγεί τις εργασίες ενός μεμονωμένου μηχανικού λογισμικού και της ομάδας ως σύνολο.
- Να ορίζει τι παραγόμενα θα αναπτυχθούν.
- Να προσφέρει κριτήρια που να ορίζουν το βαθμό προόδου των εργασιών ενός έργου.

Η παρουσία μιας καλά ορισμένης και εύκολα χειριζόμενης μεθόδου είναι το μυστικό της διαφοράς μεταξύ πετυχημένων και αποτυχημένων έργων λογισμικού. Η ενοποιημένη διεργασία (UP) είναι η λύση στα προβλήματα λογισμικού. Η ενοποιημένη διεργασία είναι μια διαδικασία ή μέθοδος ανάπτυξης λογισμικού. Μια διαδικασία ανάπτυξης λογισμικού είναι ένα σύνολο ενεργειών που απαιτούνται για να μετατραπουν οι απαιτήσεις του χρήστη σε ένα σύστημα λογισμικού (εικ. 1).



Εικ. 1 Διαδικασία ανάπτυξης λογισμικού

Παρόλα αυτά η ενοποιημένη διεργασία είναι κάτι παραπάνω από μια απλή διαδικασία. Είναι ένα ευρύτερο πλαίσιο εργασίας που μπορεί να εξειδικευτεί για μια μεγάλη κλάση συστημάτων λογισμικού, για διαφορετικές περιοχές εφαρμογής, για διαφορετικούς τύπους οργανισμών και διαφορετικά μεγέθη έργων. Η ενοποιημένη διεργασία βασίζεται στο συστατικό, το οποίο σημαίνει ότι το σύστημα λογισμικού που αναπτύσσεται αποτελείται από συστατικά λογισμικού που αλληλεπιδρούν μεταξύ τους μέσω των καλά ορισμένων διασυνδέσεων. Η ενοποιημένη διεργασία χρησιμοποιεί την ενοποιημένη γλώσσα μοντελοποίησης (UML) όταν ετοιμάζονται τα σχέδια ενός συστήματος λογισμικού. Συγκεκριμένα η UML αποτελεί αναπόσπαστο κομμάτι της ενοποιημένης διεργασίας αφού αναπτύχθηκαν παράλληλα. Παρόλα αυτά τρία είναι τα χαρακτηριστικά της ενοποιημένης διεργασίας που την διαφοροποιούν από τις υπόλοιπες μεθόδους και την κάνουν μοναδική στο είδος της:

- Καθοδηγούμενη από τις περιπτώσεις χρήσης (use-case driven).
- Η αρχιτεκτονική του συστήματος ως κεντρικός άξονα. (architecture-centric).
- Επαναληπτική και με σταδιακή ανάπτυξη του έργου (iterative and incremental).

## **1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΑΔΙΚΑΣΙΑΣ**

### **1.1 ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΠΟ ΤΙΣ ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ**

Ένα σύστημα λογισμικού δημιουργείται για να εξυπηρετήσει τους χρήστες του. Οπότε για να αναπτυχθεί ένα επιτυχημένο σύστημα πρέπει να ξέρουμε τι θέλουν και τι χρειάζονται οι αναμενόμενοι χρήστες του. Ο όρος «χρήστη» δεν αναφέρεται μόνο σε ανθρώπους άλλα και σε άλλα συστήματα. Υπό αυτήν την έννοια ο όρος χρήστη αναπαριστά κάποιον ή κάτι, όπως άλλα συστήματα, που αλληλεπιδρούν με το υπό κατασκευή σύστημα. Ένα παράδειγμα, είναι το μηχάνημα αυτόματης ανάληψης χρημάτων. Σε αυτήν την περίπτωση ο πελάτης, ως χρήστης του συστήματος, δίνει εντολή ανάληψης χρημάτων στο αυτόματο μηχάνημα. Αυτό με τη σειρά του προβαίνει στις απαραίτητες ενέργειες για να εξυπηρετήσει τον πελάτη. Μια από αυτές τις ενέργειες είναι η επικοινωνία με το κεντρικό σύστημα διαχείρισης των λογαριασμών των πελατών της τράπεζας, το οποίο παίζει το ρόλο του χρήστη στο σύστημα αυτόματης ανάληψης χρημάτων. Η αλληλεπίδραση αυτού του είδους είναι μια περίπτωση χρήσης. Μια περίπτωση χρήσης είναι ένα κομμάτι των λειτουργιών του συστήματος που δίνει στο χρήστη κάποιο αποτέλεσμα που έχει αξία για αυτόν. Οι περιπτώσεις χρήσης καταγράφουν τις λειτουργικές απαιτήσεις του συστήματος. Όλες οι περιπτώσεις χρήσης μαζί συνθέτουν το μοντέλο περιπτώσεων χρήσης το οποίο περιγράφει όλη τη λειτουργικότητα του συστήματος. Η στρατηγική των περιπτώσεων χρήσης μπορεί να χαρακτηριστεί με το ερώτημα «τι πρέπει να κάνει το σύστημα για κάθε χρήστη;». Αυτό το ερώτημα έχει μεγάλη σημασία. Μας αναγκάζει να σκεφτούμε με βάση την αξία που θα έχει κάτι για το χρήστη και όχι με βάση του τι είναι καλό να έχει το σύστημα. Εκτός αυτού, οι περιπτώσεις χρήσης δεν είναι μόνο ένα εργαλείο για τον καθορισμό των απαιτήσεων του συστήματος, αλλά καθορίζουν το σχεδιασμό, την υλοποίηση και τον έλεγχο. Αυτό σημαίνει ότι καθορίζουν τη διαδικασία ανάπτυξης. Βασιζόμενος στο

μοντέλο των περιπτώσεων χρήσης, ο δημιουργός λογισμικού αναπτύσσει μια σειρά σχεδιαστικών μοντέλων και μοντέλων υλοποίησης που ικανοποιούν τις απαιτήσεις χρήσης. Οι κατασκευαστές εξετάζουν το κάθε διαδοχικό μοντέλο ως προς την συμμόρφωση του με το μοντέλο περιπτώσεων χρήσης. Οι δοκιμαστές δοκιμάζουν την υλοποίηση του συστήματος για να επιβεβαιώσουν ότι το μοντέλο υλοποίησης ικανοποιεί τις περιπτώσεις χρήσης. Κατά αυτόν τον τρόπο οι περιπτώσεις χρήσης όχι μόνο ξεκινούν τη διαδικασία ανάπτυξης, αλλά τη συνοδεύουν ως το τέλος. Καθοδηγούμενη από τις περιπτώσεις χρήσης σημαίνει ότι η διαδικασία ανάπτυξης ακολουθεί μια ροή η οποία είναι μια σειρά εργασιών που προέρχονται από τις περιπτώσεις χρήσης. Οι περιπτώσεις χρήσης καθορίζονται, σχεδιάζονται και στο τέλος οι περιπτώσεις χρήσης είναι η πηγή από την οποία οι δοκιμαστές κατασκευάζουν τις δοκιμές χρήσης. Παρόλο που είναι αλήθεια ότι οι περιπτώσεις χρήσης καθοδηγούν τη διαδικασία ανάπτυξης, δεν επιλέγονται όμως μεμονωμένα, αλλά αναπτύσσονται σε συνεργασία με την αρχιτεκτονική του συστήματος. Αυτό σημαίνει ότι οι περιπτώσεις χρήσης καθορίζουν την αρχιτεκτονική του συστήματος και η αρχιτεκτονική του συστήματος επηρεάζει την επιλογή των περιπτώσεων χρήσης. Για αυτό οι περιπτώσεις χρήσης και η αρχιτεκτονική του συστήματος ολοκληρώνονται όσο η διαδικασία ανάπτυξης προχωράει.

## 1.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΩΣ ΚΕΝΤΡΙΚΟΣ ΑΞΟΝΑΣ

Ο ρόλος της αρχιτεκτονικής του συστήματος λογισμικού είναι παρόμοιος με το ρόλο της αρχιτεκτονικής που παίζει σε μία κατασκευή κτηρίου. Το κτήριο εξετάζεται από πολλές απόψεις: δομή, κατανάλωση ρεύματος, ύδρευση κτλ. Αυτό επιτρέπει τον κατασκευαστή να δει μια ολοκληρωμένη εικόνα πριν ξεκινήσει η κατασκευή. Παρόμοια η αρχιτεκτονική σε ένα σύστημα λογισμικού περιγράφεται ως η εικόνα από πολλές πλευρές ενός συστήματος λογισμικού πριν αυτό αρχίσει να κατασκευάζεται. Η σκέψη της αρχιτεκτονικής του συστήματος ενσωματώνει τις πιο σημαντικές στατικές και δυναμικές πλευρές του συστήματος. Η αρχιτεκτονική πηγάζει μέσα από τις ανάγκες τις επιχείρησης και των χρηστών της και οι οποίες αποτυπώθηκαν στις περιπτώσεις χρήσης. Παρόλα αυτά επηρεάζεται και από άλλα στοιχεία όπως από την πλατφόρμα στην οποία θα τρέξει (αρχιτεκτονική υπολογιστών, λειτουργικό σύστημα, σύστημα διαχείρισης βάσης δεδομένων, πρωτόκολλο επικοινωνίας δικτύων), τα επαναχρησιμοποιούμενα κατασκευαστικά κομμάτια που είναι διαθέσιμα, τις μελέτες υλοποίησης και από τις μη λειτουργικές απαιτήσεις (απόδοση, αξιοπιστία). Η αρχιτεκτονική είναι μια όψη του όλου σχεδίου με προβαλλόμενα τα σημαντικά χαρακτηριστικά μέσω της αφαίρεσης των λεπτομερειών. Αφού ο βαθμός της σημαντικότητας εξαρτάται από την κρίση του καθενός, το οποίο με την σειρά του εξαρτάται από την εμπειρία του καθενός, η αξία της αρχιτεκτονικής εξαρτάται από τα άτομα που έχουν επωμιστεί το έργο αυτό. Παρόλα αυτά η ενοποιημένη διαδικασία βοηθά αυτόν που αναπτύσσει λογισμικό να επικεντρωθεί στους σωστούς στόχους, όπως η κατανόηση του συστήματος, προσαρμοστικότητα σε μελλοντικές αλλαγές και επαναχρησιμοποίηση. Τώρα θα επικεντρωθούμε στο πως σχετίζονται οι περιπτώσεις χρήσης και η αρχιτεκτονική. Κάθε προϊόν έχει μια λειτουργία και μια μορφή. Μόνο το ένα ή μόνο το άλλο δεν είναι αρκετό. Αυτά τα δύο πρέπει να συνεργάζονται για δημιουργήσουν ένα επιτυχημένο προϊόν. Στο χώρο του λογισμικού η λειτουργία αντικατοπτρίζεται στις περιπτώσεις χρήσης και η μορφή στην αρχιτεκτονική.



Οπότε χρειάζεται να υπάρχει συνεργασία μεταξύ των περιπτώσεων χρήσης και της αρχιτεκτονικής. Είναι ένα πρόβλημα της κότας και του αυγού. Από τη μια πλευρά πρέπει οι περιπτώσεις χρήσης όταν υλοποιούνται να ταιριάζουν στην αρχιτεκτονική του συστήματος, ενώ από την άλλη πλευρά η αρχιτεκτονική πρέπει να αφήνει χώρο για την υλοποίηση όλων των περιπτώσεων χρήσης τόσο στη παρούσα φάση αλλά τόσο και στο μέλλον. Στην πραγματικότητα οι περιπτώσεις χρήσης και η αρχιτεκτονική πρέπει να αναπτύσσονται παράλληλα. Για αυτό οι κατασκευαστές λογισμικού παράγουν το σύστημα με μια μορφή. Είναι η μορφή, δηλαδή η αρχιτεκτονική, που θα πρέπει να σχεδιαστεί έτσι ώστε να επιτρέπει στο σύστημα να εξελίσσεται, όχι μόνο κατά την αρχική του ανάπτυξη αλλά και στο μέλλον. Για να βρεθεί αυτή η μορφή οι κατασκευαστές λογισμικού πρέπει να εργαστούν για την κατανόηση των κύριων λειτουργιών του συστήματος, δηλαδή των κύριων περιπτώσεων χρήσης. Αυτές οι κύριες περιπτώσεις χρήσης μπορεί να ανέρχονται από το 5% έως το 10% όλων των περιπτώσεων χρήσης, άλλα είναι οι σημαντικές, είναι αυτές που απαρτίζουν τον κύριο πυρήνα των λειτουργιών του συστήματος. Με απλοποιημένους όρους αυτός που αναπτύσσει λογισμικό:

- Δημιουργεί ένα γενικό περίγραμμα της αρχιτεκτονικής, ξεκινώντας από το μέρος της αρχιτεκτονικής που δεν σχετίζεται με τις περιπτώσεις χρήσης (π.χ. πλατφόρμα). Αλλά και παρόλο που αυτό το κομμάτι της αρχιτεκτονικής δεν σχετίζεται με τις περιπτώσεις χρήσης, ο κατασκευαστής πρέπει να έχει μια γενική κατανόηση των περιπτώσεων χρήσης πριν τη δημιουργία του γενικού περιγράμματος.
- Μετά ο κατασκευαστής δουλεύει με ένα υποσύνολο των καταγεγραμμένων περιπτώσεων χρήσης, αυτές που αναπαριστούν τις κύριες λειτουργίες του συστήματος υπό ανάπτυξη. Κάθε μία από τις επιλεγμένες περιπτώσεις χρήσης καθορίζεται λεπτομερώς και υλοποιούνται τα υποσυστήματα, οι κλάσεις και τα συστατικά.
- Καθώς οι περιπτώσεις χρήσης καθορίζονται και εξελίσσονται, όλο και περισσότερο μέρος από την αρχιτεκτονική του συστήματος προσδιορίζεται. Αυτό με τη σειρά του οδηγεί στην εξέλιξη όλο και περισσότερων περιπτώσεων χρήσης.

Η διαδικασία αυτή συνεχίζεται μέχρι η αρχιτεκτονική του συστήματος να κριθεί σταθερή.

### **1.3 ΕΠΑΝΑΛΗΠΤΙΚΗ ΚΑΙ ΜΕ ΣΤΑΔΙΑΚΗ ΑΝΑΠΤΥΞΗ**

Η ανάπτυξη ενός συστήματος λογισμικού προς εμπορική χρήση είναι ένα εγχείρημα που μπορεί να διαρκέσει από αρκετούς μήνες έως πιθανώς και χρόνια. Είναι πρακτικό να χωριστεί η εργασία του έργου σε μικρότερα έργα. Κάθε μικρό έργο είναι μια επανάληψη που συντελεί στην ανάπτυξη του έργου. Οι επαναλήψεις δηλαδή είναι βήματα που γίνονται στην εργασία και αυξάνουν σταδιακά το μέγεθος του έργου. Για να είναι πιο αποτελεσματικές οι επαναλήψεις πρέπει να είναι ελεγχόμενες, δηλαδή να επιλέγονται προσεκτικά και να πραγματοποιούνται με ένα προκαθορισμένο τρόπο. Για αυτό το λόγο ονομάζονται και μίνι έργα. Οι κατασκευαστές για να επιλέξουν το τι θα υλοποιηθεί σε κάθε επανάληψη βασίζονται σε δύο παράγοντες. Πρώτα από όλα κάθε επανάληψη έχει να κάνει με ένα σύνολο από περιπτώσεις χρήσης που επεκτείνουν τη χρησιμότητα του

συστήματος που έχει κατασκευαστεί ως τώρα. Και κατά δεύτερον κάθε επανάληψη έχει να κάνει με τα πιο σημαντικά ρίσκα. Οι διαδοχικές επαναλήψεις συνεχίζουν από το τέλος των σχεδίων που έχουν αναπτυχθεί κατά τις προηγούμενες επαναλήψεις. Μια επανάληψη λοιπόν είναι ένα μίνι έργο, οπότε από τις περιπτώσεις χρήσης συνεχίζουμε με την ανάλυση, σχεδίαση, υλοποίηση και δοκιμή, η οποία η δοκιμή πραγματοποιείται με την μορφή εκτελέσιμου κώδικα ο οποίος ικανοποιεί τις περιπτώσεις χρήσης που αναπτύχθηκαν σε αυτή την επανάληψη. Φυσικά κάθε επανάληψη δεν μεταβάλλει το συνολικό έργο μόνο ποσοτικά άλλα και ποιοτικά. Ειδικά στην αρχή της διαδικασίας οι κατασκευαστές μπορεί να αντικαθιστούν τη πρώτη πρόχειρη σχεδίαση με μια πιο μελετημένη και λεπτομερή σχεδίαση. Σε επόμενες επαναλήψεις η σταδιακή αύξηση του συστήματος είναι περισσότερο ποσοτική. Σε κάθε επανάληψη της διαδικασίας οι κατασκευαστές του συστήματος καθορίζουν τις χρήσιμες περιπτώσεις χρήσης, δημιουργούν μια σχεδίαση χρησιμοποιώντας την επιλεγμένη αρχιτεκτονική σαν οδηγό, υλοποιούν την σχεδίαση σε συστατικά, και επιβεβαιώνουν ότι τα συστατικά ικανοποιούν τις περιπτώσεις χρήσης. Όταν η επανάληψη πετύχει τους στόχους της η διαδικασία προχωράει στην επόμενη επανάληψη. Όταν όμως δεν πετυχαίνει τους στόχους της τότε οι κατασκευαστές πρέπει να επαναπροσεγγίσουν τις αποφάσεις που έχουν παρθεί. Για να επιτευχθεί η μεγαλύτερη οικονομία στην ανάπτυξη του συστήματος, η ομάδα που το αναπτύσσει θα προσπαθήσει να επιλέξει μόνο τις επαναλήψεις που απαιτούνται για να επιτευχθεί ο στόχος του συστήματος. Θα προσπαθήσει να τοποθετήσει τις επαναλήψεις σε μια λογική σειρά. Ένα πετυχημένο έργο θα ακολουθήσει την αρχική διαδρομή που έχει αρχικά χαραχθεί με μικρές μόνο αποκλίσεις. Φυσικά η όλη διαδικασία μπορεί να πάρει περισσότερο χρόνο και προσπάθεια αφού αναπάντεχα προβλήματα θα προσθέσουν επιπλέον επαναλήψεις ή θα αλλάξουν την σειρά των επαναλήψεων. Η ελαχιστοποίηση των αναπάντεχων προβλημάτων είναι ένας στόχος της ελαχιστοποίησης των ρίσκων. Είναι πολλά τα πλεονεκτήματα μιας ελεγχόμενης επαναληπτικής διαδικασίας:

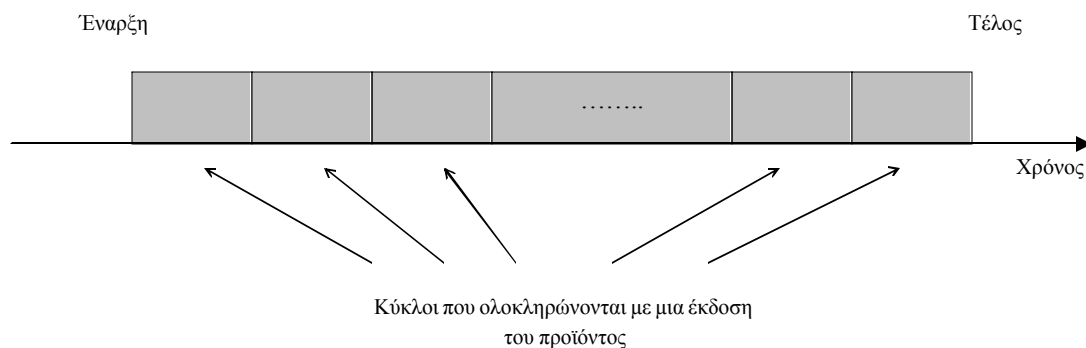
- Η ελεγχόμενη επανάληψη περιορίζει το ύψος τους κόστους σε περίπτωση αποτυχίας μόνο στα έξοδα μίας μόνο επανάληψης. Αν οι κατασκευαστές χρειάζεται να επαναλάβουν την επανάληψη, τότε η επιχείρηση χάνει μόνο την αξία της λανθασμένης προσπάθειας διεκπεραίωσης μίας μόνο επανάληψης και όχι την αξία της προσπάθειας διεκπεραίωσης ολόκληρου του έργου.
- Η ελεγχόμενη επανάληψη μειώνει τον κίνδυνο να μην προωθηθεί το προϊόν στην αγορά στην προγραμματισμένη ημερομηνία. Απομονώνοντας τους κινδύνους στην αρχή της ανάπτυξης, ο χρόνος που χρειάζεται για να αντιμετωπισθούν καταναλώνεται στην αρχή του προγράμματος όταν οι κατασκευαστές είναι λιγότερο πιεσμένοι χρονικά από ότι στο τέλος του προγράμματος. Στην παραδοσιακή προσέγγιση της ανάπτυξης λογισμικού όπου τα δύσκολα προβλήματα παρουσιάζονται κατά την φάση της δοκιμής, ο χρόνος που χρειάζεται για να αντιμετωπιστούν, συνήθως, ξεπερνά το εναπομείναντα χρόνο του προγράμματος και σχεδόν συνέχεια οδηγεί σε καθυστέρηση της παράδοσης του έργου.
- Η ελεγχόμενη επανάληψη επιταχύνει τον ρυθμό προσπάθειας ανάπτυξης του συστήματος γιατί οι κατασκευαστές δουλεύουν πιο παραγωγικά προς στόχους που είναι βραχυπρόθεσμοι και ξεκάθαροι παρά προς στόχους μακροπρόθεσμους και γενικούς.

- Η ελεγχόμενη επανάληψη παραδέχεται μια πραγματικότητα που συχνά αγνοείται. Ότι οι ανάγκες των χρηστών και οι αντίστοιχες απαιτήσεις δεν μπορούν να καθοριστούν πλήρως από την αρχή. Προσδιορίζονται και βελτιώνονται βαθμιαία κατά τις διαδοχικές επαναλήψεις. Αυτό το είδος προσέγγισης καθιστά ευκολότερη τη προσαρμογή στις μεταβαλλόμενες απαιτήσεις.

Αυτές οι αρχές, καθοδήγηση από τις περιπτώσεις χρήσης, κεντρικός άξονας η αρχιτεκτονική, και η επανάληψη με σταδιακή ανάπτυξη είναι εξίσου σημαντικές. Η αρχιτεκτονική παρέχει την δομή μέσω της οποίας η εργασία θα καθοδηγηθεί στις επαναλήψεις, ενώ οι περιπτώσεις χρήσης καθορίζουν τους στόχους και καθοδηγούν την εργασία σε κάθε μια επανάληψη. Παραλείποντας μία από αυτές τις αρχές θα έχει ως αποτέλεσμα την δραματική μείωση της αξίας της ενοποιημένης διεργασίας. Είναι σαν ένα τραπέζι με τρία πόδια, στο οποίο αφαιρώντας το ένα πόδι το τραπέζι πέφτει. Αφού έχουμε αναφερθεί στις τρεις αρχές της ενοποιημένης διεργασίας θα δούμε την όλη διαδικασία, τον κύκλο ζωής της, τις εργασίες, τις φάσεις, και τις επαναλήψεις.

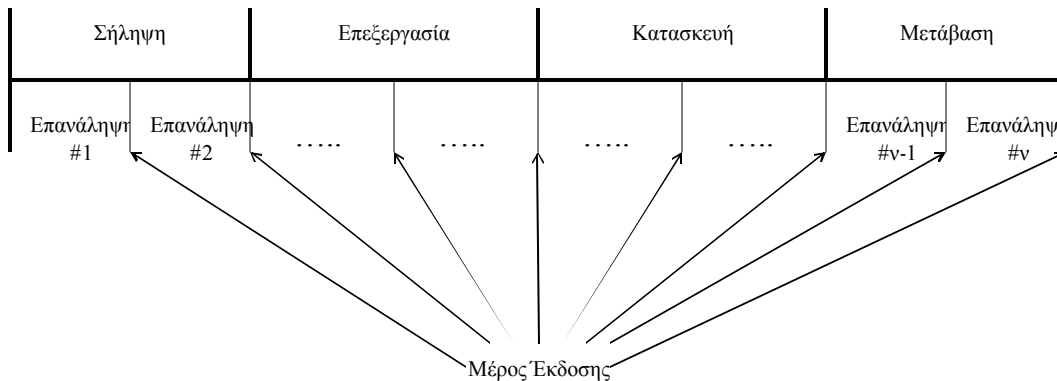
## 2 ΚΥΚΛΟΣ ΖΩΗΣ ΤΗΣ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ

Η ενοποιημένη διαδικασία επαναλαμβάνει μια σειρά κύκλων εργασιών που σταδιακά ολοκληρώνουν το σύστημα. Κάθε κύκλος εργασιών ολοκληρώνεται με την έκδοση ενός προϊόντος προς τους πελάτες (εικ. 2).



Εικ. 2 Κύκλοι εργασιών συστήματος λογισμικού

Κάθε κύκλος περιλαμβάνει τέσσερις φάσεις: σύλληψη (inception), επεξεργασία (elaboration), κατασκευή (construction) και Μετάβαση (transition). Κάθε φάση χωρίζεται σε επαναλήψεις όπως αναφέρθηκαν πριν (εικ. 3).



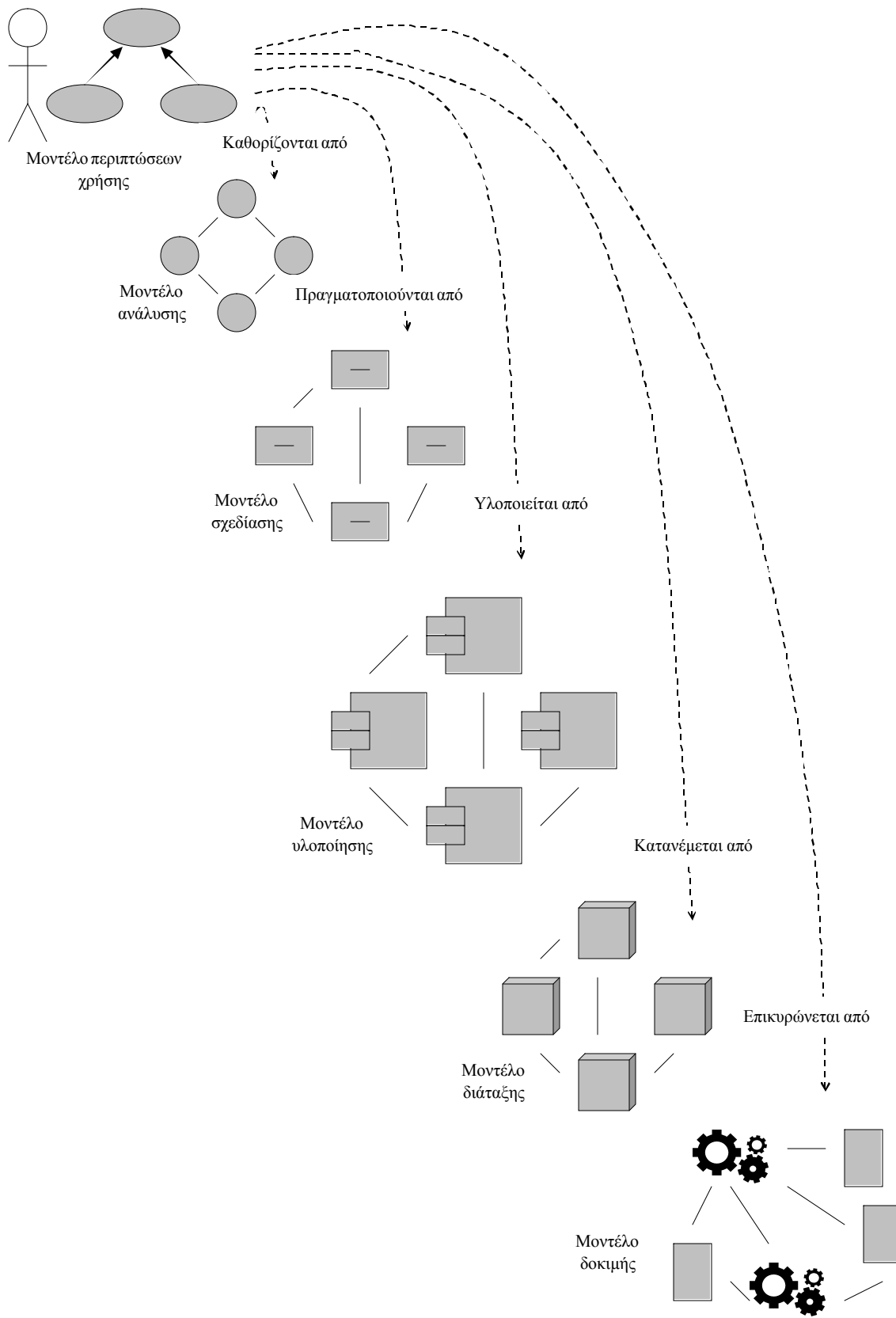
Εικ. 3 Διαδικασία ανάπτυξης λογισμικού

Κάθε κύκλος ολοκληρώνεται με μια νέα έκδοση του συστήματος και κάθε έκδοση είναι ένα προϊόν έτοιμο για παράδοση. Αποτελείται από το σώμα του πηγαίου κώδικα ενσωματωμένο σε συστατικά τα οποία μπορούν να μεταγλωττιστούν και να εκτελεστούν και μαζί τα εγχειρίδια χρήσης. Εκτός από αυτά το τελικό προϊόν πρέπει να εξυπηρετεί τις ανάγκες όχι μόνο των χρηστών αλλά και των επενδυτών και όλων όσων σχετίζονται με το προϊόν. Το προϊόν λογισμικού πρέπει να αποτελεί κάτι παραπάνω από κώδικα μηχανής ο οποίος εκτελείται. Το τελικό προϊόν περιλαμβάνει τις απαιτήσεις, τις περιπτώσεις χρήσης, τις μη λειτουργικές απαιτήσεις και τις περιπτώσεις δοκιμής. Περιλαμβάνει την αρχιτεκτονική και τα οπτικά μοντέλα, δηλαδή τα σχέδια που δημιουργήθηκαν με την χρήση της ενοποιημένης γλώσσας μοντελοποίησης (UML). Συγκεκριμένα περιλαμβάνει όλα τα στοιχεία που αναφέρθηκαν γιατί είναι αυτά τα οποία θα δώσουν τη δυνατότητα σε όλους τους εμπλεκόμενους όπως πελάτες, χρήστες, αναλυτές, σχεδιαστές, προγραμματιστές και αυτούς που θα κάνουν τις δοκιμές, να καθορίσουν, να αναλύσουν, να σχεδιάσουν, να υλοποιήσουν και να δοκιμάσουν το σύστημα αντίστοιχα. Επιπλέον, είναι αυτά τα στοιχεία που θα επιτρέψουν τους εμπλεκόμενους να χρησιμοποιήσουν και να εξελίξουν το σύστημα από γενιά σε γενιά. Παρόλο που τα εκτελέσιμα συστατικά είναι τα πιο σημαντικά από την πλευρά των χρηστών, μόνα τους όμως δεν είναι αρκετά. Αυτό συμβαίνει γιατί το περιβάλλον μεταλλάσσεται. Λειτουργικά συστήματα και συστήματα βάσεων δεδομένων βελτιώνονται. Καθώς η ανάπτυξη ενός συστήματος προχωράει και το πεδίο γίνεται πιο κατανοητό, οι απαιτήσεις μπορεί να μεταβληθούν. Συγκεκριμένα, είναι κάτι δεδομένο στην ανάπτυξη λογισμικού ότι οι απαιτήσεις μεταβάλλονται. Κάποια στιγμή οι κατασκευαστές θα πρέπει να ξεκινήσουν ένα καινούργιο κύκλο εργασιών και οι

επενδυτές πρέπει να το χρηματοδοτήσουν. Για να διεκπεραιώσουν πετυχημένα τον επόμενο κύκλο οι κατασκευαστές χρειάζονται όλες τις απεικονίσεις του προϊόντος λογισμικού που είναι οι εξής (εικ. 4):

- Ένα μοντέλο περιπτώσεων χρήσης με όλες τις περιπτώσεις χρήσης και τις συσχετίσεις τους με τους χρήστες.
- Ένα μοντέλο ανάλυσης που έχει δύο σκοπούς: να βελτιώσει τις περιπτώσεις χρήσης με περισσότερες λεπτομέρειες και να κάνει μία αρχική διανομή της συμπεριφοράς του συστήματος σε ένα σύνολο αντικειμένων που θα παρέχουν αυτήν τη συμπεριφορά.
- Ένα μοντέλο σχεδίασης που θα καθορίζει τη στατική δομή του συστήματος με υποσυστήματα, κλάσεις και διασυνδέσεις, και τις περιπτώσεις χρήσης ικανοποιημένες από τις συνεργασίες μεταξύ των υποσυστημάτων, των κλάσεων και των διασυνδέσεων.
- Ένα μοντέλο υλοποίησης το οποίο θα περιλαμβάνει τα συστατικά και την κατανομή των κλάσεων στα συστατικά.
- Ένα μοντέλο διάταξης που θα καθορίζει τους φυσικούς κόμβους υπολογιστών και την κατανομή των συστατικών στους κόμβους αυτούς.
- Ένα μοντέλο δοκιμής που θα περιγράφει τις περιπτώσεις δοκιμής που θα επιβεβαιώνουν την ικανοποίηση των περιπτώσεων χρήσης.
- Την αναπαράσταση της αρχιτεκτονικής.

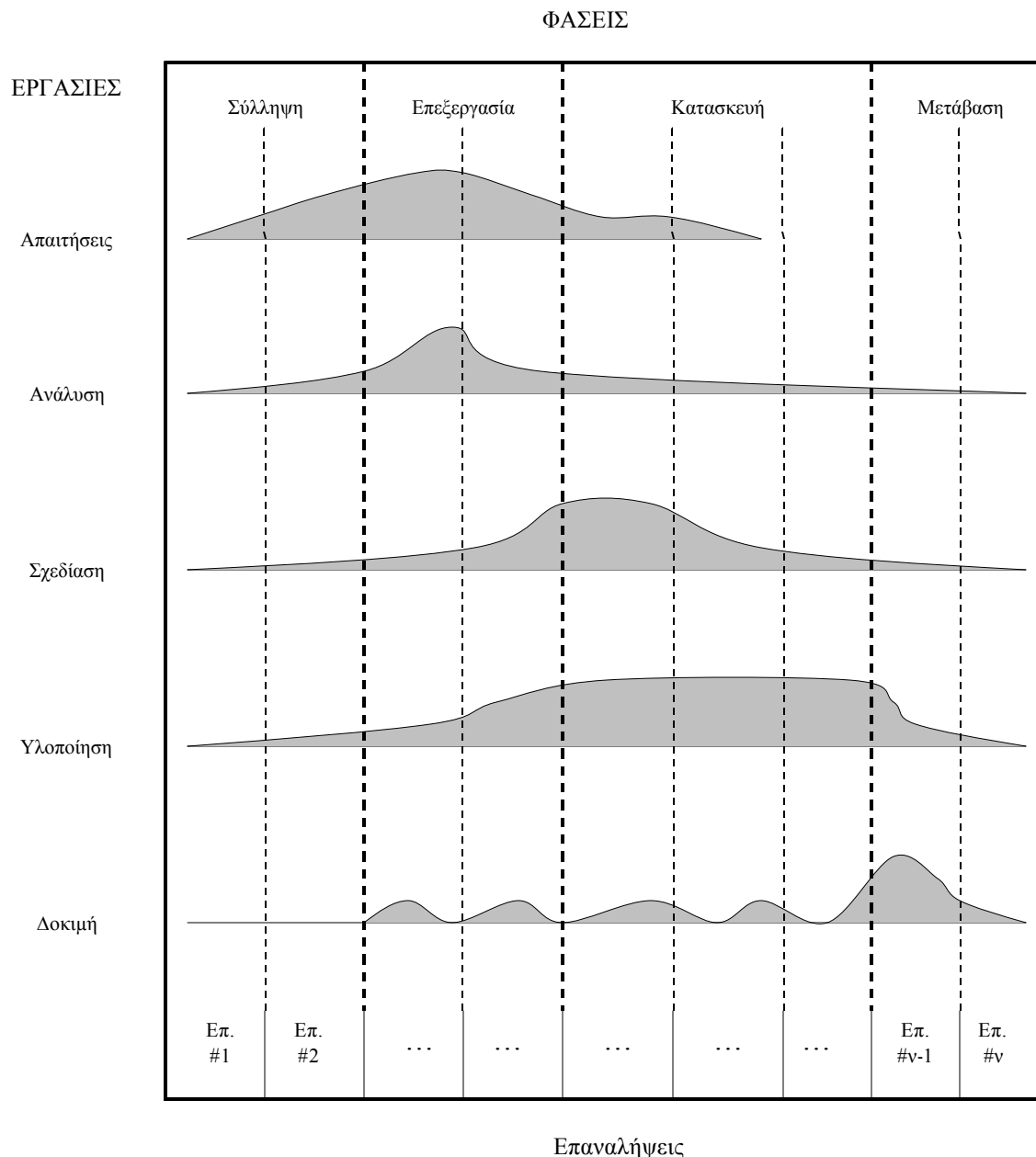
Το σύστημα μπορεί επιπλέον να έχει ένα μοντέλο πεδίου που θα περιγράφει το επιχειρηματικό περιβάλλον του συστήματος. Όλα αυτά τα μοντέλα συσχετίζονται. Μαζί αναπαριστούν το σύστημα ως σύνολο. Τα στοιχεία σε ένα μοντέλο έχουν ίχνη τα οποία φανερώνουν την εξέλιξη του στοιχείου στα επόμενα ή προηγούμενα μοντέλα. Για παράδειγμα, για μια περίπτωση χρήσης στο μοντέλο περιπτώσεων χρήσης μπορούμε να βρούμε την πραγματοποίηση της περίπτωσης χρήσης στο μοντέλο σχεδίασης και στη συνέχεια τη δοκιμή της περίπτωσης χρήσης στο μοντέλο δοκιμής. Τα ίχνη βοηθούν στην κατανόηση του συστήματος και στην περίπτωση που χρειάζεται να προβούμε σε αλλαγές.



Εικ. 4 Μοντέλα αναπαράστασης λογισμικού

## 2.1 ΟΙ ΦΑΣΕΙΣ ΤΗΣ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ

Κάθε κύκλος εργασιών κρατάει μια χρονική περίοδο. Αυτή η χρονική περίοδος χωρίζεται με τη σειρά της σε τέσσερις φάσεις όπως είπαμε πριν και φαίνεται και στην εικόνα 5. Μέσα από μια σειρά σχεδιαστικών μοντέλων οι εμπλεκόμενοι στην ανάπτυξη του συστήματος σχηματίζουν μια εικόνα για το τι γίνεται σε κάθε φάση. Μέσα σε κάθε φάση οι κατασκευαστές μπορεί να χωρίσουν την εργασία ακόμα περισσότερο σε επαναλήψεις. Κάθε φάση ολοκληρώνεται σε κάποια σημεία ορόσημο για την ανάπτυξη του συστήματος. Τα σημεία ορόσημο καθορίζονται από συγκεκριμένα μοντέλα του συστήματος που έχουν αναπτυχθεί σε ικανοποιητικό βαθμό. Τα σημεία ορόσημο εξυπηρετούν πολλούς σκοπούς. Το πιο σημαντικό είναι ότι οι επικεφαλής του έργου πρέπει να πάρουν κάποιες σημαντικές αποφάσεις πριν ξεκινήσει η επόμενη φάση. Τα σημεία επίσης επιτρέπουν την καλύτερη επίβλεψη του έργου όπως και τους κατασκευαστές να βλέπουν την πρόοδο που σημειώνεται καθώς το έργο φτάνει στα τέσσερα σημαντικά αυτά σημεία. Τέλος μελετώντας και καταγράφοντας το χρόνο και την προσπάθεια που καταβάλλεται σε κάθε φάση δημιουργούμε ένα σύνολο πληροφοριών. Αυτές οι πληροφορίες θα είναι χρήσιμες στην ανάπτυξη άλλων έργων όπου θα κληθούμε να απαντήσουμε σε ερωτήματα για το πόσο χρόνο και ανθρώπινο δυναμικό θα χρειαστούμε για την διεκπεραίωση του έργου. Η εικόνα 5 δείχνει στην αριστερή στήλη την σειρά των εργασιών, δηλαδή καταγραφή απαιτήσεων, ανάλυση, σχεδίαση, υλοποίηση και δοκιμή. Οι καμπύλες παρουσιάζουν σε τι βαθμό κάθε εργασία πραγματοποιείται σε κάθε φάση. Η κάθε φάση χωρίζεται σε επαναλήψεις. Μια τυπική επανάληψη στην φάση της επεξεργασίας περιέχει όλες τις εργασίες.



Εικ. 5 Φάσεις της ενοποιημένης διεργασίας

Κατά τη φάση της σύλληψης, για μια καλή ιδέα που μπορεί να υπάρχει, σχηματίζεται η εικόνα του τελικού προϊόντος έως το τέλος της φάσης. Συγκεκριμένα σε αυτήν τη φάση απαντώνται τα εξής ερωτήματα:

- Ποια είναι η βασική λειτουργία του συστήματος για τους βασικούς του χρήστες;
- Για αυτό το σύστημα πώς θα μπορούσε να είναι η αρχιτεκτονική του;
- Ποιο είναι το πλάνο και ποιο το κόστος για την ανάπτυξη του προϊόντος λογισμικού;



Ένα απλοποιημένο μοντέλο περιπτώσεων χρήσης που περιέχει τις πιο βασικές περιπτώσεις χρήσης θα απαντήσει στην πρώτη ερώτηση. Σε αυτό το σημείο η αρχιτεκτονική είναι αβέβαιη. Είναι απλά ένα περίγραμμα που περιέχει τα πιο κρίσιμα υποσυστήματα. Σε αυτήν την φάση προσδιορίζονται τα πιο σημαντικά ρίσκα και μπαίνουν σε μια σειρά, σχεδιάζεται αναλυτικά η επόμενη φάση, η φάση της επεξεργασίας, και γίνεται μια εκτίμηση του όλου έργου.

Κατά τη φάση της επεξεργασίας οι περισσότερες από τις περιπτώσεις χρήσης του συστήματος καθορίζονται με λεπτομέρειες και σχεδιάζεται η αρχιτεκτονική του συστήματος. Κατά τη φάση αυτή ανακαλύπτονται οι πιο κρίσιμες περιπτώσεις χρήσης και πραγματοποιούνται. Το αποτέλεσμα αυτής της φάσης είναι ότι μπαίνει η βάση της αρχιτεκτονικής του συστήματος. Κατά το τέλος της φάσης ο επικεφαλής του έργου είναι σε θέση να σχεδιάσει τις ενέργειες και να κάνει μια εκτίμηση για τους πόρους που χρειάζονται για την ολοκλήρωση του έργου. Εδώ η κύρια ερώτηση είναι η εξής: είναι οι περιπτώσεις χρήσης, η αρχιτεκτονική και τα σχέδια σταθερά και είναι οι κίνδυνοι υπό έλεγχο ώστε να είναι σε θέση να γίνει δέσμευση του όλου έργου με συμβόλαιο;

Κατά την φάση της κατασκευής το προϊόν χτίζεται. Σε αυτήν τη φάση η κεντρική αρχιτεκτονική του συστήματος επεκτείνεται. Η εικόνα του τελικού προϊόντος αποκτά σάρκα και οστά και είναι έτοιμο να παραδοθεί στους τελικούς χρήστες. Σε αυτή τη φάση της ανάπτυξης του συστήματος καταναλώνεται το μεγαλύτερο μέρος των πόρων. Η αρχιτεκτονική του συστήματος είναι σταθερή, αλλά παρόλα αυτά επειδή μπορεί να βρεθεί κάποιος καλύτερος τρόπος κατασκευής του συστήματος, μπορεί να γίνουν κάποιες αλλαγές, αλλά πάντοτε μικρού μεγέθους. Στο τέλος αυτής της φάσης το προϊόν περιέχει όλες τις περιπτώσεις χρήσης που συμφωνήθηκαν να αναπτυχθούν στην αρχή του έργου από τους κατασκευαστές και τους πελάτες. Το τελικό προϊόν αυτής της φάσης όμως δεν θα είναι χωρίς ατέλειες. Οι περισσότερες ατέλειες θα ανακαλυφθούν και θα διορθωθούν στη φάση του ελέγχου. Η ερώτηση που τίθεται στο σημείο ορόσημο είναι η εξής: καλύπτει επαρκώς το σύστημα τις ανάγκες των χρηστών ώστε να παραδοθεί νωρίτερα σε κάποιους πελάτες;

Η φάση της μετάβασης καλύπτει μια περίοδο κατά την οποία το προϊόν βρίσκεται σε μια κατάσταση δοκιμαστικής έκδοσης. Η δοκιμαστική αυτή έκδοση δοκιμάζεται από ένα μικρό αριθμό ειδικευμένων και έμπειρων χρηστών που αναφέρουν τυχόν ατέλειες και ελλείψεις. Οι κατασκευαστές μετά διορθώνουν τα αναφερθέντα προβλήματα και ενσωματώνουν τις βελτιώσεις σε μια γενική έκδοση για όλους τους χρήστες. Η φάση της μετάβασης περιλαμβάνει εργασίες όπως η εκπαίδευση του προσωπικού, που θα χειριστεί το σύστημα και διόρθωση προβλημάτων μετά την παράδοση του συστήματος. Η ομάδα συντήρησης συχνά χωρίζεται σε δύο κατηγορίες. Σε αυτά με σοβαρή επίπτωση στις λειτουργίες του συστήματος που δικαιολογεί μια έκτακτη έκδοση του συστήματος και σε αυτά που μπορούν να διορθωθούν στην επόμενη προγραμματισμένη έκδοση του συστήματος.

Η ενοποιημένη διεργασία βασίζεται στα συστατικά. Χρησιμοποιεί τη γλώσσα οπτικοποίησης UML και στηρίζεται σε τρεις αρχές. Στην καθοδήγηση από τις περιπτώσεις χρήσης, στην αρχιτεκτονική ως κεντρικό άξονα και στην επανάληψη με

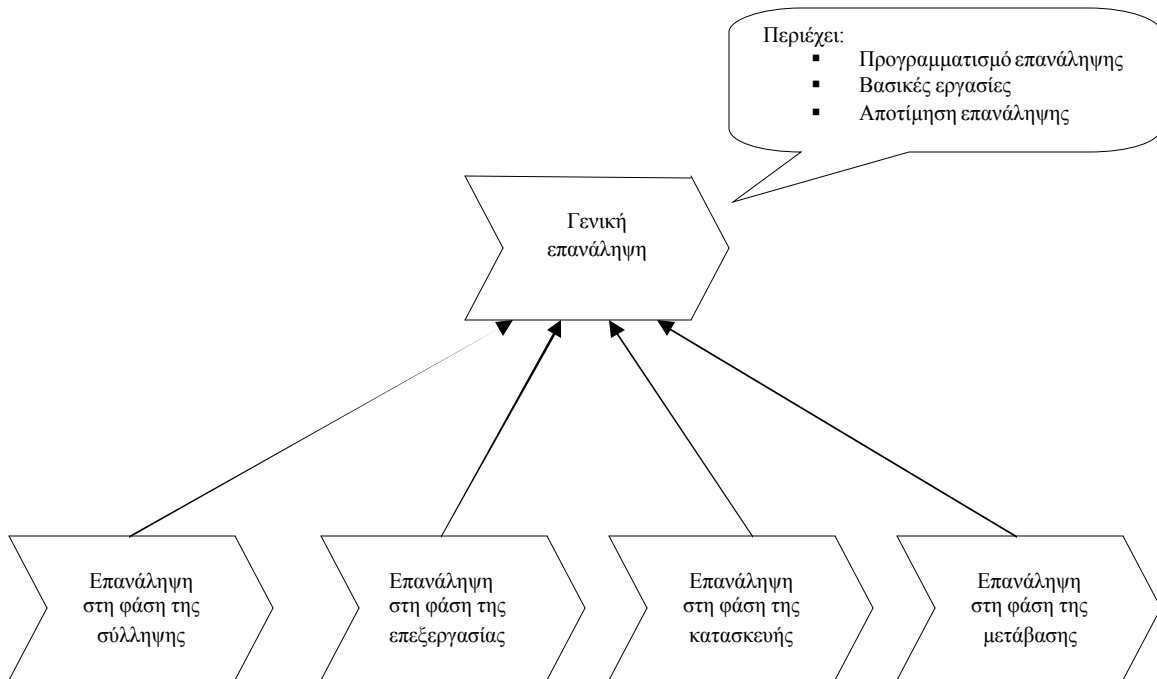
σταδιακή ανάπτυξη. Για να λειτουργήσουν και να έχουν αποτέλεσμα αυτές οι αρχές απαιτείται μια πολύπλευρη διαδικασία, μία διαδικασία που να λαμβάνει υπόψη τις φάσεις, τις σειρές εργασιών, την ελαχιστοποίηση των κινδύνων, τον έλεγχο της ποιότητας, την επίβλεψη του έργου και τον έλεγχο των μεταβολών. Η ενοποιημένη διεργασία έχει αναπτυχθεί ως πλαίσιο εργασίας που ενσωματώνει όλες αυτές τις πτυχές.

## 2.2 Η ΕΠΑΝΑΛΗΨΗ ΓΕΝΙΚΑ

Ένα σύστημα λογισμικού περνάει από κάποιο αριθμό κύκλων ανάπτυξης κατά την διάρκεια της ζωής του. Κάθε κύκλος έχει ως αποτέλεσμα μια καινούργια έκδοση του προϊόντος λογισμικού για τους πελάτες και τους χρήστες και η πρώτη έκδοση μπορεί να είναι και η πιο δύσκολη. Βάζει τα θεμέλια και την αρχιτεκτονική του συστήματος και μπορεί να εισέρχεται σε καινούργια πεδία που θα περιέχουν σοβαρούς κινδύνους. Κάθε κύκλος ανάπτυξης λοιπόν έχει διαφορετικό περιεχόμενο ανάλογα με το σημείο ολόκληρου του κύκλου ζωής στο οποίο βρίσκεται το λογισμικό. Αν η πρώτη έκδοση λοιπόν εισέρχεται σε ένα καινούργιο πεδίο τότε θα απαιτηθεί σημαντικό μέρος της εργασίας στις πρώτες φάσεις. Ενώ αν έχουμε να κάνουμε με εξέλιξη κάποιας έκδοσης τότε η περισσότερη εργασία θα καταναλωθεί στις τελευταίες φάσεις. Και αυτό γιατί η αρχιτεκτονική του συστήματος θα πρέπει να είναι επεκτάσιμη και το καινούργιο προϊόν απλά να χτιστεί επάνω σε ότι ήδη έχει χτιστεί, δηλαδή με άλλα λόγια μια καινούργια έκδοση ενός προϊόντος λογισμικού θα χτιστεί επάνω στην προηγούμενη έκδοση. Υπάρχει βέβαια και η περίπτωση να έχουν αλλάξει δραματικά το περιβάλλον του συστήματος και οι απαιτήσεις των χρηστών, οπότε και η μετέπειτα έκδοση θα απαιτήσει σημαντικό μέρος της εργασίας και στις πρώτες φάσεις.

Όλο και περισσότεροι άνθρωποι ενστερνίζονται την ιδέα ότι τα προβλήματα καλύτερα να αντιμετωπίζονται στην αρχή κάθε κύκλου ανάπτυξης παρά προς το τέλος. Οι κίνδυνοι μερικές φορές όμως είναι καλά κρυμμένοι. Πρέπει να αναγνωριστούν, να τοποθετηθούν τα όρια τους, να παρακολουθηθούν και στο τέλος να αντιμετωπιστούν. Και το καλύτερο είναι να αντιμετωπιστούν πρώτα οι μεγαλύτεροι κίνδυνοι. Παρόμοια, η σειρά με την οποία θα γίνουν οι επαναλήψεις της επαναληπτικής διαδικασίας πρέπει να προγραμματιστούν προσεχτικά ώστε τα πιο σοβαρά προβλήματα να αντιμετωπίζονται πρώτα. Στα επόμενα θα δούμε τα κοινά χαρακτηριστικά που έχουν όλες οι επαναλήψεις και των τεσσάρων φάσεων. Ο τρόπος με τον οποίο σχεδιάζουμε την κάθε επανάληψη είναι ο ίδιος. Το μόνο που διαφέρει είναι το περιεχόμενο της κάθε επανάληψης το οποίο μεταβάλλεται ανάλογα με τους στόχους της κάθε φάσης. Κάθε επανάληψη έχει πέντε βασικές εργασίες: καταγραφή απαιτήσεων, ανάλυση, σχεδίαση, υλοποίηση και δοκιμή. Επιπλέον περιέχει τον προγραμματισμό της επανάληψης, που προηγείται των πέντε εργασιών και την αποτίμηση της επανάληψης, που ακολουθεί μετά από τις πέντε εργασίες. Ο προγραμματισμός είναι απαραίτητος καθόλη την διάρκεια του κύκλου ανάπτυξης. Αλλά πριν προγραμματίσουμε πρέπει να γνωρίζουμε τι έχουμε να κάνουμε. Οι πέντε βασικές εργασίες παρέχουν ένα αρχικό σημείο. Η διαχείριση του κινδύνου, που σημαίνει η αναγνώριση και η αντιμετώπιση των κινδύνων μέσω της πραγματοποίησης των σχετικών περιπτώσεων χρήσης είναι ακόμα μια κύρια πτυχή του προγραμματισμού. Σίγουρα κανένας προγραμματισμός δεν είναι ολοκληρωμένος αν δεν πραγματοποιεί μια

εκτίμηση των πόρων που απαιτούνται και φυσικά το αποτέλεσμα της κάθε επανάληψης πρέπει να αποτιμηθεί (εικ. 6).



Εικ. 6 Η γενική επανάληψη

Σε κάθε στιγμή του κύκλου ζωής ενός έργου ανάπτυξης λογισμικού μια μεγάλη σειρά εργασιών πραγματοποιείται. Εργασίες πάνω σε καινούργιες λειτουργίες, εργασίες πάνω στην αρχιτεκτονική του συστήματος, στην καταγραφή των απόψεων των χρηστών, στην αντιμετώπιση των κινδύνων, στον προγραμματισμό μελλοντικών εργασιών και ούτω καθεξής. Πρέπει κάθε στιγμή να συγχρονίζουμε και να εξισορροπούμε αυτές τις διαφορετικές σειρές εργασιών, που όπως είναι λογικό αυξάνουν το μέγεθος της πολυπλοκότητας. Οι κατασκευαστές χωρίζουν τη συνολική εργασία, η οποία είναι υπερβολικά πολύπλοκη στο σύνολο του έργου, σε μικρότερα κομμάτια εύκολα κατανοητά. Δηλαδή καθόλη τη διάρκεια της ανάπτυξης του κύκλου ζωής του λογισμικού, χωρίζουν την εργασία σε φάσεις, και κάθε μια φάση σε επαναλήψεις. Μέσα σε κάθε επανάληψη το έργο προσπαθεί να ισορροπήσει ανάμεσα στις σειρές εργασιών που πραγματοποιούνται κατά την διάρκεια της επανάληψης. Αυτό σημαίνει ότι πρέπει να εργαστούμε πάνω στα κατάλληλα πράγματα την κατάλληλη στιγμή. Τα κατάλληλα πράγματα πάνω στα οποία πρέπει να δουλέψουμε κάθε φορά εξαρτώνται από το που

βρισκόμαστε στον κύκλο ζωής. Σε κάθε έργο είναι απαραίτητο να επιλεγούν τα κατάλληλα πράγματα, πάνω στα οποία θα επικεντρωθεί η εργασία, για κάθε μία σειρά εργασιών. Στην εύρεση της ισορροπίας μεταξύ των σειρών εργασιών είναι εξίσου σημαντικό να διασφαλίσουμε ότι τα πράγματα με τα οποία θα ασχοληθούμε είναι συγκρίσιμα ως προς τη σημαντικότητα τους, ώστε να μπορούν αποτελεσματικά να μπουν σε σειρά προτεραιότητας και να συγχρονιστούν. Τυχόν αποτυχία σε αυτού του είδους την ισορροπία και αποτυχία εκτέλεσης φαίνεται ότι ακυρώνουν την επαναληπτική και με σταδιακή ανάπτυξη κατασκευή έργων λογισμικού. Σε μια αρχική επανάληψη εργαζόμαστε πάνω σε κρίσιμους κινδύνους, κύριες περιπτώσεις χρήσης, θέματα αρχιτεκτονικής, επιλογή του περιβάλλοντος ανάπτυξης και γενικά με ενέργειες που είναι σχετικές με έρευνα. Ενώ σε μια από τις τελευταίες επαναλήψεις εργαζόμαστε πάνω σε εργασίες που έχουν να κάνουν με την ανάπτυξη όπως υλοποίηση και δοκιμή, αξιολόγηση προβλημάτων απόδοσης και κατανομής του συστήματος. Η συσχέτιση όλων αυτών των εργασιών μεταξύ τους είναι μια ευαίσθητη εργασία. Είναι αυτή η ευαισθησία που καθιστά δύσκολη την ανάπτυξη λογισμικού. Η κατανόηση αυτών των διαφορετικών εργασιών και η εύρεση ισορροπίας μεταξύ τους είναι κάτι που κάνουμε σε κάθε επανάληψη. Στην ενοποιημένη διεργασία μερικές από αυτές τις σειρές εργασιών έχουν αναγνωρισθεί και χαρακτηρισθεί ως βασικές εργασίες. Υπάρχουν και άλλες εργασίες που δεν έχουν αναγνωρισθεί επίσημα άλλα μπορούν να μεταχειριστούν κατά τον ίδιο τρόπο. Μερικές από αυτές είναι οι εξής:

- Η επικοινωνία με τους πελάτες για τυχόν καινούργιες απαιτήσεις.
- Ετοιμάζοντας μια οικονομική προσφορά για τους πελάτες.
- Η κατανόηση του περιβάλλοντος του συστήματος δημιουργώντας ένα επιχειρηματικό πλάνο.
- Ο προγραμματισμός και η διεύθυνση του έργου.
- Η εγκατάσταση και επίβλεψη του περιβάλλοντος ανάπτυξης που είναι τα εργαλεία ανάπτυξης που θα χρησιμοποιηθούν.
- Η αντιμετώπιση των κινδύνων.
- Η παρουσίαση του προϊόντος στον χώρο του πελάτη.
- Η ανταπόκριση σε σχόλια των χρηστών.

### **2.3 Η ΣΥΛΛΗΨΗ ΓΕΝΙΚΑ**

Το πρώτο βήμα προς τον χωρισμό της ανάπτυξης του συστήματος σε κομμάτια είναι ο χωρισμός σε τέσσερις φάσεις: σύλληψη, επεξεργασία, κατασκευή και μετάβαση. Κάθε φάση χωρίζεται περαιτέρω σε μια ή περισσότερες επαναλήψεις. Ο κύριος στόχος της φάσης της σύλληψης είναι δημιουργία της επιχειρηματικής χρήσης, της χρήσης που θα μας κάνει να προχωρήσουμε με το έργο. Αυτή η χρήση θα αναπτυχθεί περαιτέρω στην φάση της επεξεργασίας όπου ολοένα και περισσότερες πληροφορίες θα γίνονται διαθέσιμες. Η φάση της σύλληψης δεν είναι μία ολοκληρωμένη μελέτη του προτεινόμενου συστήματος. Ψάχνουμε μόνο ένα μικρό ποσοστό των περιπτώσεων χρήσεων για να υποστηρίξουμε την επιχειρηματική χρήση. Για να δημιουργήσουμε αυτή την χρήση πραγματοποιούμε τα εξής τέσσερα βήματα:

1. Οριοθέτηση του πεδίου του προτεινόμενου συστήματος, δηλαδή ο καθορισμός των ορίων του συστήματος και η αναγνώριση των διασυνδέσεων του συστήματος με τα εξωτερικά συστήματα.
2. Περιγραφή μιας υποψήφιας αρχιτεκτονικής του προτεινόμενου συστήματος ειδικά για εκείνα τα μέρη του συστήματος που εμπεριέχουν ρίσκο ή μεγάλο βαθμό δυσκολίας. Εκτελούμε αυτό το βήμα συνήθως μέχρι την περιγραφή της αρχιτεκτονικής και σπάνια ως ένα εκτελέσιμο πρωτότυπο. Η περιγραφή της αρχιτεκτονικής περιέχει την αρχική εικόνα των όψεων των μοντέλων. Εδώ ο στόχος είναι να κάνουμε πιστευτό ότι μπορούμε να δημιουργήσουμε μια σταθερή αρχιτεκτονική του προτεινόμενου συστήματος στην επόμενη φάση. Δεν κατασκευάζουμε αυτήν την αρχιτεκτονική σε αυτήν τη φάση. Απλά θέλουμε να καταστήσουμε εμφανές ότι μπορούμε να την κατασκευάσουμε. Η κατασκευή της αρχιτεκτονικής είναι η βασική εργασία στη φάση της επεξεργασίας.
3. Η αναγνώριση των σημείων που εμπεριέχουν ρίσκο, δηλαδή των σημείων που επηρεάζουν τη δυνατότητα κατασκευής του συστήματος, και η λήψη της απόφασης αν μπορούμε να τα αντιμετωπίσουμε έστω και σε μια επόμενη φάση. Σε αυτήν τη φάση ασχολούμαστε μόνο με πιθανούς κινδύνους οι οποίοι θα επηρεάσουν την επιτευξιμότητα του συστήματος. Κάθε άλλος κίνδυνος που τυχόν αναγνωρίζεται τον τοποθετούμε σε μια λίστα για να αντιμετωπισθεί στην επόμενη φάση.
4. Η επίδειξη στους μελλοντικούς χρήστες ή πελάτες ότι το προτεινόμενο σύστημα είναι σε θέση να λύσει το πρόβλημα του ή να υποστηρίξει την επιχείρηση στο αντικείμενο της. Σε αυτήν τη φάση μπορεί να φτιάξουμε ένα πρωτότυπο για να επιδείξουμε τη λύση του προβλήματος στους πελάτες. Το πρωτότυπο θα παρουσιάζει τις βασικές ιδέες του καινούργιου συστήματος και θα επικεντρώνεται στη χρήση του. Ο σκοπός αυτού του πρωτότυπου είναι περισσότερο διερευνητικός, δηλαδή παρουσιάζει μία πιθανή λύση και μπορεί να μην εξελιχθεί στο τελικό προϊόν. Συνήθως αυτό το πρωτότυπο καταστρέφεται μετά την ολοκλήρωση του τελικού προϊόντος. Σε αντίθεση το πρωτότυπο αρχιτεκτονικής που δημιουργείται κατά την φάση της επεξεργασίας είναι αυτό πάνω στο οποίο θα γίνει η περαιτέρω εξέλιξη της αρχιτεκτονικής του συστήματος στις επόμενες φάσεις.

Οι εργασίες σε αυτή τη φάση φτάνουν ως το σημείο που να αποδεικνύεται ότι είναι οικονομικά επωφελές να αναπτύξουμε το σύστημα. Οι εργασίες μας θα δείξουν ότι το σύστημα θα μας παρέχει κέρδη ή άλλα πλεονεκτήματα σε σχέση με την επένδυση που απαιτείται για να το κατασκευάσουμε. Με άλλα λόγια έχουμε κάνει ένα πρώτο σχέδιο του επιχειρηματικού πλαισίου το οποίο θα το βελτιώσουμε στην επόμενη φάση της επεξεργασίας.

Ο σκοπός μας σε αυτήν τη φάση είναι να ελαχιστοποιήσουμε τις δαπάνες σε χρόνο, προσπάθεια και οικονομικούς πόρους μέχρι να ανακαλύψουμε ότι το σύστημα είναι εφικτό. Σε περίπτωση που έχουμε ένα μεγάλο μεγέθους σύστημα ενός άγνωστου πεδίου λειτουργίας τότε για τις αποφάσεις μας σε αυτήν τη φάση θα χρειαστούμε σημαντικό

χρόνο και προσπάθεια η οποία μπορεί να επεκτείνεται σε πολλές επαναλήψεις. Για ένα γνωστό σύστημα που θα λειτουργεί σε ένα συνηθισμένο και γνωστό περιβάλλον ή για την επέκταση ενός υπάρχοντος συστήματος, οι κίνδυνοι και τα άγνωστα σημεία είναι λίγα, οπότε μας επιτρέπει να ολοκληρώσουμε αυτή την φάση σε μικρότερο χρόνο.

## 2.4 Η ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΝΙΚΑ

Το πρωτεύων προϊόν της φάσης της επεξεργασίας είναι μία σταθερή αρχιτεκτονική η οποία θα οδηγήσει το σύστημα καθόλη την διάρκεια της μελλοντικής του ζωής. Αυτή η φάση φτάνει ως το σημείο προγραμματισμού της φάσης της κατασκευής με μεγάλη λεπτομέρεια. Με γνώμονα τους δύο παραπάνω στόχους οι κατασκευαστές πραγματοποιούν σε αυτή την φάση τα εξής:

1. Δημιουργία μιας αρχικής αρχιτεκτονικής που καλύπτει τις σημαντικές αρχιτεκτονικές λειτουργίες και χαρακτηριστικά του συστήματος.
2. Αναγνώριση ρίσκων, δηλαδή σημείων που εμπεριέχουν τον κίνδυνο να ανατρέψουν τα πλάνα, το κόστος, τον προγραμματισμό επόμενων φάσεων και η αντιμετώπιση τους με τον καθορισμό ενεργειών που απαιτούνται και οι οποίες πρέπει να είναι και από την άποψη του χρόνου άλλα και από την άποψη του κόστους μετρήσιμες.
3. Καθορισμός του επιπέδου που πρέπει να επιτευχθεί στα ποιοτικά χαρακτηριστικά του συστήματος όπως αξιοπιστία και χρόνοι ανταπόκρισης.
4. Καταγραφή έως και του 80% περίπου των λειτουργικών περιπτώσεων χρήσης, αρκετών δηλαδή για τη φάση της κατασκευής.
5. Ετοιμασία ενός πλάνου για τους οικονομικούς πόρους και το ανθρώπινο δυναμικό που απαιτείται.

Οι απαιτήσεις του συστήματος και η αρχιτεκτονική είναι τα σημεία στα οποία επικεντρώνεται η μεγαλύτερη προσπάθεια κατά τις φάσεις της σύλληψης και επεξεργασίας.

## 2.5 Η ΚΑΤΑΣΚΕΥΗ ΓΕΝΙΚΑ

Ο κύριος στόχος της φάσης της κατασκευής υποδεικνύεται από το βασικό του σημείο ορόσημο, το οποίο είναι μια αρχική λειτουργική δυνατότητα. Αυτό σημαίνει ότι το προϊόν είναι έτοιμο για δοκιμή. Αυτή η φάση απασχολεί, περισσότερο από κάθε άλλη φάση, το περισσότερο ανθρώπινο δυναμικό για τη μεγαλύτερη χρονική περίοδο. Για αυτό και είναι τόσο σημαντικό όλα τα σημαντικά σημεία να έχουν ξεκαθαριστεί πριν μπούμε σε αυτήν τη φάση. Συνήθως, η φάση της κατασκευής πραγματοποιείται με μεγαλύτερο αριθμό επαναλήψεων από ότι οι προηγούμενες φάσεις.

Οι ενέργειες που περιλαμβάνονται σε αυτήν τη φάση είναι οι εξής:

1. Επέκταση των περιπτώσεων χρήσης έως το σημείο που να εξαντληθούν εντελώς όλες οι δυνατές περιπτώσεις χρήσης του συστήματος.
2. Διεκπεραίωση των εργασιών της ανάλυσης, σχεδίασης, κατασκευής και ελέγχου.
3. Διατήρηση της ακεραιότητας της αρχιτεκτονικής και μεταβολή της μόνο όποτε χρειάζεται.

4. Επίβλεψη κρίσιμων και σημαντικών ρίσκων που παραληφθήκαν από τις προηγούμενες φάσεις και η αντιμετώπιση τους.

## 2.6 Η ΜΕΤΑΒΑΣΗ ΓΕΝΙΚΑ

Η φάση της μετάβασης συχνά ξεκινάει με την έκδοση του προϊόντος δοκιμής, το οποίο είναι ένα προϊόν λογισμικού με δυνατότητα λειτουργίας και διανέμεται από τους κατασκευαστές σε ένα αντιπροσωπευτικό κομμάτι των χρηστών. Η λειτουργία του συστήματος στο απαιτητικό περιβάλλον των χρηστών είναι πολλές φορές η πιο δύσκολη δοκιμασία από ότι στο περιβάλλον των κατασκευαστών. Η φάση της μετάβαση περιλαμβάνει τις ακόλουθες ενέργειες:

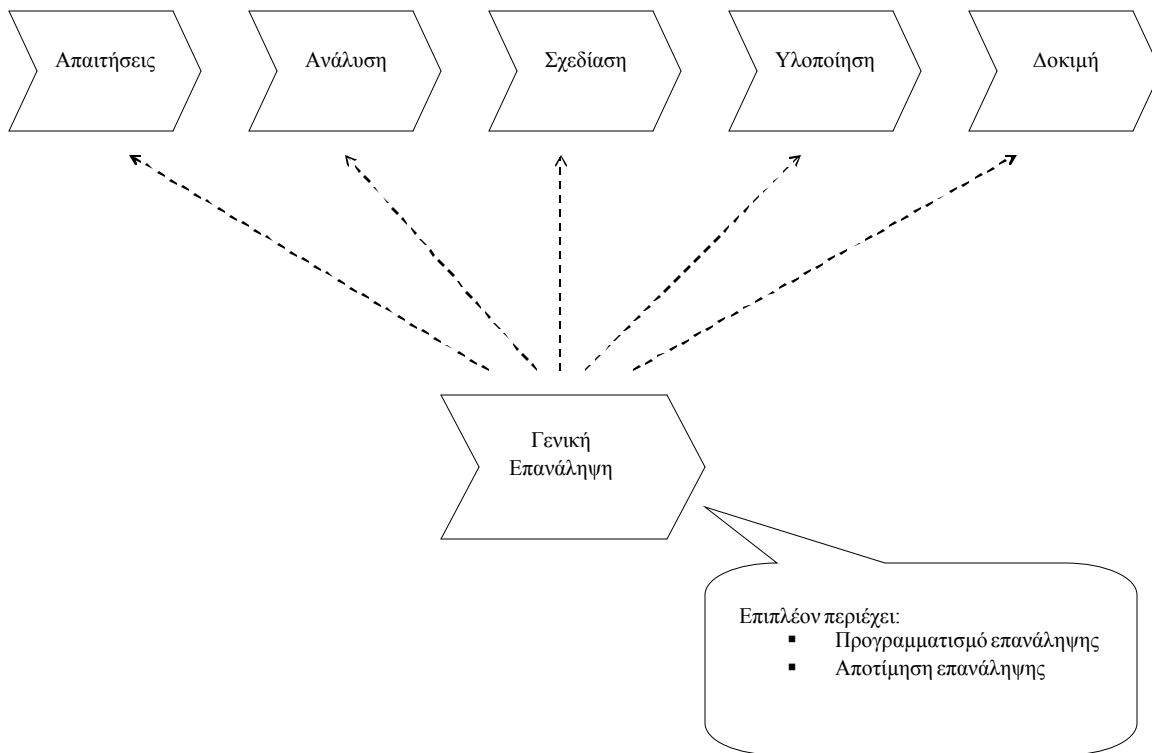
1. Ενέργειες ετοιμασίας, όπως για παράδειγμα κάποια ενημερωτική ιστοσελίδα για το καινούργιο λογισμικό.
2. Ενημέρωση των πελατών για τυχόν αναβαθμίσεις που χρειάζονται ως προς το περιβάλλον τους, τις οποίες χρειάζεται το καινούργιο λογισμικό για να λειτουργήσει.
3. Ετοιμασία των εγχειριδίων χρήσης για την έκδοση του προϊόντος.
4. Προσαρμογή του λογισμικού για να δουλέψει με τις πραγματικές παραμέτρους του περιβάλλοντος ενός χρήστη.
5. Διόρθωση ατελειών που βρέθηκαν μετά τη δοκιμή του λογισμικού από τους χρήστες-δοκιμαστές.
6. Μεταβολή του λογισμικού λόγω προβλημάτων που παρουσιάστηκαν τώρα και δεν είχαν προβλεφθεί.

Η φάση της μετάβασης τελειώνει με την επίσημη έκδοση του προϊόντος. Παρόλα αυτά πριν την τελική εκχώρηση του έργου οι κατασκευαστές κάνουν μια αποτίμηση του έργου με τους ακόλουθους στόχους:

- Να μελετήσουν, να αξιολογήσουν και να καταγράψουν τα καινούργια στοιχεία που αντιμετώπισαν κατά την διάρκεια του έργου ώστε να αποτελέσουν σημεία αναφοράς στο μέλλον.
- Να καταγράψουν στα σημεία που αξίζουν μελέτης σε επόμενες εκδόσεις τους έργου.

## 3 Η ΓΕΝΙΚΗ ΕΠΑΝΑΛΗΨΗ

Πρέπει να δείξουμε την διαφορά μεταξύ των βασικών σειρών εργασιών και των επαναληπτικών σειρών εργασιών. Οι βασικές σειρές εργασιών είναι καταγραφή των απαιτήσεων, η ανάλυση, η σχεδίαση, η υλοποίηση και η δοκιμή. Στην ενοποιημένη διεργασία αυτές οι βασικές σειρές εργασιών δεν πραγματοποιούνται μόνο μια φορά, όπως συμβαίνει με την μέθοδο του καταρράκτη, άλλα επαναλαμβάνονται σε κάθε επανάληψη. Σε κάθε επανάληψη όμως μεταβάλλεται το περιεχόμενο τους ανάλογα με την επανάληψη (εικ. 7). Εκτός από τις βασικές σειρές εργασιών κάθε επανάληψη περιλαμβάνει και τις ενέργειες του προγραμματισμού της επανάληψης και της αποτίμησής της.



Εικ. 7 Η γενική επανάληψη

Στο ξεκίνημα της φάσης της σύλληψης έχουμε στο νου μας τρία πράγματα:

- Πρόκειται να ολοκληρώσουμε το έργο μας με μια σειρά από επαναλήψεις σε τέσσερις φάσεις.
- Έχουμε πληροφορίες για το προτεινόμενο σύστημα, πληροφορίες που οδήγησαν στην ανάλυση του έργου.
- Έχουμε κάποιες δικές μας πληροφορίες για τον χώρο στον οποίο δραστηριοποιείται το προτεινόμενο σύστημα και πληροφορίες για παρόμοια συστήματα τα οποία αναπτύξαμε στο παρελθόν.

Με βάση αυτές τις πληροφορίες πρέπει να κάνουμε έναν προγραμματισμό του έργου και έναν προγραμματισμό κάθε επανάληψης. Στην αρχή, λόγω των ελάχιστων πληροφοριών που διαθέτουμε, αυτοί οι δύο προγραμματισμοί περιέχουν ελάχιστες πληροφορίες. Καθώς δουλεύουμε κατά τις φάσεις της σύλληψης και επεξεργασίας αυτοί οι προγραμματισμοί εμπλουτίζονται σε περιεχόμενο.

Στην αρχή θα μάθουμε πώς να σχεδιάζουμε τις φάσεις και τις επαναλήψεις και πώς να αξιολογούμε την επανάληψη. Στη συνέχεια θα δούμε πώς οι κίνδυνοι επηρεάζουν το σχεδιασμό μας και πώς αντιμετωπίζουμε αυτούς τους κινδύνους επιλέγοντας τις κατάλληλες περιπτώσεις χρήσης. Στο τέλος θα δούμε την κατανομή των πόρων.



### 3.1 ΣΧΕΔΙΑΣΜΟΣ ΤΩΝ ΤΕΣΣΑΡΩΝ ΦΑΣΕΩΝ

Ξέρουμε από την μεθοδολογία της ενοποιημένης διεργασίας τι περιέχουν οι κάθε φάσεις. Στο σχεδιασμό του έργου στόχος μας είναι να κάνουμε αυτό το περιεχόμενο διακριτό:

- Κατανομή του χρόνου. Αποφασίζουμε πόσο χρόνο θα αφιερώσουμε σε κάθε φάση και την ημερομηνία έως την οποία η κάθε φάση θα πρέπει να έχει ολοκληρωθεί. Αυτοί οι χρόνοι παρόλο που καθορίζονται με λεπτομέρεια στην αρχή, μπορεί να είναι όμως αμφίβολοι στην αρχή της φάσης της σύλληψης άλλα θα σταθεροποιηθούν καθώς προχωράμε. Εξάλλου μόνο όταν θα φτάσουμε στο τέλος της φάσης της επεξεργασίας θα κάνουμε ένα αυστηρό χρονοδιάγραμμα.
- Βασικά σημεία ορόσημο. Μια φάση έχει ολοκληρωθεί όταν κάποιο κριτήριο έχει ικανοποιηθεί.
- Επαναλήψεις σε κάθε φάση. Σε κάθε φάση αναπτύσσουμε το έργο μέσα από μια σειρά επαναλήψεων. Ο γενικός χαρακτήρας της κάθε επανάληψης περιέχεται στο σχέδιο του έργου.
- Σχέδιο έργου. Το σχέδιο του έργου περιέχει την γενική πορεία του έργου που σημαίνει το χρονοδιάγραμμα, τις ημερομηνίες των σημείων ορόσημο και των κριτηρίων, και το διαχωρισμό των φάσεων σε επαναλήψεις.

Η πρώτη επανάληψη της φάσης της σύλληψης θα είναι μάλλον δύσκολη. Να περιμένετε ότι έχετε να κάνετε με τα εξής:

- Να προσαρμόσετε την ενοποιημένη διεργασία ώστε να ταιριάζει στο έργο σας και να επιλέξετε τα εργαλεία για την αυτοματοποίηση της διαδικασίας.
- Η ανεύρεση ανθρώπων με ειδικότητες που απαιτούνται για το έργο.
- Το χτίσιμο διαπροσωπικών σχέσεων που θα καταστήσουν την ομάδα κατασκευής του έργου παραγωγική.
- Η κατανόηση του πεδίου στο οποίο θα λειτουργήσει το προτεινόμενο σύστημα, το οποίο πολλές φορές μπορεί να είναι καινούργιο στην ομάδα κατασκευής.
- Η κατανόηση της φύσης του έργου, που θα είναι πιο δύσκολη για ένα έργο που ξεκινάει από την αρχή από ένα έργο το οποίο θα επεκταθεί.
- Η εξοικείωση της ομάδας κατασκευής με απαραίτητα εργαλεία για την διαδικασία και το έργο.

### 3.2 ΣΧΕΔΙΑΣΜΟΣ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ

Κάθε φάση περιέχει μία ή περισσότερες επαναλήψεις. Ο σχεδιασμός των επαναλήψεων γίνεται μέσα από μια σειρά βημάτων που είναι περίπου παρόμοια με αυτών που ακολουθούνται όταν σχεδιάζουμε τις φάσεις:

- Χρονοδιάγραμμα επανάληψης. Καθορίζουμε πόσο χρόνο κάθε επανάληψη επιτρέπεται να καταναλώσει και ορίζουμε την ημερομηνία ολοκλήρωσης της, στην αρχή στο περίπου και στη συνέχεια, καθώς αποκτάμε περισσότερες πληροφορίες, με ακρίβεια.
- Περιεχόμενο επανάληψης. Παρόλο που ο γενικός σχεδιασμός τους έργου περιγράφει τα κύρια σημεία των επαναλήψεων, σχεδιάζουμε το τι είναι να γίνει με περισσότερες λεπτομέρειες. Το περιεχόμενο μιας επανάληψης θα βασιστεί στα εξής:

- Στο ποιες περιπτώσεις χρήσης είναι να συμπληρωθούν έστω κατά ένα μέρος τους στη διάρκεια της επανάληψης.
- Για ποια τεχνικά σημεία που εμπεριέχουν κινδύνους έχει έρθει η ώρα να αναγνωριστούν, να συνδεθούν με τις αντίστοιχες περιπτώσεις χρήσης και να αντιμετωπισθούν.
- Στις αλλαγές στις απαιτήσεις που έχουν προκύψει ή στις ατέλειες που έχουν επισημανθεί.
- Στο ποια υποσυστήματα πρόκειται να υλοποιηθούν μερικώς ή στο σύνολο τους. Αυτό το σημείο ποικίλει ανάλογα με την φάση στην οποία βρισκόμαστε. Στην φάση της επεξεργασίας για παράδειγμα, καθορίζουμε τα περισσότερα από τα υποσυστήματα και όλες τις αρχιτεκτονικά σημαντικές κλάσεις. Στην φάση της κατασκευής προσθέτουμε στα υποσυστήματα όλο και περισσότερες λειτουργίες με αποτέλεσμα όλο και πιο ολοκληρωμένα συστατικά.

Ο σχεδιασμός μιας επανάληψης είναι καταγεγραμμένος με λεπτομέρεια, ενώ ο σχεδιασμός μιας επόμενης επανάληψης γίνεται όλο και πιο λεπτομερής αφού αποκτάμε όλο και περισσότερες πληροφορίες. Οι πληροφορίες για μετέπειτα επαναλήψεις μπορεί να είναι περιορισμένες αυτήν τη χρονική στιγμή. Επιπλέον στοιχείο που λαμβάνεται υπόψη είναι τα δευτερεύοντα σημεία ορόσημο. Η επίτευξη κριτηρίων που καθορίστηκαν στο σχεδιασμό της επανάληψης σηματοδοτεί το τέλος της επανάληψης. Ο αριθμός των επαναλήψεων για κάθε φάση ποικίλει ανάλογα με την πολυπλοκότητα του προτεινόμενου συστήματος. Ένα απλό έργο μπορεί να πραγματοποιηθεί με μία μόνο επανάληψη για κάθε φάση. Ένα πιο πολύπλοκο έργο θα περιλαμβάνει περισσότερες από μία επαναλήψεις για κάθε φάση. Για παράδειγμα:

- Φάση της σύλληψης. Μία επανάληψη κυρίως αφιερωμένη στον σκοπό του συστήματος.
- Φάση της επεξεργασίας. Δύο επαναλήψεις, η μία για την δημιουργία μιας γενικής εικόνας της αρχιτεκτονικής, και η δεύτερη για την δημιουργία του βασικού τμήματος της αρχιτεκτονικής.
- Φάση της κατασκευής. Δύο επαναλήψεις για την επιβεβαίωση ότι τα επιπλέον κομμάτια λειτουργούν ικανοποιητικά.
- Φάση της μετάβασης. Μία επανάληψη για τον έλεγχο του έργου.

Όσο το υπό κατασκευή έργο είναι πιο μεγάλο και πιο πολύπλοκο πρέπει να περιμένουμε το μέγεθος της όλης διαδικασίας να μεγαλώνει ανάλογα. Σε μεγάλα έργα το πλήθος των επαναλήψεων είναι μεγάλο, όπως και η διάρκεια τους, που μπορεί να κυμαίνεται από μία εβδομάδα έως και τρεις μήνες για μία επανάληψη.

### **3.2.1 ΣΧΕΔΙΑΣΜΟΣ ΤΟΥ ΚΡΙΤΗΡΙΟΥ ΑΠΟΤΙΜΗΣΗΣ**

Οι επαναλήψεις είναι μικρές σε σχέση με τα παραδοσιακά έργα ανάπτυξης λογισμικού. Για να αποτρέψουμε την παρατεταμένη διάρκεια της επανάληψης εκτός από τον καθορισμό ενός χρονοδιαγράμματος πρέπει να περιγράψουμε και το βασικό στόχο της κάθε επανάληψης. Ο βασικός στόχος της κάθε επανάληψης αποτελεί το κριτήριο που υποδεικνύει την ολοκλήρωση του έργου. Αυτό το κριτήριο βοηθάει επιπλέον και στην

καλύτερη εστίαση της προσπάθειας κατά τη διάρκεια της επανάληψης. Παραδείγματα κριτηρίων αποτελούν τα εξής:

- Λειτουργικές απαιτήσεις, εκφραζόμενες με την μορφή περιπτώσεων χρήσης.
- Μη λειτουργικές απαιτήσεις, συνδεδεμένες με τις περιπτώσεις χρήσης στις οποίες εφαρμόζονται.
- Μη λειτουργικές απαιτήσεις, που δεν συνδέονται με συγκεκριμένες απαιτήσεις χρήσης και περιέχονται στο σχέδιο των συμπληρωματικών απαιτήσεων.

Ένας από τους στόχους σε μια από τις αρχικές επαναλήψεις μπορεί να είναι για παράδειγμα, η αποσαφήνιση αμφιβολιών για εκφραζόμενες από τον πελάτη απαιτήσεις. Το κριτήριο καθορίζει τι πρέπει να πετύχει μια επανάληψη με όρους που είναι μετρήσιμοι, όπως η απόδοση, ή μπορούν να διαπιστωθούν, όπως μία επιθυμητή λειτουργία. Ο επιβλέπων το έργο θέτει το κριτήριο αποτίμησης στην αρχή κάθε επανάληψης και κάθε φάσης. Κάθε μία χρειάζεται να έχει ένα ξεκάθαρο σημείο τέλους όπου και οι κατασκευαστές από μόνοι τους θα μπορούν να παρατηρούν πόσο έχουν προχωρήσει. Επιπλέον αυτά τα κριτήρια παρέχουν σημεία ορόσημο όπου μπορεί να εκτιμηθεί η όλη πρόοδος του έργου. Γενικά τα κριτήρια αξιολόγησης μπορούν να χωριστούν σε δύο κατηγορίες, αυτά που επαληθεύουν απαιτήσεις χρήσης και αυτά που είναι πιο γενικής μορφής. Αυτοί που μετέχουν στη διαδικασία δοκιμής του έργου συμμετέχουν στην ανάπτυξη του έργου από τη φάση της σύλληψης. Αναγνωρίζουν ποια χαρακτηριστικά των περιπτώσεων χρήσης μπορούν να επιβεβαιωθούν με δοκιμές. Ετοιμάζουν τις περιπτώσεις δοκιμής του συστήματος. Στην ενοποιημένη διεργασία ακόμη και ο κύκλος δοκιμής του συστήματος είναι επίσης επαναληπτικός. Κάθε κατασκευή που πραγματοποιείται από κάποια επανάληψη αποτελεί αντικείμενο προς δοκιμή. Οι δοκιμαστές προσθέτουν και μεταβάλουν τις δοκιμές που ήδη έχουν γίνει στις κατασκευές συνθέτοντας ένα σώμα δοκιμών που θα εκτελεστεί σε μετέπειτα στάδια. Οι αρχικές επαναλήψεις παράγουν περισσότερες καινούργιες λειτουργίες για το σύστημα, οπότε και περισσότερα σχέδια δοκιμών από ότι οι τελικές επαναλήψεις αντίστοιχα. Επίσης οι δοκιμές σε αρχικές επαναλήψεις έχουν να κάνουν περισσότερο με δοκιμές σχεδίασης, ενώ οι δοκιμές των τελευταίων επαναλήψεων έχουν να κάνουν με δοκιμές εκτέλεσης και απόδοσης.

### 3.2.2 ΟΙ ΚΙΝΔΥΝΟΙ

Ο τρόπος με τον οποίο σχεδιάζουμε την ανάπτυξη ενός καινούργιου συστήματος επηρεάζεται σε μεγάλο βαθμό από τον τρόπο που αντιλαμβανόμαστε τους κινδύνους. Οπότε, ένα από τα πρώτα βήματα στην αρχή της φάσης της σύλληψης είναι να δημιουργήσουμε μια λίστα με κινδύνους. Στην αρχή μπορεί να δυσκολευτούμε από την έλλειψη πληροφοριών, αλλά το πιθανότερο είναι να έχουμε μια αίσθηση των κρίσιμων σημείων, δηλαδή αυτών που θα καθορίσουν αν θα έχουμε τη δυνατότητα να αναπτύξουμε το σύστημα. Καθώς προχωράμε με τις εργασίες, θα έχουμε όλο και καλύτερη γνώση των κινδύνων για σημαντικά σημεία του έργου, τέτοιων που πρέπει να αντιμετωπισθούν για να είμαστε σε θέση να κάνουμε ένα χρονοδιάγραμμα και ένα οικονομικό πλάνο και να πετύχουμε ένα ικανοποιητικό στόχο.

### 3.2.2.1 ANTIMETΩΠΙΣΗ ΤΩΝ ΚΙΝΔΥΝΩΝ

Είναι δεδομένο ότι η ανάπτυξη λογισμικού εμπεριέχει κινδύνους. Ένα άλλο δεδομένο χαρακτηριστικό είναι ότι πρέπει να εντοπιστούν και να αντιμετωπιστούν. Αυτός είναι και ο στόχος της λίστας με τους κινδύνους. Δεν είναι κάτι το οποίο συμπληρώνεται και μετά παραμελείται. Όλα όσα πρέπει να ξέρει κάποιος για κάποιο σημείο κινδύνου για να δουλέψει με αυτό πρέπει να υπάρχει μέσα στη λίστα, η οποία πρέπει να περιλαμβάνει και τα εξής:

- Περιγραφή. Γενική περιγραφή η οποία συμπληρώνεται καθώς παίρνουμε περισσότερες πληροφορίες.
- Προτεραιότητα. Ορίζεται μια προτεραιότητα ξεκινώντας με τα πιο κρίσιμα μετά τα σημαντικά και στη συνέχεια τα συνηθισμένα. Όσο η λίστα ενημερώνεται θα είναι πιθανό να υπάρξουν και περισσότερες κατηγορίες.
- Επίπτωση. Ποια μέρη του έργου ή του συστήματος επηρεάζονται από το επικίνδυνο σημείο.
- Επίβλεψη. Ποιος είναι υπεύθυνος να καταγράφει την εξέλιξη ενός συνεχή κινδύνου.
- Υπευθυνότητα. Ποιος είναι υπεύθυνος να αντιμετωπίσει τον κίνδυνο.
- Απρόοπτα. Τι πρέπει να γίνει σε περίπτωση που πραγματοποιηθεί ο κίνδυνος.

Σε ένα έργο μεγάλου μεγέθους οι κίνδυνοι μπορεί να είναι εκατοντάδες. Οπότε πρέπει να μπου σε μια βάση δεδομένων ώστε να μπορεί να γίνει μια αποτελεσματική ταξινόμηση και αναζήτηση. Η αντιμετώπιση των κινδύνων δεν μπορεί να γίνει με μια φορά. Αυτός είναι ένας λόγος ύπαρξης της επαναληπτικής ανάπτυξης. Οι κίνδυνοι ταξινομούνται ανάλογα με το βαθμό σημαντικότητας ή ανάλογα με την επίπτωση που έχουν στο σύστημα και αντιμετωπίζονται με τη σειρά. Όπως έχουμε επισημάνει, οι κίνδυνοι που αντιμετωπίζονται πρώτοι είναι αυτοί που μπορούν να κάνουν το σύστημα να αποτύχει. Κάποιοι κίνδυνοι δεν είναι εύκολοι στην αντιμετώπιση τους και παραμένουν στη λίστα για αρκετό καιρό. Η λίστα με τους κινδύνους δεν είναι ένα στατικό εργαλείο. Καθώς ανακαλύπτονται επιπλέον κίνδυνοι η λίστα μεγαλώνει. Καθώς οι κίνδυνοι αντιμετωπίζονται ή περνάμε το σημείο έως το οποίο θα μπορούσαν να είχαν εμφανισθεί αφαιρούνται από την λίστα.

### 3.2.2.2 ΟΙ ΚΙΝΔΥΝΟΙ ΕΠΗΡΕΑΖΟΥΝ ΤΟ ΠΛΑΝΟ ΤΗΣ ΕΠΑΝΑΛΗΨΗΣ

Κατά την φάση της σύλληψης τα κρίσιμα σημεία αναγνωρίζονται και γίνεται προσπάθεια να αντιμετωπιστούν. Γίνεται μια διερεύνηση της φύσης των κινδύνων στο σημείο κατά το οποίο γίνεται το πλάνο της επανάληψης. Για να αποκτηθούν περισσότερες πληροφορίες μπορεί να χρειαστεί για παράδειγμα να αναπτυχθούν μερικές περιπτώσεις χρήσης σχετιζόμενες με τον κίνδυνο και να κατασκευαστεί ένα πρωτότυπο. Μετά θα γίνει εισαγωγή κάποιων συγκεκριμένων δεδομένων που θα οδηγήσουν στην εμφάνιση του κινδύνου και την παραγωγή του σφάλματος. Εκτός από την επίδραση που έχουν οι σημαντικοί κίνδυνοι στην επιτυχία του έργου, όλοι οι κίνδυνοι έχουν κάποια επίπτωση στο χρονοδιάγραμμα, στο κόστος και στην ποιότητα του έργου. Κάποιοι από αυτούς τους κινδύνους μπορεί να είναι τόσο σημαντικοί ώστε να επιμηκύνουν τη διάρκεια του έργου ή να αυξήσουν το κόστος του πέρα από αυτό που έχει προγραμματιστεί, εκτός και αν

έχουν αντιμετωπιστεί πριν εμφανισθούν τα ανεπιθύμητα αποτελέσματα τους. Σε σχεδόν όλες τις περιπτώσεις κάποια μεταβολή στο χρονοδιάγραμμα έχει επίπτωση και στην προσπάθεια που καταβάλλεται αλλά και στο κόστος. Σε άλλες περιπτώσεις, ενώ οι κίνδυνοι έχουν μικρή επίπτωση στο χρονοδιάγραμμα ή στο κόστος, επηρεάζουν άλλες πτυχές του έργου, όπως η ποιότητα και η απόδοση.

### **3.2.2.3 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΝΕΡΓΕΙΩΝ ΚΑΤΑ ΤΩΝ ΚΙΝΔΥΝΩΝ**

Ο γενικός κανόνας είναι οι ενέργειες που πραγματοποιούνται κατά των κινδύνων να είναι προγραμματισμένες. Οι φάσεις και οι επαναλήψεις μεταξύ των φάσεων παρέχουν το μηχανισμό που απαιτείται για τον προγραμματισμό των ενεργειών κατά των κινδύνων. Για παράδειγμα, σύμφωνα με την ενοποιημένη διεργασία ο προγραμματισμός των ενεργειών που απαιτούνται για την αντιμετώπιση των κινδύνων που επηρεάζουν τη δυνατότητα κατασκευής του συστήματος γίνεται στις επαναλήψεις της φάσης της σύλληψης. Τότε είναι που οι κίνδυνοι αντιμετωπίζονται ή τουλάχιστον δημιουργείται ένα εφικτό πλάνο για τη μετέπειτα αντιμετώπιση τους. Η εμπειρία δείχνει ότι ο μη προγραμματισμός του τρόπου αντιμετώπισης των κινδύνων δεν έχει καλά αποτελέσματα. Στην περίπτωση απουσίας προσπάθειας αντιμετώπισης των κινδύνων στα πρώτα στάδια, οι κίνδυνοι εμφανίζονται στη φάση της δοκιμής του συστήματος. Σε εκείνο το σημείο η αντιμετώπιση τέτοιων σοβαρών προβλημάτων θα προκαλέσει εκτεταμένες αλλαγές στο σύστημα οι οποίες θα καθυστερήσουν την παράδοση του έργου κατά εβδομάδες ή και παραπάνω. Στην ενοποιημένη διεργασία η κατασκευή πρωτοτύπων και σχεδίων από την πρώτη φάση αποκαλύπτει τους κινδύνους όσο υπάρχει ακόμα χρόνος να αντιμετωπιστούν δυναμικά. Αναγνωρίζουμε βέβαια τη δυσκολία που υπάρχει στην αναγνώριση και περιγραφή κάποιων τύπων κινδύνων. Για κάποιους λόγους κάποιοι κίνδυνοι μπορεί να δείχνουν καλά κρυμμένοι γιατί συνήθως δεν γίνεται αρκετή προσπάθεια αναζήτησής τους. Παρόλα αυτά, όποιος και να είναι ο λόγος, σε κάποια έργα κάποιοι κίνδυνοι θα αγνοηθούν μέχρι προς το τέλος του έργου, ειδικά αν η ομάδα ανάπτυξης έχει λίγη εμπειρία στη διαχείριση κινδύνων. Με την εξάσκηση και την εμπειρία οι ομάδες ανάπτυξης θα βελτιώσουν την ικανότητα τους στην τοποθέτηση των κινδύνων σε σειρά προτεραιότητας που θα επιτρέψει το έργο να προχωρήσει σε ένα λογικό μονοπάτι. Στη φάση της κατασκευής για παράδειγμα, οι κίνδυνοι που μπορεί να αναγκάσουν τη δεύτερη επανάληψη να βγει εκτός χρονοδιαγράμματος πρέπει να αντιμετωπιστούν το αργότερο μέχρι την πρώτη επανάληψη εκείνης της φάσης. Ο στόχος είναι κάθε επανάληψη της φάσης της κατασκευής να προχωράει αδιατάρακτα και σύμφωνα με το πλάνο. Αυτό δεν πρόκειται να συμβεί αν το έργο συναντήσει ένα αναπάντεχο σημείο κινδύνου το οποίο δεν μπορεί να αντιμετωπισθεί σύντομα και αποτελεσματικά.

### **3.2.2.4 ΣΕΙΡΑ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ ΤΩΝ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ**

Σε αυτήν την ενότητα θα δείξουμε τον τρόπο επιλογής των περιπτώσεων χρήσης ως οδηγούς μέσα σε μια επανάληψη. Κάθε επανάληψη της ενοποιημένης διεργασίας οδηγείται από ένα σύνολο περιπτώσεων χρήσης. Στην πραγματικότητα, είναι καλύτερο

να πούμε ότι μια επανάληψη οδηγείται από ένα σύνολο σεναρίων μέσω των περιπτώσεων χρήσης. Είναι πιο ακριβές γιατί στις αρχικές επαναλήψεις δεν παίρνουμε αναγκαστικά ολόκληρες περιπτώσεις χρήσης. Παίρνουμε μόνο εκείνα τα σενάρια ή μονοπάτια που είναι σχετικά με την εργασία που έχουμε να κάνουμε. Για αυτό και όταν λέμε ότι επιλέγουμε περιπτώσεις χρήσης εννοούμε ότι επιλέγουμε τα σενάρια που είναι σχετικά με την επανάληψη. Η εργασία που συντελεί στην επιλογή ονομάζεται τοποθέτηση των περιπτώσεων χρήσης σε σειρά προτεραιότητας. Οι περιπτώσεις χρήσεις, ή τα σενάρια τους, μπαίνουν σε μια σειρά ανάλογα με τη σειρά που πρέπει να διαχειριστούν μέσα στις επαναλήψεις. Κατατάσσονται σε πολλές επαναλήψεις. Σε αρχικές επαναλήψεις αρκετές περιπτώσεις χρήσης κατατάσσονται, αλλά πολλές είναι και αυτές που δεν έχουν ακόμα καταγραφεί οπότε δεν μπορούν να καταταχθούν. Όλες οι καινούργιες περιπτώσεις χρήσης κατατάσσονται μόλις εντοπιστούν. Η κατάταξη υπάρχει σε μια λίστα κατάταξης περιπτώσεων χρήσης. Η διαμόρφωση αυτή της λίστας είναι ένα ρίσκο. Κατατάσσουμε τις περιπτώσεις χρήσης σε μια σειρά ανάλογα με τον κίνδυνο που εμπεριέχουν. Εδώ ο όρος κίνδυνος έχει ένα πιο γενικό νόημα. Για παράδειγμα, η ανάγκη για αλλαγή της αρχιτεκτονικής του συστήματος σε μετέπειτα φάσεις είναι ένας κίνδυνος που θέλουμε να αποφύγουμε. Το να μην κατασκευάσουμε το σωστό σύστημα είναι ένας κίνδυνος που θέλουμε να αντιμετωπιστεί στην αρχή βρίσκοντας τις πραγματικές απαιτήσεις. Η διαδικασία της επιλογής λοιπόν γίνεται με βάση τον κίνδυνο. Τοποθετούμε τους κινδύνους λοιπόν που αναγνωρίζουμε, σε μια λίστα κινδύνων όπως είπαμε προηγουμένως και συσχετίζουμε κάθε κίνδυνο με μια περίπτωση χρήσης που όταν υλοποιείται αντιμετωπίζει τον κίνδυνο. Αυτή η περίπτωση χρήσης μετά θα μπει σε μια θέση στη λίστα προτεραιότητας των περιπτώσεων χρήσης που θα ανταποκρίνεται στο βαθμό επικινδυνότητας της. Σε αρχικές επαναλήψεις λοιπόν, οι ενέργειες μας στη δημιουργία σειράς προτεραιότητας των περιπτώσεων χρήσης αφορούν κυρίως τις περιπτώσεις χρήσης που έχουν να κάνουν με την εμβέλεια του συστήματος και την αρχιτεκτονική. Σε μετέπειτα επαναλήψεις επιλέγουμε καινούργιες περιπτώσεις χρήσης που εμπλουτίζουν την αρχιτεκτονική του συστήματος προσθέτοντας καινούργιες λειτουργίες. Οι επόμενες περιπτώσεις χρήσης μπαίνουν σε μια λογική σειρά. Για παράδειγμα, οι περιπτώσεις χρήσης που προαπαιτούν κάποιες άλλες περιπτώσεις χρήσης για να λειτουργήσουν, κατατάσσονται χαμηλότερα στην λίστα οπότε και αναπτύσσονται αργότερα από αυτές που έχουν ανάγκη για να λειτουργήσουν. Στα επόμενα θα δείξουμε τις τρεις κατηγορίες κινδύνων, τους συγκεκριμένους κινδύνους, τους αρχιτεκτονικούς κινδύνους και τους κινδύνους των απαιτήσεων.

### **3.2.2.5 ΤΥΠΟΙ ΚΙΝΔΥΝΩΝ**

Πρώτα από όλα έχουμε τους κινδύνους που είναι σχετικοί με ένα συγκεκριμένο προϊόν. Αυτό είναι το είδος των τεχνικών κινδύνων. Συσχετίζονται με περιπτώσεις χρήσης που όταν υλοποιούνται σωστά τότε αντιμετωπίζουν τον κίνδυνο. Πρέπει να αναγνωρίσουμε αυτούς τους κινδύνους έναν προς έναν πριν η παρουσία τους επηρεάσει την πορεία της ανάπτυξης.

Επιπλέον κίνδυνος είναι αυτός που είναι σχετικός με την αρχιτεκτονική. Ένας από τους κύριους κινδύνους είναι να κατασκευαστεί το σύστημα κατά τέτοιο τρόπο ώστε να μην

μπορεί να αναπτυχθεί ομαλά στις επόμενες φάσεις ή κατά τη διάρκεια ζωής του, δηλαδή να μην κατασκευαστεί το σύστημα με ελαστική και προσαρμοστική αρχιτεκτονική. Ο κίνδυνος αντιμετωπίζεται εκτενώς στη φάση της σύλληψης και επεξεργασίας, όταν βλέπουμε ότι έχουμε τη σωστή αρχιτεκτονική και μπορούμε να την παγώσουμε, εκτός από μικρές αλλαγές βέβαια στη φάση της κατασκευής. Τίθενται όμως κάποια ερωτήματα, όπως πώς γίνεται η επιλογή των περιπτώσεων χρήσης που θα οδηγήσουν στην κατασκευή της σωστής αρχιτεκτονικής και πώς αντιμετωπίζουμε τον κίνδυνο να μην έχουμε μια σταθερή αρχιτεκτονική. Η απάντηση είναι ότι επιλέγουμε αυτές που είναι αρχιτεκτονικά σημαντικές, δηλαδή αυτές που καλύπτουν τις βασικές λειτουργίες που θα εκτελεί το σύστημα. Στο δεύτερο ερώτημα η απάντηση είναι οι κρίσιμες περιπτώσεις χρήσης, δηλαδή, οι περιπτώσεις χρήσης που είναι οι πιο σημαντικές για τους χρήστες του συστήματος. Επιπλέον, οι περιπτώσεις χρήσης που έχουν σημαντικές μη λειτουργικές απαιτήσεις, όπως απόδοση και χρόνος απόκρισης εντάσσονται σε αυτήν την κατηγορία των κρίσιμων περιπτώσεων χρήσης. Αυτές οι περιπτώσεις χρήσης είναι συνήθως που συντελούν στην εύρεση του σκελετού του συστήματος πάνω στον οποίο προσθέτουμε τις υπόλοιπες λειτουργίες. Άλλες περιπτώσεις χρήσης είναι οι εξής:

- Δευτερεύουσες. Αυτές οι περιπτώσεις χρήσης υποστηρίζουν τις κρίσιμες. Περιέχουν δευτερεύουσες λειτουργίες όπως συλλογή και επίβλεψη στατιστικών λειτουργίας. Αυτές οι περιπτώσεις χρήσης έχουν ένα πολύ μικρό ποσοστό επιρροής της αρχιτεκτονικής, αν και μερικές φορές μπορεί και αυτές να χρειαστεί αναπτυχθούν νωρίς, λόγω κάποιας ιδιαίτερης απαίτησης του πελάτη, οπότε θα πρέπει να μετακινηθούν σε υψηλότερη θέση στην λίστα των περιπτώσεων χρήσης ώστε να αποφευχθεί ο κίνδυνος της μη ικανοποίησης των πραγματικών απαιτήσεων.
- Βοηθητικές. Αυτές οι περιπτώσεις χρήσης δεν επηρεάζουν την αρχιτεκτονική ούτε εμπεριέχουν κινδύνους. Αυτές οι περιπτώσεις χρήσης σπάνια επεξεργάζονται κατά τις φάσεις της σύλληψης και επεξεργασίας. Αν τυχόν συμβεί κάτι τέτοιο σημαίνει ότι είναι συμπληρωματικές για την ολοκλήρωση των κρίσιμων και σημαντικών περιπτώσεων χρήσης.
- Επιλεκτικές. Μερικές περιπτώσεις χρήσης μπορεί να είναι κρίσιμες ή σημαντικές παρόλο που μπορεί να μην είναι συνέχεια παρούσες. Ίσως χρειαστεί να τις αντιμετωπίσουμε γιατί επηρεάζουν την αρχιτεκτονική όταν είναι παρούσες.

Επιπλέον θέλουμε να είμαστε σίγουροι ότι έχουμε αναγνωρίσει όλες τις περιπτώσεις χρήσης που πιθανώς να επηρεάσουν την αρχιτεκτονική του συστήματος. Δεν θέλουμε να αγνοήσουμε καμιά λειτουργικότητα την οποία θα την ανακαλύψουμε αργότερα και δεν θα έχουμε μια σταθερή αρχιτεκτονική. Χρειάζεται να καλύψουμε πλήρως όλες τις περιπτώσεις χρήσης που μπορούν να επηρεάσουν την αρχιτεκτονική. Η εκτενής κάλυψη είναι σημαντική όχι μόνο για την εύρεση της αρχιτεκτονικής αλλά και για να είμαστε σίγουροι ότι μπορούμε να προβλέψουμε το κόστος της ανάπτυξης του συστήματος. Πρέπει να αποφύγουμε τον κίνδυνο να ανακαλύψουμε πολύ αργά ότι δεν μπορούμε να ικανοποιήσουμε μια καινούργια επιθυμητή λειτουργία του συστήματος. Για αυτό το λόγο πρέπει να καλύψουμε περίπου το 80% των περιπτώσεων χρήσης έως τη φάση της επεξεργασίας. Με τον όρο κάλυψη εννοούμε να κατανοήσουμε τις περιπτώσεις χρήσης και την επίπτωση που έχουν στο σύστημα. Στην πράξη αναγνωρίζουμε το 80% των περιπτώσεων χρήσης και τις περιλαμβάνουμε στο μοντέλο περιπτώσεων χρήσης, άλλα συνήθως δεν είναι απαραίτητο να περιγράψουμε με κάθε λεπτομέρεια όλες από αυτές.

Σε ένα τυπικό έργο μπορεί να βρούμε ότι είναι απαραίτητο να περιγράψουμε μόνο κάποια μέρη από τις περιπτώσεις χρήσης. Κάποιες από αυτές τις περιγραφές μπορεί να είναι γενικές, αφού όμως είναι ικανές να διαλευκάνουν τι απαιτείται να γνωρίζουμε σε αυτήν τη φάση. Σε σχετικά απλά έργα μπορεί να περιγράψουμε ένα ελάχιστο ποσοστό των περιπτώσεων χρήσης όταν βρίσκουμε τις απαιτήσεις. Σε μεγάλα έργα με υψηλούς κινδύνους είναι φρόνιμο να περιγράψουμε με λεπτομέρεια το 80% ή και παραπάνω των περιπτώσεων χρήσης.

Ένας άλλος κίνδυνος είναι αυτός που είναι σχετικός με τις απαιτήσεις. Δηλαδή να μην κατασκευάσουμε το σύστημα που να κάνει αυτό που οι τελικοί χρήστες πραγματικά θέλουν να κάνει. Με το τέλος της φάσης της επεξεργασίας θέλουμε να είμαστε σίγουροι ότι κατασκευάζουμε το σωστό σύστημα. Αυτό δεν μπορεί να αλλάξει μετά, γιατί το κόστος αυξάνεται στη φάση της κατασκευής. Τίθενται τα ερωτήματα, όπως ποιες περιπτώσεις χρήσης απαιτούνται για να είμαστε σίγουροι ότι το σύστημα που αναπτύσσουμε είναι το σωστό για τους χρήστες, ποιες περιπτώσεις χρήσης βεβαιώνουν ότι το σύστημα μπορεί να αναπτυχθεί έτσι στις επόμενες φάσεις ώστε να ικανοποιεί όλες τις αρχικές απαιτήσεις. Η απάντηση στην πρώτη ερώτηση είναι να κάνουμε την εργασία των απαιτήσεων σωστά. Μπορεί να δημιουργήσουμε ένα επιχειρηματικό μοντέλο ή σε κάποιες περιπτώσεις μόνο ένα περιορισμένο μοντέλο του πεδίου. Η απάντηση στη δεύτερη ερώτηση είναι μέσα από τις αρχικές επαναλήψεις και την κατασκευή ενός πρωτότυπου που χρειάζονται οι χρήστες ώστε να καταθέσουν τις γνώμες τους όσο τον δυνατόν πιο νωρίς. Μόνο μέσα από την πραγματική χρήση μπορούμε να είμαστε σίγουροι ότι έχουμε κατασκευάσει το σωστό σύστημα.

### 3.2.3 ΠΟΡΟΙ ΠΟΥ ΧΡΕΙΑΖΟΝΤΑΙ

Μπορεί να φαίνεται ότι το επαναληπτικό πλάνο της βασισμένης σε φάσεις ανάπτυξης λογισμικού κυλάει πολύ ομαλά. Αλλά δεν έχουν απαντηθεί ακόμα τα εξής ερωτήματα:

- Πόσο πρόκειται να κοστίσουν οι φάσεις της σύλληψης και επανάληψης, σε σχέση με το εξειδικευμένο προσωπικό που απαιτείται και σε σχέση με την προσπάθεια που πρέπει να καταβληθεί.
- Το οικονομικό κόστος για αυτές τις φάσεις ποιος θα το καλύψει.
- Πόσος χρόνος θα καταναλωθεί σε αυτές τις φάσεις.
- Πόσο χρόνο θα καθυστερήσουν οι πρώτες φάσεις την φάση της κατασκευής που θεωρείται στον επιχειρηματικό κόσμο η ουσιαστική ανάπτυξη λογισμικού.

#### 3.2.3.1 ΔΙΑΦΟΡΑ ΕΡΓΑ

Είναι δεδομένο ότι τα προτεινόμενα έργα διαφέρουν πολύ ως προς την ετοιμότητα τους να περάσουν κατευθείαν στη φάση της ανάπτυξης. Για παράδειγμα:

1. Ένα τελείως καινούργιο προϊόν το οποίο θα λειτουργεί σε ένα εντελώς άγνωστο περιβάλλον (green field). Κανένας δεν ξέρει τι πρέπει να γίνει ή αν μπορεί κίόλας να γίνει. Είναι ελάχιστη η εμπειρία πάνω στην οποία μπορείς να βασιστείς και να προχωρήσεις. Πρέπει να βασιστούμε σε έμπειρους ανθρώπους που να κάνουν



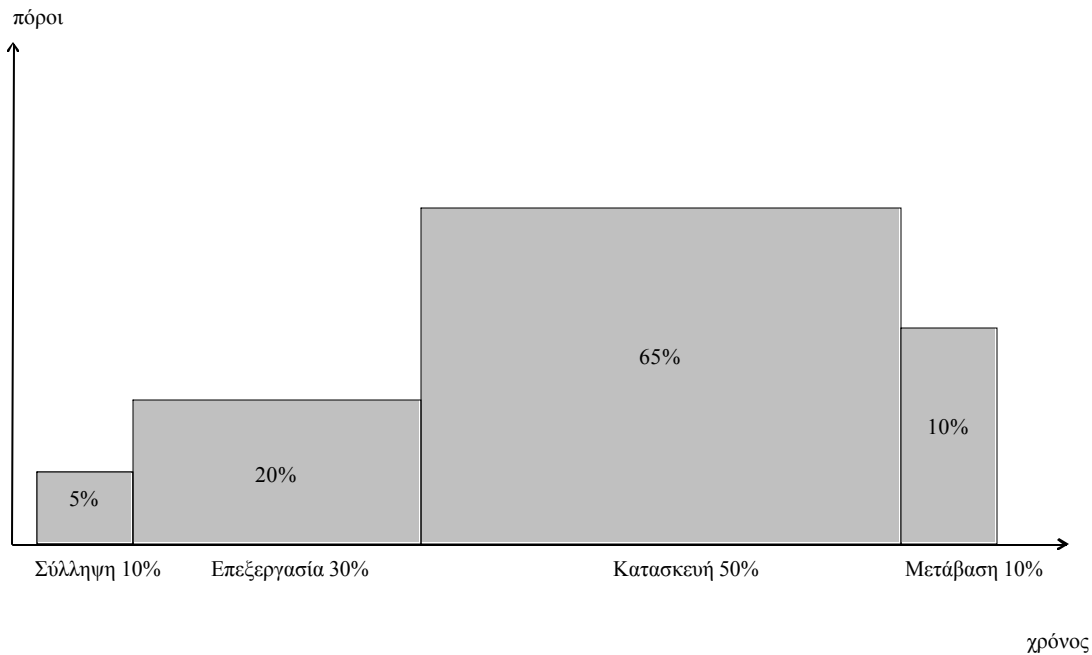
τεκμηριωμένες υποθέσεις. Κάτω από αυτές τις συνθήκες αυτός που ζητάει ένα τέτοιο σύστημα είναι κατά κάποιο τρόπο υπεύθυνος και για τη χρηματοδότηση των φάσεων της σύλληψης και επεξεργασίας. Οι φάσεις πρέπει να χρηματοδοτηθούν ως περίπου με την μορφή κάποιας έρευνας και θα είναι ένα επιπλέον κόστος εκτός του βασικού κόστους. Η έλλειψη πληροφοριών δεν μπορεί να κρατήσει το έργο στις φάσεις της σύλληψης και επεξεργασίας μέσα σε προβλεπόμενο προϋπολογισμό ή χρονοδιάγραμμα. Σε αυτού του είδους την περίπτωση ο προσδιορισμός του αντικειμένου, η εύρεση μιας υποψήφιας αρχιτεκτονικής, η αναγνώριση των κρίσιμων κινδύνων και η δημιουργία του επιχειρηματικού πλάνου είναι χρονοβόρες εργασίες για τη φάση της σύλληψης. Παρομοίως, για να επιτευχθεί και ο στόχος της φάσης της επεξεργασίας που είναι η ετοιμασία του έργου για την φάση της κατασκευής θα πάρει περισσότερο χρόνο.

2. Ο τύπος προϊόντος που έχει κατασκευαστεί στο παρελθόν για ένα συγκεκριμένο περιβάλλον, όπου τα προηγούμενα προϊόντα μας παρέχουν παραδείγματα αλλά όχι επαναχρησιμοποιήσιμα συστατικά. Αυτά τα προηγούμενα παραδείγματα παρέχουν έναν οδηγό για την υποψήφια αρχιτεκτονική, αλλά μπορεί να πάρει μερικές μέρες μέχρι να βεβαιωθούμε ότι η προηγούμενη αρχιτεκτονική όντως ταιριάζει. Υπό αυτές τις συνθήκες η φάση της σύλληψης θα είναι πιθανώς σύντομη. Θα χρειαστεί μόνο ένα ή δύο έμπειρους ανθρώπους. Επειδή αυτός ο τύπος προϊόντος έχει κατασκευαστεί στο παρελθόν, οι βασικοί κίνδυνοι θα είναι λίγοι, αλλά ίσως να χρειαστεί μερικές μέρες ώστε να αναγνωρισθούν.
3. Ένα προϊόν υπάρχει, αλλά υπάρχει η ανάγκη να εκσυγχρονισθεί, όπως για παράδειγμα να μετατραπεί μια εφαρμογή κεντρικού υπολογιστή σε εφαρμογή εξυπηρετητή-πελάτη. Ως ένα σημείο μέρη του προηγούμενου προϊόντος μπορούν να ενσωματωθούν και να χρησιμοποιηθούν και στο καινούργιο σύστημα. Στη φάση της σύλληψης πρέπει να βρεθεί μια υποψήφια αρχιτεκτονική.
4. Ο τύπος προϊόντος του οποίου τα συστατικά υπάρχουν είτε στο εμπόριο είτε εσωτερικά στην εταιρία ανάπτυξης λογισμικού (in-house). Η εταιρία περιμένει ότι ένα σημαντικό ποσοστό του καινούργιου συστήματος, περίπου 50% έως 90%, μπορεί να συναρμολογηθεί από αυτά τα συστατικά, αλλά θα υπάρξουν και κενά τα οποία πρέπει να συμπληρωθούν με καινούργιο κώδικα. Οι κατασκευαστές πρέπει να καθορίσουν τις διασυνδέσεις μεταξύ των επαναχρησιμοποιήσιμων συστατικών και των καινούργιων συστατικών, όπως και μεταξύ των εξωτερικών συστημάτων και χρηστών. Η ανάπτυξη πάνω σε υπάρχοντα συστατικά θέλει χρόνο και προσπάθεια και μπορεί να παρουσιαστούν κίνδυνοι. Στο σύνολο πάντως η ανάπτυξη με προηγούμενα συστατικά είναι πιο γρήγορη και πιο οικονομική από την ανάπτυξη από την αρχή.

Αυτά τα παραδείγματα έργων δεν έχουν σκοπό να διαμορφώσουν διακριτές κατηγορίες. Αντιθέτως αντιπροσωπεύουν παραδείγματα με αρκετά κοινά χαρακτηριστικά. Για αυτό πρέπει να τα βλέπουμε ως σημεία εκκίνησης.

### 3.2.3.2 ΕΝΑ ΤΥΠΙΚΟ ΕΡΓΟ

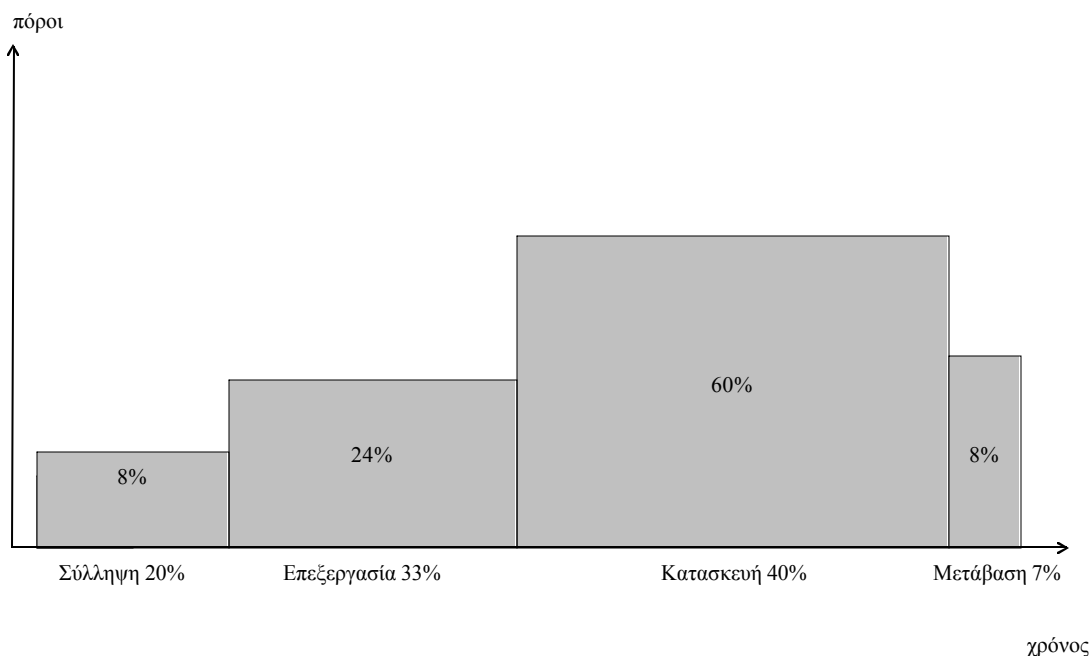
Παρά τις ιδιαιτερότητες που περιέχουν οι διάφοροι τύποι έργων, ο αρχικός κύκλος ανάπτυξης ενός μέτριου μεγέθους έργου θα κατανέμει το ποσοστό των εργασιών περίπου όπως δείχνει η εικόνα 8. Το μεγαλύτερο μέρος της προσπάθειας και του χρόνου απαιτεί η φάση της κατασκευής, άλλα και οι άλλες τρεις φάσεις απαιτούν σημαντικό ποσοστό εργασιών και χρόνου.



Εικ. 8 Ποσοστιαίος καταμερισμός της εργασίας κατά τη διάρκεια ενός τυπικού έργου

### 3.2.3.3 ΠΟΛΥΠΛΟΚΑ ΕΡΓΑ ΕΧΟΥΝ ΜΕΓΑΛΥΤΕΡΕΣ ΑΝΑΓΚΕΣ

Ποια θα είναι όμως η επίπτωση αν υποθέσουμε ότι έχουμε ένα μεγάλο και ποιο πολύπλοκο έργο, ένα έργο με καινούργιες λειτουργίες, κατανεμημένη αρχιτεκτονική, ή για παράδειγμα λειτουργία σε πραγματικό χρόνο, που χρησιμοποιεί καινούργια τεχνολογία. Πιθανώς να έχουμε να κάνουμε ένα μεγαλύτερο αριθμό επαναλήψεων. Θα πρέπει να καταβάλουμε περισσότερη προσπάθεια και χρόνο στις φάσεις της σύλληψης και επεξεργασίας. Ως αποτέλεσμα είναι αυτές οι φάσεις να μεγαλώσουν όπως φαίνεται και στην εικόνα 9.



Εικ. 9 Ποσοστιαίος καταμερισμός της εργασίας κατά τη διάρκεια ενός πολύπλοκου έργου

Τα ποσοστά που φαίνονται στις εικόνες δεν είναι πραγματικά άλλα ενδεικτικά. Έχουν σκοπό να υποδείξουν ότι όσο πιο άγνωστο είναι ένα σύστημα, τόσο περισσότερο χρόνο και προσπάθεια πρέπει να καταναλωθεί στις φάσεις της σύλληψης και επεξεργασίας. Έτσι βλέπουμε στο παράδειγμα της εικόνας 9 ότι ξοδεύουμε περισσότερο χρόνο στις δύο πρώτες φάσεις από τις δύο τελευταίες φάσεις. Παρόλα αυτά, η προσπάθεια δεν έχει την ίδια επίπτωση οπότε, αν και αυξάνεται κατά ένα ποσοστό, δεν χρειάζεται να αυξήσουμε τους πόρους με την ίδια αναλογία. Σαν σύνολο πάντως προσπάθεια και χρόνος αυξάνονται την ίδια στιγμή.

### 3.2.3.4 ΤΟ ΚΑΙΝΟΥΡΓΙΟ ΠΡΟΪΟΝ ΑΠΑΙΤΕΙ ΕΜΠΕΙΡΙΑ

Στις περισσότερες περιπτώσεις και κυρίως σε καινούργια και δύσκολα συστήματα η ομάδα που αναπτύσσει το λογισμικό πρέπει να αποκτήσει πληροφορίες περισσότερες από τις οποίες διαθέτει. Η πηγή αυτών των πληροφοριών είναι άτομα ειδικευμένα στο πεδίο του προτεινόμενου συστήματος. Ακόμα και αν υπάρχουν λεπτομερείς απαιτήσεις η ομάδα χρειάζεται αυτές να μπουν σε μια σειρά για να βρει την αρχιτεκτονική και να επικεντρωθεί στους κινδύνους. Η εύρεση ειδικευμένων ατόμων είναι αρκετά σημαντική υπόθεση. Οι απλοί χρήστες το μόνο που μπορεί να ξέρουν είναι το δικό τους κομμάτι εργασίας και όχι την όλη διαδικασία. Ένα συνηθισμένο λάθος κατά την έναρξη

ανάπτυξης ενός καινούργιου προϊόντος λογισμικού είναι να γίνεται προσπάθεια ανάπτυξης του χωρίς την χρησιμοποίηση της προϋπάρχουσας τεχνογνωσίας. Αφού κιάλας η περισσότερη τεχνογνωσία μιας εταιρίας ανάπτυξης λογισμικού υπάρχει στο μυαλό των ανθρώπων που δουλεύουν και όχι σε έντυπη μορφή, η επαναχρησιμοποίηση της τεχνογνωσίας έχει να κάνει με την επαναχρησιμοποίηση έμπειρων ατόμων.

### **3.2.4 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΚΑΙ ΤΩΝ ΦΑΣΕΩΝ**

Για να γίνει εκμετάλλευση στο έπακρο των πλεονεκτημάτων του επαναληπτικού τρόπου εργασίας, πρέπει να γίνεται μια αποτίμηση του τι έχει επιτευχθεί στο τέλος κάθε επανάληψης και κάθε φάσης. Η εργασία αυτή δεν γίνεται μόνο για την αποτίμηση άλλα και για την προώθηση δύο επιπλέον στόχων:

- Στον επανασχεδιασμό της επόμενης επανάληψης με βάση το τι έχουμε μάθει από την παρούσα επανάληψη και στο να προβούμε στις απαραίτητες μετατροπές.
- Στην προσαρμογή της διαδικασίας από την αποτίμηση της παρούσας επανάληψης.

Ο βασικός στόχος της αποτίμησης είναι να ελεγχθεί το τι έχει επιτευχθεί σε σχέση με τα κριτήρια αξιολόγησης της παρούσας επανάληψης. Ο δεύτερος στόχος είναι ο απολογισμός της όλης διαδικασίας απαντώντας στα εξής ερωτήματα:

- Το έργο προχωράει εντός του οικονομικού προϋπολογισμού και εντός του χρονοδιαγράμματος;
- Από τις μετρήσεις και τους ελέγχους που γίνονται στο πρωτότυπο και στα συστατικά ικανοποιούνται οι ποιοτικές απαιτήσεις;

Το ιδανικό είναι το έργο να πληρεί τα παραπάνω κριτήρια. Όμως οι αποτιμήσεις σπάνια προχωρούν τόσο ομαλά. Συχνά μια επανάληψη δεν εκπληρώνει τα κριτήρια σε ικανοποιητικό βαθμό. Οπότε η εργασία πρέπει να γίνει στην επόμενη ή σε κάποια από τις επόμενες επαναλήψεις. Αυτή η εργασία μπορεί να περιλαμβάνει:

- Μεταβολή ή επέκταση του μοντέλου περιπτώσεων χρήσης.
- Μεταβολή ή επέκταση της αρχιτεκτονικής.
- Μεταβολή ή επέκταση των υποσυστημάτων που έχουν αναπτυχθεί έως τώρα.
- Αναζήτηση περαιτέρω κινδύνων.

Ένα σημείο που πρέπει να μελετηθεί σοβαρά είναι τα κριτήρια. Η ομάδα μπορεί να είχε καθορίσει τα κριτήρια σε κάποια χρονική στιγμή που δεν είχε διαθέσιμες όλες τις σχετικές πληροφορίες. Κατά την επανάληψη μπορεί να ανακάλυψε κάποια απαραίτητα στοιχεία που αρχικά κρίθηκαν μη απαραίτητα. Οπότε πρέπει να γίνει μία αλλαγή στα κριτήρια και όχι απλά ο έλεγχος αν έχουν επιτευχθεί.

### **3.2.5 Η ΕΠΟΜΕΝΗ ΕΠΑΝΑΛΗΨΗ**

Ένα βασικό σημείο ορόσημο χαρακτηρίζει το σημείο ολοκλήρωσης μιας φάσης, δηλαδή το σημείο όπου όχι μόνο η ομάδα ανάπτυξης του έργου, αλλά όλοι οι εμπλεκόμενοι στο έργο, όπως οι επενδυτές και οι πελάτες, εγκρίνουν ότι το έργο έχει πετύχει ένα κριτήριο ορόσημο και επομένως δικαιολογείται η μετάβαση του σε επόμενη φάση. Στη φάση της αποτίμησης γίνονται επιπλέον τα εξής:

- Η λήψη της απόφασης ότι η εργασία είναι έτοιμη να προχωρήσει στην επόμενη επανάληψη.
- Αν απαιτείται επιπλέον εργασία τότε αποφασίζεται σε ποια από τις επόμενες φάσεις θα γίνει αυτή η εργασία.
- Ο λεπτομερής σχεδιασμός της επόμενης φάσης.
- Η ενημέρωση με όχι και τόσες λεπτομέρειες των επαναλήψεων πέρα της επόμενης.
- Η ενημέρωση της λίστα κινδύνων και του πλάνου του έργου.
- Η σύγκριση του πραγματικού κόστους και χρόνου που απαιτήθηκε με αυτά που είχαν προβλεφθεί.

Πρέπει να επισημάνουμε ότι στην ανάπτυξη συστημάτων βασισμένα σε συστατικά ο αριθμός των γραμμών του κώδικα που έχει γραφτεί δεν είναι αξιόπιστος δείκτης προόδου του έργου. Ο κατασκευαστής μπορεί να έχει επαναχρησιμοποιήσει κατασκευασμένα κομμάτια όπως υποσυστήματα, κλάσεις και συστατικά, οπότε μια καλή πρόοδος μπορεί να υπάρχει, ενώ λίγος κώδικας έχει γραφτεί.

## 4 Η ΕΞΕΛΙΞΗ ΤΟΥ ΕΡΓΟΥ

Ένα κύριο χαρακτηριστικό της επαναληπτικής ανάπτυξης είναι ο τρόπος εξέλιξης του έργου. Αυτή η εξέλιξη έρχεται σε αντίθεση με την μέθοδο ανάπτυξης του καταρράκτη, κατά την οποία σκεφτόμαστε ότι πρώτα καταγράφονται οι απαιτήσεις, μετά η ανάλυση, μετά η σχεδίαση και μετά η υλοποίηση και η δοκιμή. Στην επαναληπτική ανάπτυξη τα μοντέλα αναπτύσσονται παράλληλα και σε αρμονία μεταξύ τους κατά τη διάρκεια των φάσεων. Σε αρχικές επαναλήψεις κάποια μοντέλα είναι πιο αναπτυγμένα από κάποια άλλα. Για παράδειγμα το μοντέλο των περιπτώσεων χρήσης είναι πιο ολοκληρωμένο από το μοντέλο της υλοποίησης. Αντί για ένα μοντέλο να αναπτύσσεται ανεξάρτητα από το επόμενο, φανταζόμαστε το όλο σύστημα να εξελίσσεται από μια κατάσταση σε μια πιο ολοκληρωμένη κατάσταση. Κάθε επανάληψη αναπαριστά μια βαθμιαία πρόοδο στην κατάσταση του όλου συστήματος έως τη σταδιακή του ολοκλήρωση.

### 4.1 ΣΥΛΛΗΨΗ

Ο σκοπός της φάσης της σύλληψης είναι να εκκινήσει το έργο. Πριν τη σύλληψη αυτό που υπάρχει είναι μόνο μια αξιόλογη ιδέα ή σκέψη. Μπορεί να είναι μια ιδέα για την κατασκευή ενός καινούργιου προϊόντος σε ένα καινούργιο περιβάλλον ή μια ιδέα για μια καινούργια έκδοση ενός υπάρχοντος προϊόντος λογισμικού. Η φάση της σύλληψης μπορεί να είναι τόσο απλή όπου ένα άτομο κάνει την παρουσίαση του τελικού προϊόντος, το σχεδιασμό μιας αρχιτεκτονικής μέσω διαγραμμάτων, και τη σύνθεση ενός επιχειρηματικού πλάνου. Άλλα μπορεί να είναι και πολύπλοκη όσο ένα ερευνητικό έργο. Το βασικό σημείο είναι ότι δεν βάζουμε τη σύλληψη σε ένα στερεότυπο. Μετά τη σύλληψη, ακόμα και αν το σύστημα είναι καινούργιο, έχει κατανοηθεί το πρόβλημα που προσπαθούμε να λύσουμε και έχουμε την αυτοπεποίθηση ότι θέλουμε και μπορούμε να αναπτύξουμε το σύστημα. Φυσικά η αυτοπεποίθηση δεν αφορά μόνο τους κατασκευαστές. Η εργασία που γίνεται σε αυτήν τη φάση παρέχει τη βεβαιότητα στους

πελάτες, στους επενδυτές και γενικά σε όλους τους εμπλεκόμενους ότι υπάρχει η δυνατότητα να αντιμετωπιστούν οι κρίσιμοι κίνδυνοι, να δημιουργηθεί μια υποψήφια αρχιτεκτονική και να γίνει ένα αρχικό επιχειρηματικό πλάνο.

#### **4.1.1 Η ΣΥΛΛΗΨΗ ΓΕΝΙΚΑ**

Ο στόχος της φάσης της σύλληψης είναι να δημιουργηθεί το επιχειρηματικό πλάνο στο βαθμό που απαιτείται για να δικαιολογήσει την κατασκευή του έργου. Για να το κάνουμε αυτό πρέπει να καθορίσουμε τα όρια του προτεινόμενου συστήματος. Πρέπει να ξέρουμε την εμβέλεια του έργου για να καταλάβουμε τι έχει να καλύψει η αρχιτεκτονική του. Πρέπει να ξέρουμε την εμβέλεια του για ξέρουμε μέσα σε ποια όρια θα εξερευνήσουμε τους κρίσιμους κινδύνους. Επίσης πρέπει να ξέρουμε την εμβέλεια του έργου για να θέσουμε τα όρια του κόστους και του χρονοδιαγράμματος. Προσπαθούμε να βρούμε ένα μέρος της αρχιτεκτονικής για να βεβαιωθούμε ότι υπάρχει αρχιτεκτονική που μπορεί να υποστηρίξει το προτεινόμενο σύστημα. Αυτό είναι που εννοούμε με τον όρο «υποψήφια αρχιτεκτονική». Θέλουμε επίσης να προλάβουμε τυχόν αποτυχία του έργου. Πολλά δύσκολα έργα απέτυχαν γιατί συνάντησαν κρίσιμους κινδύνους στις φάσεις της ολοκλήρωσης και ελέγχου του έργου και οι οποίοι δεν μπόρεσαν να αντιμετωπιστούν μέσα σε λογικά όρια υπέρβασης του προϋπολογισμού και χρονοδιαγράμματος. Είναι δεδομένο ότι υπάρχουν αβεβαιότητες και κίνδυνοι. Η μόνη διέξοδος είναι να αντιμετωπιστούν νωρίς. Προσπαθούμε να δημιουργήσουμε το αρχικό επιχειρηματικό πλάνο, να φανταστούμε το έργο με οικονομικούς όρους, δηλαδή αρχικές εκτιμήσεις κόστους, χρονοδιαγράμματα και κέρδη. Αναρωτιόμαστε τα εξής:

- Τα έσοδα από την πώληση ή χρήση του προϊόντος λογισμικού θα είναι περισσότερα από το κόστος της ανάπτυξης;
- Θα παραδοθεί στο εμπόριο ή για εσωτερική χρήση εγκαίρως ώστε να επιτευχθούν τα προσδοκώμενα κέρδη;

Αυτά είναι και τα θέματα που απασχολούν τη φάση της σύλληψης.

#### **4.1.2 ΠΡΙΝ ΤΗΝ ΕΝΑΡΞΗ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ**

Ακόμα και πριν τη φάση της σύλληψης, έχουμε κάποιες πληροφορίες για το τι πρόκειται να κάνουμε. Κάποιος είχε μια ιδέα και τη δικαιολόγησε έτσι ώστε να ξεκινήσει η όλη διαδικασία. Το μέγεθος της εργασίας πριν τη φάση της σύλληψης ποικίλει άλλα μπορούν να σημειωθούν τρία σημεία.

- Εταιρίες λογισμικού που παράγουν προϊόντα για γενική πώληση. Το πλήθος των πληροφοριών που έχουμε για να ξεκινήσουμε είναι σχεδόν πλήρες. Οι άνθρωποι του μάρκετινγκ και της διεύθυνσης της εταιρίας έχουν εξετάσει σοβαρά το προϊόν και στην ουσία έχουν κάνει κάποιες από τις εργασίες της σύλληψης.
- Τμήματα εταιριών που παράγουν συστήματα για άλλα τμήματα της ίδιας εταιρίας, δηλαδή συστήματα για εσωτερική χρήση. Γενικά συναντάμε δύο καταστάσεις. Στην πρώτη ένα τμήμα έχει ανάγκη για ένα σύστημα λογισμικού και ζητά από το τμήμα ανάπτυξης λογισμικού να το κατασκευάσει, κάνοντας μια περιγραφή του τι θ'έλει, χωρίς όμως να μπορεί να το εκφράσει με όρους

προγραμματιστικούς. Με άλλα λόγια το τμήμα της ανάπτυξης λογισμικού έχει λίγες πληροφορίες καθώς ξεκινάει η φάση της σύλληψης. Στη δεύτερη κατάσταση η διεύθυνση της εταιρίας νιώθει την ανάγκη ενός συστήματος για την εταιρία και κάποια ομάδα έχει κάνει κάποιες εργασίες και έχει βρει το τι θα υποστηρίζει. Σε μια τέτοια περίπτωση η φάση της σύλληψης θα ξεκινήσει με μια καλή κατανόηση των απαιτήσεων.

- Εταιρίες λογισμικού που παράγουν λογισμικό για πελάτες. Η αρχική πρόταση για το προτεινόμενο λογισμικό συχνά περιέχει αρκετές πληροφορίες και ίσως αρκετές σελίδες απαιτήσεων. Σε άλλες περιπτώσεις ο πελάτης μπορεί να έχει δώσει μόνο μια γενική εικόνα των αναγκών του.

Αν το προϊόν θα εξελιχθεί από μια προηγούμενη έκδοση, μέρος από την εργασία του σχεδιασμού της πρώτης επανάληψης του καινούργιου έργου θα έχει γίνει στην τελευταία επανάληψη του προηγούμενου έργου.

#### **4.1.2.1 ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ**

Στο ξεκίνημα ενός έργου τίθεται ένα δίλλημα. Είναι καλό να σχεδιάζουμε, αλλά δεν έχουμε αρκετές πληροφορίες για να βασιστεί το πλάνο μας. Αναγνωρίζουμε τι πρέπει να σχεδιαστεί όταν γνωρίζουμε όμως τι πρέπει να γίνει. Στην αρχή ξεκινάμε με τα εξής βήματα:

- Μαζεύουμε τις πληροφορίες που υπάρχουν πριν το έργο ξεκινήσει.
- Οργανώνουμε τις προηγούμενες πληροφορίες ώστε να γίνουν αξιοποιήσιμες.
- Μαζεύουμε άτομα που γνωρίζουν πώς να χρησιμοποιήσουν τις πληροφορίες.
- Να ξεκαθαρίσουμε τι λείπει, όχι σε σχέση με όλες τις φάσεις αλλά σε σχέση με την φάση της σύλληψης.

Με άλλα λόγια περιορίζουμε τις εργασίες μας στο τι χρειαζόμαστε για να επιτύχουμε τους στόχους αυτής της φάσης. Μετά κάνουμε ένα πλάνο για να ξεκαθαρίσουμε ποιες απαιτήσεις σχετίζονται με αυτούς τους αρχικούς στόχους και τις συσχετίζουμε με τις αντίστοιχες περιπτώσεις χρήσης. Σχεδιάζουμε να δημιουργήσουμε μία υποψήφια αρχιτεκτονική. Σχεδιάζουμε να εξελίξουμε αυτήν την αρχιτεκτονική μόνο έως ότου ξεκαθαρίσουμε ότι το έργο είναι εφικτό. Πρέπει να έχουμε υπόψη μας ότι το πλάνο είναι μεταβαλλόμενο. Όσο συλλέγουμε περισσότερες πληροφορίες αλλάζουμε το πλάνο για να ενσωματώσουμε τα ευρήματα μας.

#### **4.1.2.2 ΕΠΕΚΤΑΣΗ ΤΗΣ ΕΙΚΟΝΑΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ**

Στην αρχή του κύκλου ανάπτυξης του συστήματος η ομάδα ανάπτυξης μπορεί να μην έχει παραπάνω από μια σελίδα πληροφορίες για την εικόνα του συστήματος. Περιέχει μια λίστα χαρακτηριστικών, μερικές πληροφορίες απόδοσης, λίγες πληροφορίες για τους κινδύνους που οι κατασκευαστές μπορεί να αντιμετωπίσουν, ίσως μια ασαφή περιγραφή μιας πιθανής αρχιτεκτονικής και ένα στρογγυλωμένο νούμερο για τα οικονομικά που απαιτούνται. Σε μερικές περιπτώσεις έργων η φάση της σύλληψης μπορεί να διαρκέσει μόνο μια μέρα, ενώ σε περιπτώσεις όπως η κατασκευή ενός προϊόντος σε ένα καινούργιο περιβάλλον μπορεί να διαρκέσει και μήνες.

#### 4.1.2.3 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ

Όταν υπάρχουν αρκετές πληροφορίες για να δημιουργηθεί ένα πλάνο για την πρώτη επανάληψη, τότε δημιουργείται και το κριτήριο αξιολόγησης που θα υποδεικνύει ότι η επανάληψη θα έχει πετύχει τους στόχους της. Το πλάνο της πρώτης επανάληψης μπορεί να είναι λίγο γενικό λόγω της έλλειψης πληροφοριών, οπότε και το κριτήριο αξιολόγησης θα είναι γενικό. Όσο προχωράει η επανάληψη και ανακαλύπτονται περισσότερα, το κριτήριο αξιολόγησης γίνεται όλο και πιο συγκεκριμένο. Ωστόσο για τους τέσσερις στόχους της φάσης της σύλληψης υπάρχουν πέντε γενικά κριτήρια αξιολόγησης:

1. Οριοθέτηση του συστήματος. Η αρχική εικόνα, δεν υπάρχει αμφιβολία ότι μας παρείχε μια ιδέα για την εμβέλεια του συστήματος, αλλά δεν την έθεσε συγκεκριμένα. Στη φάση της σύλληψης σχεδιάζεται ένας κύκλος εντός του οποίου θα είναι αυτά που θα περιέχει το προτεινόμενο σύστημα και εκτός αυτά που δεν θα περιέχει. Μετά ορίζονται οι εξωτερικοί χρήστες, οι οποίοι μπορεί να είναι συστήματα ή άνθρωποι, με τους οποίους το σύστημα θα αλληλεπιδρά. Επίσης περιλαμβάνονται τα εξής:
  - Είναι ξεκάθαρο κάθε τι που θα είναι εντός του συστήματος;
  - Έχουν αναγνωριστεί όλοι οι χρήστες;
  - Έχει διατυπωθεί η γενική φύση των διασυνδέσεων του συστήματος με τους χρήστες;
  - Μπορεί κάθε τι που είναι εντός της εμβέλειας του συστήματος να σταθεί ως αυτόνομο σύστημα;
2. Καθορισμός των απαιτήσεων που απαιτούνται σε αυτήν τη φάση. Οι απαιτήσεις του συστήματος στην αρχή της φάσης της σύλληψης ποικίλουν από μια γενική εικόνα έως πολλές σελίδες κειμένου που τις περιγράφουν. Αυτές οι αρχικές απαιτήσεις θα περιέχουν πολλές αμφιβολίες. Οπότε πρέπει να μας απασχολήσουν τα εξής:
  - Έχουν αναγνωριστεί και περιγραφεί λεπτομερώς οι περιπτώσεις χρήσης, λειτουργικών και μη λειτουργικών, που χρειάζονται για να επιτευχθεί ο στόχος της φάσης;
  - Έχουν αναγνωριστεί και καταγραφεί απαιτήσεις στη λίστα με τις συμπληρωματικές απαιτήσεις;
3. Καθορισμός μιας υποψήφιας αρχιτεκτονικής. Η εμπειρία λέει ότι τα άτομα που δουλεύουν στη φάση της σύλληψης επικεντρώνονται γρήγορα σε καινούργιες λειτουργίες ή σε λειτουργίες που απαιτούν μεγάλη απόδοση. Μεταξύ αυτών πρέπει να επιλέξουν αυτές που θα επηρεάσουν την ανάπτυξη όλου του συστήματος. Για αυτές τις λίγες λειτουργίες πρέπει να επιλέξουν τουλάχιστον μια εφαρμόσιμη αρχιτεκτονική. Τα κριτήρια επιλογής είναι τα εξής:
  - Η υποψήφια αρχιτεκτονική ικανοποιεί τις ανάγκες των χρηστών;
  - Είναι σίγουρο ότι θα δουλέψει;

Για να απαντηθεί το τελευταίο ερώτημα πρέπει να λάβουμε υπόψη μας πολλά ζητήματα. Μπορεί να ενσωματώσει κατάλληλα τις τεχνολογίες που θα χρησιμοποιηθούν; Θα είναι αποτελεσματικό; Θα είναι αξιόπιστο και ανεκτικό σε



- λάθη; Θα είναι συμπαγές και προσαρμοστικό; Θα εξελιχθεί ομαλά καθώς θα προσθέτονται επιπλέον απαιτήσεις;
4. Αντιμετώπιση κρίσιμων κινδύνων. Κρίσιμοι κίνδυνοι είναι αυτοί που αν δεν αντιμετωπιστούν θα επηρεάσουν την επιτυχία του έργου. Θέματα που πρέπει να μας προβληματίσουν είναι τα εξής:
    - Έχουν αναγνωριστεί όλοι οι κρίσιμοι κίνδυνοι;
    - Έχουν αντιμετωπιστεί όλοι οι κρίσιμοι κίνδυνοι ή έχει γίνει έστω ένα σχέδιο αντιμετώπισης τους στο μέλλον;Με τον όρο αντιμετώπιση δεν εννοούμε αποκλειστικά την πλήρη εξάλειψη τους σε αυτήν τη φάση. Μπορεί να σημαίνει ότι ο πελάτης μπορεί να αλλάξει τις σχετιζόμενες με τον κίνδυνο απαιτήσεις του για να αποτραπεί μια τυχόν αποτυχία του έργου. Μπορεί να σημαίνει ότι η ομάδα ανάπτυξης έχει ένα γενικό σχέδιο αντιμετώπισης άλλα θα μπει σε λεπτομέρειες σε επόμενες φάσεις. Σε άλλες περιπτώσεις μπορεί να σημαίνει ότι υπάρχει κάποιο σχέδιο για την αντιμετώπιση των κινδύνων στο στάδιο της εμφάνισης τους, δηλαδή μείωση ή πλήρης εξάλειψη των συνεπειών των κινδύνων. Ο βασικός στόχος της φάσης της σύλληψης είναι να παράσχει την δυνατότητα στους εμπλεκόμενους στην ανάπτυξη του έργου να πάρουν μία καθαρά επιχειρηματική απόφαση για το έργο.
  5. Εκτίμηση της αξίας τους αρχικού επιχειρηματικού πλάνου. Το θέμα προς αξιολόγηση είναι αν το αρχικό επιχειρηματικό σχέδιο είναι αρκετά καλό για να δικαιολογήσει την συνέχεια του έργου. Σε ένα γνώριμο πεδίο δραστηριοποίησης του έργου το επιχειρηματικό σχέδιο θα είναι μάλλον σαφή ενώ σε ένα δύσκολο και καινούργιο πεδίο τα στοιχεία που θα έχει το επιχειρηματικό πλάνο δεν θα είναι τόσο σαφή.

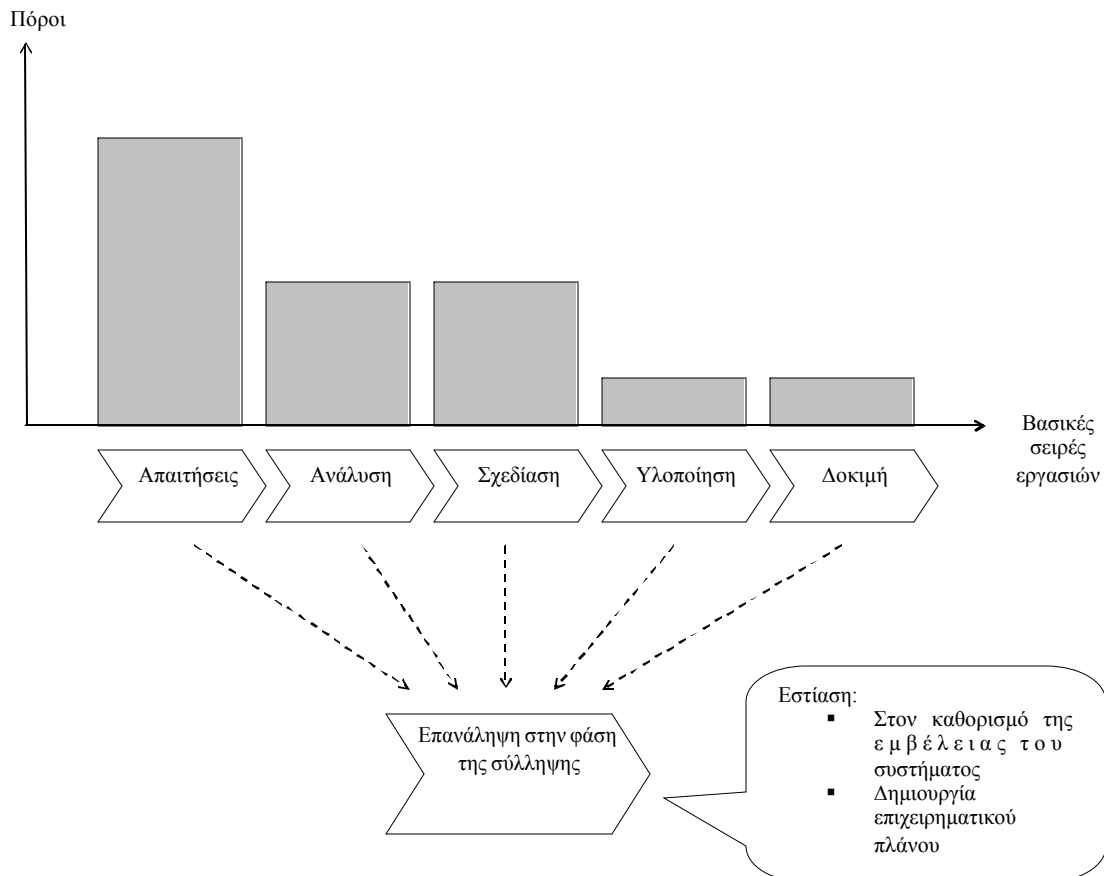
### **4.1.3 ΟΙ ΕΡΓΑΣΙΕΣ ΤΗΣ ΕΠΑΝΑΛΗΨΗΣ ΣΤΗ ΦΑΣΗ ΤΗΣ ΣΥΛΛΗΨΗΣ**

Στη σύλληψη διεξάγουμε τρεις ομάδες εργασιών. Η μία είναι ο προγραμματισμός των επαναλήψεων, η δεύτερη είναι οι πέντε βασικές σειρές εργασιών, και η τρίτη είναι η επιλογή του περιβάλλοντος ανάπτυξης που είναι κατάλληλο για το έργο. Επιπλέον, θα κάνουμε μια περιγραφή του αρχικού επιχειρηματικού πλάνου και την αποτίμηση της φάσης.

#### **4.1.3.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΠΕΝΤΕ ΒΑΣΙΚΕΣ ΣΕΙΡΕΣ ΕΡΓΑΣΙΩΝ**

Ο βασικός στόχος της φάσης της σύλληψης είναι να αποδείξει μια επιχειρηματική υπόθεση, την υπόθεση αν αξίζει από επιχειρηματικής άποψης να προχωρήσουμε με το έργο. Για να το αποδείξουμε αυτό πρέπει να βρούμε την εμβέλεια του προτεινόμενου συστήματος, να σχεδιάσουμε μια υποψήφια αρχιτεκτονική, να αναγνωρίσουμε τους κινδύνους που είναι κρίσιμοι για την επιτυχία του έργου και να κάνουμε ένα σχέδιο αντιμετώπισης τους. Αν έχουμε να κάνουμε με ένα καινούργιο τύπο συστήματος μπορεί να χρειάζεται να κάνουμε μια παρουσίαση της χρήσης του, κατασκευάζοντας ένα πρωτότυπο. Τα πρωτότυπα μπορούν επίσης να χρησιμοποιηθούν και για να την επίβλεψη και αντιμετώπιση των κινδύνων. Το περισσότερο μέρος των εργασιών κατά τη φάση της

σύλληψης πραγματοποιείται στην πρώτη σειρά εργασιών που είναι οι απαιτήσεις όπως φαίνεται και στην εικόνα 10, και στη συνέχεια να πραγματοποιείται ένα ποσοστό εργασιών στην ανάλυση και σχεδίαση. Το μέγεθος των τελευταίων σειρών εργασιών, της υλοποίησης και της δοκιμής είναι μικρό. Αυτή είναι η φάση κατά την οποία θα επιδείξουμε καινούργιες ιδέες και όχι κάποιο τυχόν πρωτότυπο να λειτουργεί σωστά με κάθε λεπτομέρεια.



Εικ. 10 Κατανομή εργασίας κατά τη φάση της σύλληψης

Γίνεται αναγνώριση των περιπτώσεων χρήσης και χρηστών του συστήματος που καθορίζουν την εμβέλεια του. Στη συνέχεια γίνεται η τοποθέτηση των περιπτώσεων χρήσης σε μια σειρά, επιλέγοντας αυτές που είναι χρήσιμες για την υποψήφια αρχιτεκτονική του συστήματος. Μετά γίνεται μια γενική περιγραφή της αρχικής αρχιτεκτονικής του συστήματος. Έπειτα συμπληρώνονται λεπτομερώς τα μονοπάτια των περιπτώσεων χρήσης του συστήματος, οι οποίες θα συμβάλουν στην εύρεση της εμβέλειας του συστήματος και της υποψήφιας αρχιτεκτονικής του και στην αναγνώριση των κρίσιμων κινδύνων, ώστε να είμαστε σε θέση να κάνουμε ένα αρχικό επιχειρηματικό

πλάνο. Ένα αποτέλεσμα αυτής της εργασίας είναι το μοντέλο περιπτώσεων χρήσης, αν το έργο είναι καινούργιο, ή ένα ανανεωμένο μοντέλο περιπτώσεων χρήσης αν το έργο είναι εξέλιξη κάποιου προηγούμενου έργου. Επίσης φτιάχνουμε και μια λίστα προτεραιότητας των περιπτώσεων χρήσης. Στην εργασία της ανάλυσης δημιουργούμε ένα μοντέλο ανάλυσης για τις περιπτώσεις χρήσης ή για τα σενάρια περιπτώσεων χρήσης που αναπτύξαμε στη σύλληψη. Το μοντέλο ανάλυσης είναι σημαντικό για να καταλάβουμε τις περιπτώσεις χρήσης. Είναι σημαντικό επίσης για να καταλάβουμε αυτά που αποτελούν τη βάση για την αρχική εικόνα της αρχιτεκτονικής. Αν το σύστημα είναι καινούργιο κατά τη φάση της σύλληψης μπορεί να δημιουργηθεί ένα διερευνητικό πρωτότυπο που αναπτύσσεται γρήγορα για να παρουσιάσει τις βασικές ιδέες. Αυτό το πρωτότυπο είναι για να παρουσιάσει τις ιδέες που περιλαμβάνονται από το σύστημα και όχι για να εξελιχθεί σε τελικό προϊόν. Με άλλα λόγια το πιο πιθανό είναι να είναι άχρηστο μετά το τέλος του έργου.

#### **4.1.3.2 ΕΥΡΕΣΗ ΚΡΙΣΙΜΩΝ ΚΙΝΔΥΝΩΝ**

Στη φάση της σύλληψης γίνονται οι ενέργειες μας για την αντιμετώπιση των κινδύνων ή για την δημιουργία ενός πλάνου αντιμετώπισης. Ένας κρίσιμος κίνδυνος είναι κάτι το οποίο μπορεί να κάνει το έργο να αποτύχει. Είναι υψίστης σημασίας να εντοπίζουμε τέτοιου μεγέθους κινδύνους στη φάση της σύλληψης. Αν εντοπίσουμε έναν τέτοιο κίνδυνο και δεν μπορούμε να τον αντιμετωπίσουμε ή να σχεδιάσουμε ένα πλάνο αντιμετώπισης του, τότε πρέπει να σκεφτούμε σοβαρά την εγκατάλειψη του έργου.

#### **4.1.3.3 Η ΕΚΤΕΛΕΣΗ ΤΩΝ ΣΕΙΡΩΝ ΕΡΓΑΣΙΩΝ**

Στα επόμενα θα δούμε τι έχουμε να κάνουμε σε αυτή τη φάση με μεγαλύτερη λεπτομέρεια. Θα θεωρήσουμε ότι έχουμε να αναπτύξουμε ένα καινούργιο προϊόν. Η εξέλιξη ενός υπάρχοντος προϊόντος είναι μια πιο εύκολη διαδικασία. Κάθε επανάληψη είναι μια μικρή μέθοδος καταρράκτη, που περνάει από όλες τις σειρές εργασιών από την καταγραφή των απαιτήσεων έως την δοκιμή.

##### **4.1.3.3.1 ΚΑΤΑΓΡΑΦΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ**

Στη φάση της σύλληψης η κύρια μας ενασχόληση είναι με τη πρώτη σειρά εργασιών, δηλαδή τις απαιτήσεις. Αυτή η σειρά εργασιών έχει να κάνει με την αναγνώριση και καταγραφή των περιπτώσεων χρήσης που είναι σχετικές με αυτήν τη φάση. Επιπλέον περιλαμβάνονται τα εξής:

- Καταγραφή των απαιτήσεων που είναι υποψήφιες για τη λίστα λειτουργιών του συστήματος.
- Κατανόηση του περιβάλλοντος του συστήματος.
- Καταγραφή των σχετικών λειτουργικών απαιτήσεων ως περιπτώσεις χρήσης.
- Καταγραφή των σχετικών μη λειτουργικών απαιτήσεων.

Λίστα των υποψήφιων απαιτήσεων. Τα χαρακτηριστικά του προτεινόμενου συστήματος διαμορφώνονται συνήθως από την εμπειρία των πελατών ή των χρηστών από τη χρήση προηγούμενων παρόμοιων συστημάτων. Στην περίπτωση προϊόντων με μεγάλες πωλήσεις, καινούργια χαρακτηριστικά προκύπτουν από τις ανάγκες της αγοράς. Επιπλέον και οι κατασκευαστές λογισμικού συλλαμβάνουν από μόνοι τους καινούργιες υποψήφιες ιδέες για το σύστημα. Κάποια χαρακτηριστικά προέρχονται από την αλληλεπίδραση μεταξύ των ομάδων ανάπτυξης λογισμικού και χρηστών και άλλα προέρχονται από τις ανάγκες της αγοράς. Χαρακτηριστικά από όλες αυτές τις κατηγορίες αποτελούν πηγή υποψήφιων απαιτήσεων για την λίστα χαρακτηριστικών του συστήματος.

Κατανόηση του περιβάλλοντος του συστήματος. Αν ο πελάτης έχει ένα επιχειρηματικό μοντέλο, τότε τα άτομα που εργάζονται στη φάση της σύλληψης μπορούν να εργαστούν με αυτό. Αν ο πελάτης δεν διαθέτει τέτοιο μοντέλο, τότε πρέπει να ενθαρρύνουμε τον πελάτη να το αναπτύξει, παρόλο που αυτό θα απαιτήσει χρόνο και πόρους.

Καταγραφή των λειτουργικών απαιτήσεων. Οι λειτουργικές απαιτήσεις θα καταγραφούν και θα εκφραστούν ως περιπτώσεις χρήσης.

Καταγραφή μη λειτουργικών απαιτήσεων. Οι μη λειτουργικές απαιτήσεις που είναι συγκεκριμένες για κάποια περίπτωση χρήσης προσαρτώνται σε εκείνη την περίπτωση χρήσης. Εκείνες που είναι πιο γενικές και είναι λιγότερες σε αριθμό, καταγράφονται στη λίστα συμπληρωματικών απαιτήσεων. Κάποιες από τις τελευταίες μπορεί να είναι πολύ σημαντικές για την επιλογή της αρχιτεκτονικής.

Εύρεση χρηστών και περιπτώσεων χρήσης. Ολοκληρωμένες περιπτώσεις χρήσης και ένα πλήρες μοντέλο περιπτώσεων χρήσης είναι πέρα από τα όρια της φάσης της σύλληψης. Οι κατασκευαστές σε αυτή τη φάση πρέπει πρώτα να βάλουν σε μια σειρά το υποσύνολο των περιπτώσεων χρήσης που χρειάζονται για την ολοκλήρωση αυτής της φάσης και έπειτα να τις συμπληρώσουν με λεπτομέρειες. Το πρόβλημα στη φάση της σύλληψης είναι να περιορίσουμε τις εργασίες μας κυρίως σε αυτές τις περιπτώσεις χρήσης που θα μας βοηθήσουν να πετύχουμε τους στόχους αυτής της φάσης. Πρέπει να αγνοήσουμε μονοπάτια περιπτώσεων χρήσης ή περιπτώσεις χρήσης που δεν είναι σχετικές με την εμβέλεια του συστήματος, την υποψήφια αρχιτεκτονική, που δεν εμπεριέχουν κανένα κρίσιμο κίνδυνο ή έχουν μικρή σημασία για τον σχεδιασμό του αρχικού επιχειρηματικού πλάνου.

Οι περιπτώσεις χρήσης σε προτεραιότητα. Στη συνέχεια δημιουργείται το μοντέλο περιπτώσεων χρήσης που θα βοηθήσει στην δημιουργία του πλάνου των επαναλήψεων. Αυτή η εργασία προχωράει παράλληλα με τις βασικές σειρές εργασιών σε κάθε φάση. Το πλάνο των επαναλήψεων υποδεικνύει τη σειρά προτεραιότητας των περιπτώσεων χρήσης που σχετίζονται με αυτήν τη φάση.

Καταγραφή των λεπτομερειών περίπτωσης χρήσης. Την ίδια στιγμή που δεχόμαστε τον περιορισμό των εργασιών σε συγκεκριμένες περιπτώσεις χρήσης, πρέπει να επισημάνουμε την σημασία της ολοκλήρωσης όλων των μονοπατιών των περιπτώσεων

χρήσης που σχετίζονται με τη φάση της σύλληψης, δηλαδή αυτές που απαιτούνται για τον καθορισμό της εμβέλειας του συστήματος, το σχεδιασμό της αντιμετώπισης των κρίσιμων κινδύνων, και την εύρεση της υποψήφιας αρχιτεκτονικής. Σχεδόν σε όλες τις περιπτώσεις οι κατασκευαστές πιστεύουν ότι καταλαβαίνουν ποιες απαιτήσεις χρειάζονται, αλλά στην πραγματικότητα έχουν παραλείψει κάποιες σημαντικές. Επίσης έχουν την τάση να πιστεύουν ότι τα οφέλη από τη λεπτομερή συμπλήρωση των απαιτούμενων περιπτώσεων χρήσης δεν αξίζουν το κόστος, ενώ αντιθέτως αυτό που ισχύει είναι να καταγράφουμε κάθε λεπτομέρεια που χρειάζεται. Το να ξέρουμε προς τα πού οδηγούμαστε είναι ένας στόχος. Δεν σημαίνει ότι έχουμε εργαστεί με ένα τεράστιο μέρος περιπτώσεων χρήσης. Συνήθως συμπληρώνουμε λεπτομερώς περίπου το 10% των περιπτώσεων χρήσης. Αυτό που θέλουμε να δείξουμε είναι ότι ενώ μπορεί να έχουμε κοιτάξει πολλές περιπτώσεις χρήσης, έχουμε συμπληρώσει λεπτομερώς μόνο ένα μέρος αυτών, αυτών που είναι σχετικές με αυτή τη φάση. Για παράδειγμα αν έχουμε επιλέξει το 50% όλων των πιθανών περιπτώσεων χρήσης ως σχετικές με αυτή τη φάση και βρίσκουμε ότι για κάθε μία από αυτές τι περιπτώσεις χρήσης χρειαζόμαστε μόνο το 20% των σεναρίων της που πρέπει να συμπληρωθούν λεπτομερώς, τότε στην ουσία έχουμε εργαστεί στο 10% όλων των περιπτώσεων χρήσης. Το όλο θέμα είναι να κρατήσουμε το κόστος και το χρόνο αυτής της φάσης όσο πιο χαμηλά. Τα ακριβή ποσοστά κάθε ξεχωριστού έργου ποικίλουν ανάλογα με το βαθμό δυσκολίας του.

#### 4.1.3.3.2 ΑΝΑΛΥΣΗ

Οι στόχοι της εργασίας της ανάλυσης γενικά είναι η ανάλυση των απαιτήσεων, η βελτίωση και η σύνθεση τους ως ένα μοντέλο αντικειμένων που παίζει το ρόλο μιας πρώτης εικόνας του μοντέλου σχεδίασης. Το αποτέλεσμα αυτής της φάσης είναι ένα μοντέλο ανάλυσης. Χρησιμοποιούμε αυτό το μοντέλο ανάλυσης για να καθορίσουμε ακριβώς τις περιπτώσεις χρήσης και να μας οδηγήσει στη σωστή υποψήφια αρχιτεκτονική. Μια πιο ολοκληρωμένη εικόνα της αρχιτεκτονικής θα έχουμε στη φάση της επεξεργασίας. Αυτό σημαίνει ότι πολύ μικρό μέρος του μοντέλου ανάλυσης συμπληρώνεται στη φάση της σύλληψης, περίπου 5%. Συγκεκριμένα αυτό που θα περιγράψουμε θα είναι μια γενική εικόνα του μοντέλου ανάλυσης.

Αρχιτεκτονική ανάλυση. Το έργο μας στη φάση της σύλληψης είναι να βάλουμε σε μια σειρά τις περιπτώσεις χρήσης ή τα μονοπάτια που χρειάζεται να προσέξουμε σε αυτή τη φάση και κυρίως να τις βελτιώσουμε και να τις κατανοήσουμε. Με δεδομένο αυτό το υποσύνολο των περιπτώσεων χρήσης κατασκευάζουμε μια πρώτη μορφή του μοντέλου ανάλυσης για εκείνα τα μέρη. Δεν χρειάζεται να είναι εκτενές και τέλειο, αλλά απλά θα έχει ένα καθοδηγητικό ρόλο.

Ανάλυση περίπτωσης χρήσης: Σε κάποιες περιπτώσεις, το να μελετούμε κάθε περίπτωση χρήσης μία προς μία δεν είναι αρκετό. Δηλαδή το μοντέλο περιπτώσεων χρήσης ασχολείται με μία περίπτωση χρήσης κάθε φορά. Η πραγματικότητα όμως είναι διαφορετική, γιατί οι περιπτώσεις χρήσης μοιράζονται πόρους, όπως βάσεις δεδομένων και υπολογιστικούς πόρους, μέσα στο σύστημα. Το μοντέλο ανάλυσης είναι αυτό που αποκαλύπτει αυτές τις μοιρασιές των πόρων. Οπότε, συχνά μπορεί να χρειάζεται να

εμβαθύνουμε την ανάλυση για να λύσουμε αυτές συγκρούσεις. Στη φάση της σύλληψης μπορεί να χρειαστεί να αναλύσουμε και να βελτιώσουμε κάποιες από τις περιπτώσεις χρήσης, περίπου το 10% του συνόλου.

Ανάλυση κλάσης και ανάλυση πακέτου. Αν γίνουν σε αυτήν τη φάση τότε γίνονται σε πολύ μικρό βαθμό.

#### **4.1.3.3.3 ΣΧΕΔΙΑΣΗ**

Ο βασικός στόχος της εργασίας της σχεδίασης σε αυτή τη φάση είναι να σχεδιάσουμε ένα μοντέλο σχεδίασης για την υποψήφια αρχιτεκτονική και να το περιλάβουμε στην αρχική αρχιτεκτονική περιγραφή.

Αρχιτεκτονική σχεδίαση: Ο σκοπός της εργασίας της σχεδίασης είναι να αναπτύξουμε μια αρχική περιγραφή του μοντέλου σχεδίασης, που θα αποτελέσει το πρώτο βήμα για να αποκτήσουμε από αρχιτεκτονικής πλευράς μια εικόνα του μοντέλου σχεδίασης που θα πραγματοποιεί τις περιπτώσεις χρήσης, που αναγνωρίσαμε στην εργασία των απαιτήσεων, με αλληλεπιδράσεις μεταξύ των υποσυστημάτων ή των κλάσεων. Το μοντέλο σχεδίασης δεν πραγματοποιεί μόνο τις λειτουργικές απαιτήσεις που προσδιορίζονται από τις περιπτώσεις χρήσης, αλλά και τις μη λειτουργικές, όπως η απόδοση. Αν το σύστημα θα κατανεμηθεί σε κόμβους τότε θα σχεδιαστεί και ένα μικρής κλίμακας μοντέλο διάταξης.

Σχεδίαση περίπτωσης χρήσης: Στη φάση της σύλληψης η εργασία που γίνεται επάνω στη σχεδίαση περιπτώσεων χρήσης είναι ελάχιστη.

Σχεδίαση κλάσης και σχεδίαση υποσυστήματος: Ομοίως αν είναι να πραγματοποιηθεί τέτοια εργασία σε αυτή τη φάση τότε το μέγεθος της είναι πολύ μικρό.

#### **4.1.3.3.4 ΥΛΟΠΟΙΗΣΗ**

Σε αυτό το σημείο πρέπει να αποφασίσουμε αν η φάση της σύλληψης θα σταματήσει στο σημείο της περιγραφής της υποψήφιας αρχιτεκτονικής ή θα προχωρήσει και στην εργασία της υλοποίησης. Από τη μια πλευρά κάποιο από το περιεχόμενο του συστήματος δεν μπορούμε να είμαστε σίγουροι ότι θα δουλέψει, αν δεν το δούμε να δουλεύει σε ένα πρωτότυπο, όπως και επίσης δεν μπορούμε να είμαστε σίγουροι και για έναν κίνδυνο ότι έχει αντιμετωπιστεί, αν δεν δούμε το σχετικό σημείο του πρωτότυπου να δουλεύει. Από την άλλη πλευρά για να κρατήσουμε χαμηλά το κόστος σε ανθρώπινο δυναμικό και χρόνο περιορίζουμε τη φάση της σύλληψης έως το σημείο όπου έχουμε μια υποψήφια αρχιτεκτονική που το πιθανότερο είναι να δουλέψει. Στη συνήθη περίπτωση η φάση της σύλληψης θα σταματήσει στο σημείο εύρεσης της υποψήφιας αρχιτεκτονικής, οπότε περαιτέρω συνέχεια με την εργασία της υλοποίησης δεν χρειάζεται. Παρόλα αυτά στην περίπτωση που είναι αναγκαία η επίδειξη λειτουργίας ενός πρωτότυπου η σύλληψη θα προχωρήσει στην εργασία της υλοποίησης που θα είναι όμως μικρή.

#### **4.1.3.3.5 ΔΟΚΙΜΗ**

Καθώς προχωράνε οι εργασίες της ανάλυσης, σχεδίασης και υλοποίησης, εξοικειωνόμαστε όλο και περισσότερο με τη γενική φύση του έργου και σκεφτόμαστε τι είδους έλεγχος θα χρειαστεί να πραγματοποιήσουμε και κάνουμε μερικά μη ολοκληρωμένα πλάνα. Αλλά κανένα σημαντικό μέγεθος ελέγχου δεν πραγματοποιείται κατά τη φάση της σύλληψης, γιατί το πρωτότυπο είναι περισσότερο διευκρινιστικό παρά λειτουργικό. Ακόμα και έτσι αν είναι, ένα μικρό μέγεθος δοκιμών είναι χρήσιμο.

#### **4.1.3.4 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΠΛΑΝΟΥ**

Όταν έχουμε βρει προς το τέλος της φάσης της σύλληψης ότι το έργο έχει μια υποψήφια αρχιτεκτονική και μπορεί να ξεπεράσει τους κρίσιμους κινδύνους τότε είναι η ώρα να σχηματίσουμε την εικόνα του συστήματος από οικονομικής πλευράς λαμβάνοντας υπόψη τις απαιτήσεις σε πόρους και την εσωτερική αποδοχή ή αποδοχή στην αγορά. Από τη μια πλευρά του επιχειρηματικού πλάνου θα είναι το κόστος του έργου και από την άλλη τα οφέλη από τυχόν χρήση του έργου.

#### **4.1.3.5 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ**

Κοντά στο ξεκίνημα της φάσης της σύλληψης, όταν επαρκείς πληροφορίες γίνουν διαθέσιμες, ορίζουμε τα κριτήρια με τα οποία θα κάνουμε την αποτίμηση της πρώτης επανάληψης και όλης της φάσης. Στην εργασία της αποτίμησης πρέπει να υπάρχει και ένας εκπρόσωπος των πελατών ή χρηστών. Κάποια από τα κριτήρια μπορεί να αποδειχθεί ότι δεν ήταν επιτεύξιμα σύμφωνα με το αρχικό πλάνο. Παραδείγματα τέτοιων κριτηρίων είναι τα εξής:

- Επέκταση του μοντέλου περιπτώσεων χρήσης ως το σημείο που είναι απαραίτητο από αυτή τη φάση.
- Δημιουργία ενός διερευνητικού πρωτότυπου ως το σημείο που είναι δυνατή η επίδειξη του.
- Υποψία ότι όλοι οι κρίσιμοι κίνδυνοι δεν έχουν βρεθεί.
- Το γεγονός ότι οι καταγεγραμμένοι κρίσιμοι κίνδυνοι δεν έχουν αντιμετωπιστεί αποτελεσματικά ή δεν έχει προβλεφθεί η αντιμετώπιση τους από ένα σχέδιο.

Τα κριτήρια που δεν έχουν επιτευχθεί μεταφέρονται σε μετέπειτα επαναλήψεις και μεταβάλλεται ο προγραμματισμός τους και το χρονοδιάγραμμα τους ανάλογα. Ένα σημαντικό αποτέλεσμα της αποτίμησης της φάσης της σύλληψης είναι η βασική απόφαση να προχωρήσουμε με το έργο ή όχι. Εξετάζουμε τους στόχους αυτής της φάσης όπως η εμβέλεια, οι κρίσιμοι κίνδυνοι και η υποψήφια αρχιτεκτονική και αποφασίζουμε είτε να προχωρήσουμε είτε να σταματήσουμε. Μπορεί να χρειαστεί να περιμένουμε μέχρι το πρώτο σημείο ορόσημο για να αποφασίσουμε. Αλλά πρέπει να σταματήσουμε από την πρώτη στιγμή που υπάρχουν γεγονότα που δικαιολογούν μια τέτοια απόφαση, γιατί δεν υπάρχει λόγος να ξοδεύουμε περισσότερη προσπάθεια. Η απόφαση αυτή πρέπει

να παρθεί από όλους τους εμπλεκομένους, από τους επενδυτές έως τους χρήστες, γιατί σε περίπτωση ματαίωσης οι πελάτες μπορεί να προτείνουν ένα εναλλακτικό δρόμο για την παράκαμψη των εμποδίων.

#### **4.1.4 ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

Προς το τέλος της φάσης της σύλληψης, καθώς αρχίζει να μας ενδιαφέρει το κόστος και το χρονοδιάγραμμα της φάσης της επεξεργασίας, ξεκινάμε να φτιάχνουμε το πλάνο για αυτή τη φάση. Κατά αυτή τη φάση θέλουμε να επεξεργαστούμε το 80% περίπου των απαιτήσεων και δεν θέλουμε να αγνοήσουμε τίποτα σημαντικό σε σχέση με την αρχιτεκτονική του συστήματος. Χρειάζεται να τα κάνουμε αυτά για να είμαστε σε θέση να κάνουμε ένα πιο ακριβή προϋπολογισμό για το έργο, σε αντίθεση με τη φάση της σύλληψης όπου είχαμε λίγες πληροφορίες, και για να είμαστε σε θέση να επιλέξουμε την αρχιτεκτονική. 80% είναι περίπου το ποσοστό του συνολικού όγκου των περιπτώσεων χρήσης που συνήθως χρειάζεται για να μπορούμε να κάνουμε ένα οικονομικό προϋπολογισμό. Από αυτό το 80% μπορεί να χρειάζεται να αναλύσουμε το 50% των απαιτήσεων για να κατανοήσουμε τις απαιτήσεις καλά. Επίσης για να βρούμε το βασικό κορμό της αρχιτεκτονικής μπορεί να χρειαστούμε έως και το 80% για να είμαστε σίγουροι ότι δεν αφήσαμε τίποτα από έξω. Από αυτό το 80% επιλέγουμε το σημαντικό μέρος των περιπτώσεων χρήσης πάνω στις οποίες θα στηριχτούμε για να σχεδιάζουμε το κορμό της αρχιτεκτονικής μας. Αυτές οι περιπτώσεις χρήσης είναι περίπου το 40% των περιπτώσεων χρήσης και περίπου το 40% κάθε μιας κατά μέσο όρο. Σε αυτό το δρόμο κινούμαστε για τις επαναλήψεις της φάσης της επεξεργασίας. Αποφασίζουμε τι είναι κατάλληλο για κάθε επανάληψη, ποιες απαιτήσεις να υλοποιήσουμε και να ελέγξουμε και ποιους κινδύνους να αντιμετωπίσουμε. Η εμπειρία δείχνει ότι όποια υλοποίηση είχαμε στη φάση της σύλληψης, όπως για παράδειγμα το διερευνητικό πρωτότυπο, δεν θα είναι κατάλληλο για να συνεχιστεί η κατασκευή πάνω σε αυτό. Για να είναι πιο κατανοητά τα ποσοστά μπορούμε να δούμε την εικόνα 11.



	Επιχειρηματικό πλάνο	Αναγνωρισμένες περιπτώσεις χρήσης	Συνολικός όγκος περιπτώσεων χρήσης που έχουν περιγραφεί	Συνολικός όγκος περιπτώσεων χρήσης που έχουν αναλυθεί	Συνολικός όγκος περιπτώσεων χρήσης που έχουν σχεδιαστεί, υλοποιηθεί και ελεγχθεί
Σύλληψη	50%-70%	50%	10%	5%	Μικρό ποσοστό για το πρωτότυπο
Επεξεργασία	Σχεδόν 100%	>80%	40%-80%	20%-40%	<10%
Υλοποίηση	100%	100%	100%	100%	100%
Μετάβαση					

Εικ. 11 Ενδεικτικά ποσοστά εργασιών κατά την διάρκεια των φάσεων της ενοποιημένης διεργασίας

#### 4.1.5 ΤΑ ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΣΥΛΛΗΨΗΣ

Η φάση της σύλληψης παράγει τα εξής παρακάτω:

- Μία λίστα χαρακτηριστικών του συστήματος.
- Μία πρώτη έκδοση του επιχειρηματικού πεδίου που περιγράφει το περιβάλλον του συστήματος.
- Μια πρώτη μορφή του μοντέλου περιπτώσεων χρήσης, του μοντέλου ανάλυσης και του μοντέλου σχεδίασης. Για τα μοντέλα της υλοποίησης και των δοκιμών μπορεί να υπάρχουν κάποια ελάχιστα στοιχεία. Επίσης υπάρχει και μια πρώτη μορφή της λίστας συμπληρωματικών απαιτήσεων.
- Ένα πρώτο γράφημα της υποψήφιας αρχιτεκτονικής.
- Πιθανώς διερευνητικό πρωτότυπο που επιδεικνύει τη χρήση του καινούργιου συστήματος.
- Μια αρχική λίστα κινδύνων και μια λίστα κατάταξης των περιπτώσεων χρήσης.
- Ένα αρχικό γενικό πλάνο για τις φάσεις.
- Μια αρχική περιγραφή του επιχειρηματικού πλάνου που περιλαμβάνει το επιχειρηματικό περιβάλλον και τα κριτήρια επιτυχίας.

Όλοι οι εμπλεκόμενοι τώρα πρέπει να έχουν κατανοήσει την εικόνα του συστήματος. Οι περιπτώσεις χρήσης έχουν μπει σε μια σειρά, και υπάρχουν αρκετές πληροφορίες για να

σχεδιαστεί η επόμενη φάση. Τα αποτελέσματα αυτής της φάσης εξελίσσονται στην φάση της επεξεργασίας.

## 4.2 ΕΠΕΞΕΡΓΑΣΙΑ

Καθώς εισερχόμαστε στη φάση της επεξεργασίας περνάμε από ένα σημείο ορόσημο που σηματοδοτεί τρία επιτεύγματα που είναι τα εξής:

- Έχουμε σχηματίσει μια αρχική αρχιτεκτονική, την υποψήφια αρχιτεκτονική, που δηλώνει ότι ξέρουμε πώς να κατασκευάσουμε για το προτεινόμενο σύστημα μια αρχιτεκτονική που να περικλείει όλα τα καινούργια και δύσκολα μέρη του.
- Έχουμε αναγνωρίσει του σημαντικούς κινδύνους, δηλαδή τους κρίσιμους κινδύνους, τους οποίους εξερευνήσαμε ως το σημείο που νιώθουμε σίγουροι ότι η κατασκευή του συστήματος είναι εφικτή.
- Έχουμε δημιουργήσει ένα αρχικό επιχειρηματικό πλάνο με αρκετές πληροφορίες ώστε να προχωρήσουμε στη δεύτερη φάση και έχουμε πάρει και την συγκατάθεση όλων των εμπλεκομένων στο έργο και κυρίως των επενδυτών.

### 4.2.1 Η ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΕΝΙΚΑ

Οι κύριοι στόχοι μας είναι οι εξής:

- Η καταγραφή των περισσότερων εναπομεινάντων απαιτήσεων και η διατύπωση τους ως περιπτώσεις χρήσης.
- Η δημιουργία ενός σταθερού κορμού αρχιτεκτονικής που θα καθοδηγήσει τις εργασίες των φάσεων της κατασκευής και μετάβασης και να εκτείνει την καθοδήγηση και σε μελλοντικές εκδόσεις.
- Η συνέχεια της επίβλεψης των υπόλοιπων κρίσιμων κινδύνων και η αναγνώριση σημαντικών κινδύνων ως του σημείου που μπορούμε να προβλέψουμε τις συνέπειες τους στο επιχειρηματικό πλάνο.
- Η συμπλήρωση με περισσότερες λεπτομέρειες του πλάνου του έργου.

Για να πετύχουμε αυτούς τους στόχους κοιτάμε όλο το έργο και δεν προχωράμε σε βάθος (“a mile wide and inch deep”). Σε κάποιες περιπτώσεις όπου οι τεχνικοί κίνδυνοι κυριαρχούν ή είναι οι πιο σημαντικοί κίνδυνοι, τότε ίσως χρειαστεί να εξερευνήσουμε σε μεγαλύτερο βάθος για να δημιουργήσουμε μια γερή αρχιτεκτονική. Σε ένα μεγάλο έργο συνεπώς μπορεί να χρειαστεί παράλληλα να δουλέψουμε και σε βάθος στα ιδιαίτερα σημεία του έργου. Παίρνουμε αποφάσεις αρχιτεκτονικής φύσεως στη βάση της κατανόησης του όλου συστήματος, δηλαδή της εμβέλειας του και των λειτουργικών και μη λειτουργικών απαιτήσεων, όπως η απόδοση. Εκτός αυτού πρέπει να εξισορροπήσουμε τις απαιτήσεις, όπως εκφράστηκαν με τις περιπτώσεις χρήσης και το μοντέλο περιπτώσεων χρήσης, σε σχέση με την αρχιτεκτονική. Αυτά τα δύο αναπτύσσονται σε συνεργασία μεταξύ τους και επηρεάζει το ένα το άλλο. Η βασική εστίαση στη φάση της επεξεργασίας είναι η δημιουργία ενός κορμού αρχιτεκτονικής. Αυτή η εστίαση περιλαμβάνει την έρευνα του 80% περίπου των περιπτώσεων χρήσης και την αναγνώριση των κινδύνων που παρεμβαίνουν στην επίτευξη του στόχου. Προς το τέλος αυτής της φάσης θα έχουμε συγκεντρώσει όλες τις απαραίτητες πληροφορίες για

να κάνουμε το πλάνο της φάσης της κατασκευής. Επιπλέον θα έχουμε αρκετές πληροφορίες για να κάνουμε ένα αξιόπιστο επιχειρηματικό πλάνο, εργασία που ξεκινήσαμε στη φάση της σύλληψης.

#### **4.2.2 ΑΡΧΙΚΑ ΣΤΗ ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

Στο ξεκίνημα της φάσης της επεξεργασίας έχουμε από τη φάση της σύλληψης ένα πλάνο για τη φάση της επεξεργασίας, ένα μερικώς συμπληρωμένο μοντέλο περιπτώσεων χρήσης και μία υποψήφια αρχιτεκτονική. Μπορεί επίσης να έχουμε τα αρχικά σχέδια των μοντέλων ανάλυσης και σχεδίασης. Όμως δεν μπορούμε να στηριχτούμε στην επαναχρησιμοποίηση αυτών των μοντέλων, παρόλο που μας παρέχουν μία καθοδήγηση. Συγκεκριμένα μία από τις εργασίες μας στη φάση της επεξεργασίας είναι να συμπληρώσουμε αυτά τα μοντέλα, άλλα και πάλι όχι εντελώς, άλλα στο βαθμό που κρίνεται απαραίτητο για να βρούμε τον κορμό αρχιτεκτονικής. Μπορεί επίσης να έχουμε και ένα πρωτότυπο για να επιδείξουμε τη χρήση του συστήματος. Άλλα δεν μπορούμε να περιμένουμε να εξελίξουμε την κατασκευή μας πάνω σε αυτό το πρωτότυπο. Είναι, συνήθως, κατασκευασμένο κατά τον πιο γρήγορο τρόπο για να αποδείξει ότι το έργο είναι εφικτό να κατασκευαστεί και όχι να μας παράσχει τη βάση πάνω στην οποία θα κατασκευάσουμε το έργο μας στις επόμενες φάσεις.

#### **4.2.3 ΣΧΕΔΙΑΣΗ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

Το πλάνο αυτής της φάσης που έγινε στο τέλος της φάσης της σύλληψης μπορεί να μην είναι ολοκληρωμένο. Συχνά σε μεγάλα έργα οι πόροι που θα είναι διαθέσιμοι για τη φάση της επεξεργασίας δεν είναι γνωστοί μέχρι το σημείο όπου ξεκινάει η φάση. Οπότε με μια πιο ενημερωμένη γνώση των πόρων και του χρονοδιαγράμματος του έργου γίνεται μια ενημέρωση και των πλάνων των επαναλήψεων και της φάσης.

#### **4.2.4 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ**

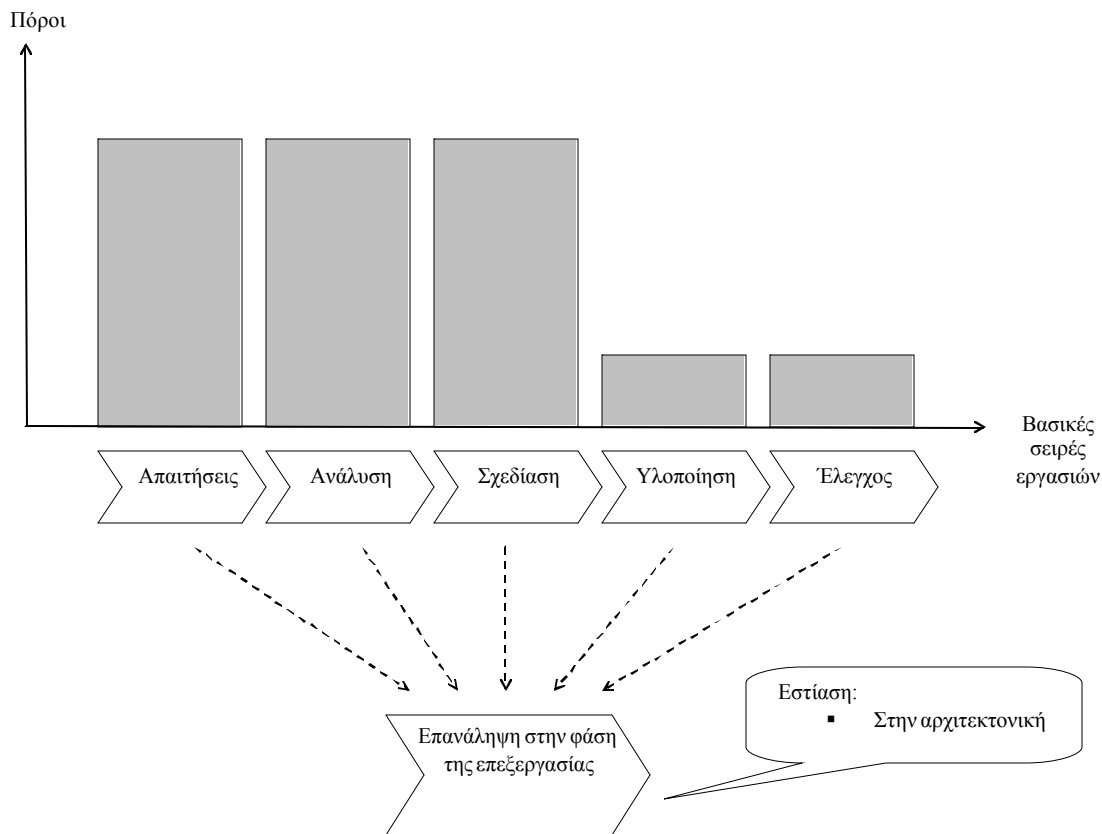
Τα συγκεκριμένα κριτήρια που πρέπει να επιτευχθούν από μία επανάληψη ή από την φάση της επεξεργασίας σαν σύνολο είναι μοναδικά για κάθε έργο. Αλλά μπορούμε να δούμε τα αποτελέσματα σε σχέση με τους στόχους αυτής της φάσης:

- Επέκταση των απαιτήσεων. Τα κριτήρια είναι τα εξής:
  - Έχουν αναγνωριστεί οι απαιτήσεις, οι χρήστες, και οι περιπτώσεις χρήσης που χρειάζονται για να σχεδιαστεί ο κορμός της αρχιτεκτονικής, για να αναγνωριστούν οι σημαντικοί κίνδυνοι, και να υποστηριχθεί το επιχειρηματικό πλάνο;
  - Έχουν συμπληρωθεί οι απαιτήσεις με τις απαιτούμενες λεπτομέρειες για την επίτευξη των στόχων αυτής της φάσης;
- Δημιουργία του κορμού της αρχιτεκτονικής. Τα κριτήρια είναι τα εξής:

- Ο εκτελέσιμος κορμός της αρχιτεκτονικής ικανοποιεί τις απαιτήσεις που έχουν καταγραφεί έως τώρα;
- Ο βασικός κορμός της αρχιτεκτονικής είναι αρκετά γερός για να αντέξει τη φάση της κατασκευής και τυχόν προσθήκη επιπλέον χαρακτηριστικών σε μετέπειτα εκδόσεις;
- Αντιμετώπιση σημαντικών κινδύνων. Τα κριτήρια αξιολόγησης είναι τα εξής:
  - Έχουν αντιμετωπιστεί επαρκώς οι κρίσιμοι κίνδυνοι, δηλαδή να έχουν εξαλειφθεί εντελώς ή να υπάρχει ένα ικανοποιητικό σχέδιο αντιμετώπισης τους;
  - Έχουν αναγνωριστεί όλοι οι σημαντικοί κίνδυνοι;
  - Έχουν εξεταστεί όλοι οι σημαντικοί κίνδυνοι ώστε να είναι προβλέψιμοι;
  - Είναι, οι υπόλοιποι κίνδυνοι στη λίστα κινδύνων, εύκολοι στην αντιμετώπιση τους στη φάση της κατασκευής;
- Εκτίμηση της αξίας του επιχειρηματικού πλάνου. Τα κριτήρια είναι τα εξής:
  - Είναι το έργο αρκετά προσδιορισμένο ως προς τον οικονομικό του προϋπολογισμό, το χρονοδιάγραμμα και την ποιότητα του;
  - Σύμφωνα με το επιχειρηματικό πλάνο το έργο θα μας αποφέρει οφέλη;
  - Εν ολίγης είμαστε έτοιμοι να κάνουμε ένα συμβόλαιο με τον πελάτη ή κάτι αντίστοιχο σε περίπτωση εσωτερικής χρήσης του προϊόντος;

#### **4.2.5 Η ΕΠΑΝΑΛΗΨΗ ΣΤΗ ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

Η επανάληψη αποτελείται από πέντε σειρές εργασιών όπως φαίνεται στην εικόνα 12.



Εικ. 12 Γενική επανάληψη στη φάση της επεξεργασίας

Διεξάγουμε τέσσερις εργασίες παράλληλα. Η πρώτη είναι οι βασικές σειρές εργασιών, η δεύτερη ο σχεδιασμός των επαναλήψεων, η τρίτη είναι η αποτίμηση και η τέταρτη είναι η επιπλέον προετοιμασία του περιβάλλοντος ανάπτυξης. Κατά την αρχιτεκτονική σχεδίαση καταγράφουμε, αναλύουμε, σχεδιάζουμε, υλοποιούμε και ελέγχουμε μόνο τις αρχιτεκτονικά απαραίτητες απαιτήσεις. Δεν δίνουμε μεγάλη σημασία σε πληροφορίες που δεν είναι αρχιτεκτονικά σημαντικές. Αυτές τις πληροφορίες τις αφήνουμε για τη φάση της κατασκευής. Ο κορμός της αρχιτεκτονικής που θα είναι το αποτέλεσμα αυτών των προσπαθειών θα είναι απλά ο σκελετός του συστήματος. Από μόνο του δεν θα μπορεί να κάνει πολλά πράγματα εκτός από τα μέρη εκείνα που πρέπει να υλοποιήσουμε σε βάθος για να επιβεβαιώσουμε ότι η αρχιτεκτονική ως σύνολο θα δουλέψει. Αναπτύσσουμε τον κορμό της αρχιτεκτονικής σε μια ή περισσότερες επαναλήψεις ανάλογα με την εμβέλεια του συστήματος, τους κινδύνους, το βαθμό δυσκολίας, την πολυπλοκότητα των τεχνικών λύσεων και φυσικά την εμπειρία του κατασκευαστή. Ο σκοπός της διερεύνησης των κινδύνων σε αυτή τη φάση δεν είναι να εξαλείψουμε εντελώς τους κινδύνους άλλα να τους μειώσουμε σε ένα αποδεκτό επίπεδο για τη φάση

της κατασκευής. Ένας διαφορετικό τρόπος να το θέσουμε αυτό είναι ότι στη φάση της επεξεργασίας αντιμετωπίζουμε του τεχνικούς κινδύνους που είναι αρχιτεκτονικής φύσεως, υλοποιώντας την αρχιτεκτονική. Τώρα, για το ποιο είναι το αποδεκτό επίπεδο των κινδύνων για τη φάση της κατασκευής, εννοούμε ότι οι κίνδυνοι έχουν διερευνηθεί έως του σημείου που έχουμε σκεφτεί τον τρόπο αντιμετώπισης τους και μπορούμε να υπολογίσουμε την προσπάθεια και το χρόνο που θα απαιτηθεί για αυτήν την αντιμετώπιση. Ο κίνδυνος δεν θα έχει εξαλειφθεί μέχρις ότου οι περιπτώσεις χρήσης που εμπεριέχουν τον κίνδυνο υλοποιηθούν, μερικές φορές στη φάση της επεξεργασίας, αλλά συνήθως στη φάση της κατασκευής και ενίοτε στη φάση της μετάβασης.

Έχουμε πει πως στη φάση της επεξεργασίας θα πρέπει να στοχεύουμε στο να αναγνωρίσουμε περίπου το 80% των περιπτώσεων χρήσης. Μπορεί να περιγράψουμε με λεπτομέρεια το 40% έως το 80% όλου του όγκου των περιπτώσεων χρήσης. Δεν χρειάζεται να αναγνωρίσουμε όλες τις περιπτώσεις χρήσης και δεν χρειάζεται να περιγράψουμε λεπτομερώς αυτές που αναγνωρίσαμε, γιατί ξέρουμε από προηγούμενες εμπειρίες ότι κάποια συγκεκριμένα υποσυστήματα μπορούν εύκολα να σχεδιαστούν αρχιτεκτονικά, δεν περιέχουν αναπάντεχους κινδύνους και μπορεί γίνει ακριβής εκτίμηση για αυτά. Από τον όγκο των περιπτώσεων χρήσης που περιγράψαμε αναλυτικά, ίσως να επιλέξουμε τον μισό για να τον εξετάσουμε προσεκτικά μέσω της ανάλυσης. Από αυτό το μισό μπορεί να κρίνουμε απαραίτητο να λάβουμε υπόψη μας μόνο ένα κομμάτι των σεναρίων για να σχεδιάσουμε, υλοποιήσουμε και να ελέγξουμε για να φτάσουμε στην αρχιτεκτονική και να αντιμετωπίσουμε τους κινδύνους, όπως φαίνεται και στην εικόνα 11. Ο στόχος είναι να συλλάβουμε τις απαιτήσεις έως το σημείο που είναι απαραίτητο για να πετύχουμε τους στόχους αυτής της φάσης.

Τοποθετούμε σε μια σειρά τις περιπτώσεις χρήσης και διεξάγουμε την ανάλυση, την σχεδίαση και υλοποίηση, σε αρχιτεκτονικό επίπεδο. Επίσης αναλύουμε τις κλάσεις και τα πακέτα και σχεδιάζουμε τις κλάσεις και τα υποσυστήματα. Τέλος κατασκευάζουμε ένα περιβάλλον ελέγχου και ελέγχουμε τα συστατικά και τον όλο κορμό της αρχιτεκτονικής που υλοποιεί τις αρχιτεκτονικές σημαντικές περιπτώσεις χρήσης.

Όσο η ομάδα είναι μικρή όπως είναι κατά τη φάση της επεξεργασίας, είναι η κατάλληλη ώρα να πραγματοποιούμε επαναλήψεις και να δοκιμάζουμε διαφορετικές λύσεις όπως για παράδειγμα σε τεχνολογίες, πλαίσια εργασίας και δομές. Αν το έργο είναι πολύπλοκο μπορεί να χρειαστούν πολλές επαναλήψεις μέχρι να βρεθεί μια σταθερή αρχιτεκτονική. Αργότερα στη φάση της κατασκευής, όταν θα εργάζονται πολλά άτομα στο έργο και θα υπάρχουν χιλιάδες γραμμές κώδικα για να επιβλεφθούν, χρειάζεται να εργαστούμε πάνω σε μια σταθερή αρχιτεκτονική και να αυξήσουμε το σύστημα σταδιακά. Λίγες επαναλήψεις ίσως να είναι αρκετές αν το σύστημα είναι μικρό και απλό, αλλά θα είναι μόνο το πρώτο βήμα αν το σύστημα είναι μεγάλο και δύσκολο. Επιπλέον επαναλήψεις εξαρτώνται από πτυχές του συστήματος, όπως η πολυπλοκότητα του και η σοβαρότητα των κινδύνων που εμπεριέχει. Οι επαναλήψεις συνεχίζονται έως ότου η αρχιτεκτονική πλησιάσει σε ένα σταθερό επίπεδο, που είναι το επίπεδο το οποίο αντιπροσωπεύει επαρκώς το σύστημα και όπου τυχόν περαιτέρω αλλαγές θα είναι ελάχιστες.

#### **4.2.6 ΕΚΤΕΛΕΣΗ ΤΩΝ ΣΕΙΡΩΝ ΕΡΓΑΣΙΩΝ**

Στη φάση της επεξεργασίας προχωράμε πάνω στην εργασία που έγινε στη φάση της σύλληψης. Παρόλα αυτά, στη σύλληψη το μόνο που είχαμε να κάνουμε ήταν να αποδείξουμε ότι μπορούμε να κάνουμε το έργο σε επόμενες φάσεις. Τώρα στη φάση της επεξεργασίας στην κυριολεξία το κάνουμε. Θα εξετάσουμε πάλι τι κάναμε πριν, άλλα πιθανώς θα είναι λίγα αυτά που θα επαναχρησιμοποιήσουμε. Τώρα δεν κοιτάμε μόνο για περιπτώσεις χρήσης που εμπεριέχουν κρίσιμους κινδύνους, άλλα για περιπτώσεις χρήσης που είναι αρχιτεκτονικά σημαντικές. Επίσης χρειαζόμαστε μεγαλύτερη κάλυψη των περιπτώσεων χρήσης για να υποστηρίξουμε έναν ακριβή προϋπολογισμό. Επιπλέον, εργαζόμαστε για να τελειώσουμε τη φάση με έναν εκτελέσιμο κορμό αρχιτεκτονικής, που θα είναι σταθερός για να μπορούμε να προσθέσουμε πάνω σε αυτόν στη φάση της κατασκευής. Οπότε πρέπει να δώσουμε περισσότερη προσοχή στην ποιότητα και επεκτασιμότητα της κατασκευής μας από ότι στη φάση της σύλληψης. Στο ξεκίνημα αυτής της φάσης εξετάζουμε τους κινδύνους και αναγνωρίζουμε τις περιπτώσεις χρήσης. Πρέπει να καλύψουμε περίπου το 80% των απαιτήσεων για να βρούμε τις αρχιτεκτονικά σημαντικές και επιπλέον να βρούμε αρκετές πληροφορίες για να κάνουμε τον προϋπολογισμό του έργου. Στο ακόλουθο υποθετικό έργο, υποθέτουμε ένα μέτριας δυσκολίας σύστημα στο οποίο μπορούμε να βρούμε τον κορμό της αρχιτεκτονική σε μία επανάληψη. Επίσης υποθέτουμε ότι είναι ένα σύστημα ενός καινούργιου πεδίου. Όπως έχουμε πει υπάρχουν τα αρχικά πλάνα του έργου και το αρχικό πλάνο της επανάληψης που δημιουργήθηκαν κατά τη φάση της σύλληψης. Το πρώτο βήμα είναι να προσθέσουμε κάποιες λεπτομέρειες στο πλάνο της επανάληψης.

##### **4.2.6.1 ΚΑΤΑΓΡΑΦΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ**

Σε αυτό το σημείο βρίσκουμε, κατατάσσουμε και συμπληρώνουμε τις λεπτομέρειες των περιπτώσεων χρήσης. Αναγνωρίζουμε επιπλέον περιπτώσεις χρήσης και χρήστες πέρα από αυτά που αναγνωρίστηκαν στη φάση της σύλληψης. Ενώ χρειάζεται να κατανοήσουμε το 80% περίπου των περιπτώσεων χρήσης για να πετύχουμε τους στόχους αυτής της φάσης, δεν χρειάζεται να συμπληρώσουμε λεπτομερώς τόσες πολλές. Μπορεί να αναγνωρίσουμε σχεδόν όλο το 80%, άλλα να περιγράψουμε μόνο ένα μέρος αυτών, και να αναλύσουμε ένα επίσης μέρος αυτών που περιγράψαμε. Με τον όρο κατανοήσουμε, εννοούμε να καταλάβουμε τι είναι αρχιτεκτονικά σημαντικό και να είμαστε σίγουροι ότι δεν παραλείψαμε τίποτα το οποίο θα επηρεάσει την αρχιτεκτονική και τον προϋπολογισμό. Το πόσες θα καταγράψουμε εξαρτάται επίσης και από πόση ακρίβεια θέλουμε. Για κάποια πολύπλοκα συστήματα μπορεί να χρειάζεται να αναγνωρίσουμε σχεδόν όλες τις περιπτώσεις χρήσης και να συμπληρώσουμε λεπτομερώς έως και το 80% αυτών. Αν χρηματοδοτούμε το έργο μόνοι μας τότε μπορούμε να σταματήσουμε σε ένα μικρότερο ποσοστό. Σταματώντας βέβαια σε ένα μικρότερο ποσοστό θα αυξηθεί ο κίνδυνος. Θα είναι ένα ρίσκο όπου αποδεχόμαστε μεγαλύτερο

κίνδυνο σε αντάλλαγμα για μικρότερο χρόνο και προσπάθεια στη φάση της επεξεργασίας.

Ακόμα μια εργασία που πραγματοποιείται κατά την καταγραφή των απαιτήσεων είναι η αναγνώριση των διασυνδέσεων χρήστη. Μας απασχολούν οι διασυνδέσεις χρήστη κατά τη φάση της επεξεργασίας μόνο αν είναι αρχιτεκτονικά σημαντικές. Παρόλα αυτά είναι σπάνιο το φαινόμενο και παρουσιάζεται μόνο σε μερικές περιπτώσεις όπου οι διασυνδέσεις χρήστη είναι ιδιόμορφες. Ένα παράδειγμα είναι όταν το σύστημα που αναπτύσσουμε από μόνο του είναι ένα πλαίσιο διασύνδεσης χρήστη. Επιπρόσθετο παράδειγμα είναι ένα σύστημα με ιδιαίτερα πρωτόκολλα επικοινωνίας που είναι σημαντικά για την αρχιτεκτονική σε σχέση με την απόδοση και χρόνο ανταπόκρισης. Υπάρχει ακόμα ένας λόγος για να αναγνωρίσουμε τις διασυνδέσεις χρήστη ακόμα και αν δεν είναι αρχιτεκτονικά σημαντικές. Αυτός είναι να μάθουμε από τους πραγματικούς χρήστες ότι το σύστημα δουλεύει. Παρόλα αυτά πρέπει να προχωράμε προς αυτή την ακραία κατεύθυνση μόνο όταν δεν έχουμε καταφέρει να επιδείξουμε την αξία του συστήματος στη φάση της σύλληψης. Ως γενικός κανόνας δεν χρειάζεται να δημιουργήσουμε πρωτότυπο των διασυνδέσεων χρήστη κατά την επεξεργασία.

Βασιζόμενοι στο μερικώς συμπληρωμένο μοντέλο περιπτώσεων χρήσης που ετοιμάστηκε στη φάση της σύλληψης ακολουθούμε δύο δρόμους. Ο πρώτος είναι να συμπληρώσουμε και άλλες περιπτώσεις χρήσης και να δουλέψουμε με τον κορμό της αρχιτεκτονικής. Στην αρχή ξοδεύουμε περισσότερο χρόνο στην εύρεση περισσότερων περιπτώσεων χρήσης και στη συνέχεια προχωράμε προς την αρχιτεκτονική. Παρόλα αυτά πρέπει να συντονίσουμε αυτούς τους δύο δρόμους.

Συμπληρώνουμε με λεπτομέρειες τις περιπτώσεις χρήσης που χρειαζόμαστε για να κατανοήσουμε επαρκώς τις απαιτήσεις και για να δημιουργήσουμε τον κορμό της αρχιτεκτονικής. Σε αυτή τη φάση περιορίζουμε τις προσπάθειες μας σε περιγραφές των αρχιτεκτονικά σημαντικών και πολύπλοκων περιπτώσεων χρήσης. Σε αυτές τις περιπτώσεις χρήσης συνήθως δεν συμπληρώνουμε με λεπτομέρειες όλα τα μέρη τους αλλά αντιθέτως μόνο τα σενάρια που χρειαζόμαστε. Αποφεύγουμε να περιγράψουμε οτιδήποτε δεν χρειαζόμαστε. Παρόλα αυτά σε κάποιες περιπτώσεις πολύπλοκων συστημάτων μπορεί να είναι χρήσιμο να περιγράψουμε σχεδόν όλα τα σενάρια όλων των περιπτώσεων χρήσης, δηλαδή σχεδόν το 100% του όγκου των περιπτώσεων χρήσης.

Επανεξετάζουμε το τι έχουμε κάνει και ψάχνουμε για ομοιότητες, απλοποιήσεις και ευκαιρίες για να βελτιώσουμε τη δομή του μοντέλου περιπτώσεων χρήσης. Εφαρμόζουμε μηχανισμούς, όπως επέκταση και γενίκευση, για να έχουμε ένα καλύτερα δομημένο και εύκολα κατανοήσιμο μοντέλο. Το μοντέλο μπορεί να γίνει πιο εύκολο για να αλλαχθεί, να επεκταθεί και να συντηρηθεί, αφού για παράδειγμα μειώσουμε τους πλεονασμούς. Ωστόσο μερικές φορές μπορεί να μην ανακαλύψουμε καλύτερη δομή σε αυτό το σημείο. Μπορεί να χρειαστεί να περιμένουμε έως προς το τέλος της επανάληψης.



#### 4.2.6.2 ΑΝΑΛΥΣΗ

Ξεκινήσαμε την σχεδίαση του μοντέλου ανάλυσης στη φάση της σύλληψης. Τώρα συνεχίζουμε πάνω σε αυτό, αν και μπορεί να βρούμε ότι δεν χρειαζόμαστε πλέον μεγάλα μέρη αυτού. Στη φάση της επεξεργασίας χρειάζεται να δουλέψουμε με τις περιπτώσεις χρήσης που είναι αρχιτεκτονικά σημαντικές και με τις πολύπλοκες που πρέπει να τις επανεξετάσουμε για να κατανοήσουμε καλύτερα το σύστημα και να κάνουμε ένα ακριβή προϋπολογισμό. Σε αυτό το μέρος θα δούμε τις εργασίες της αρχιτεκτονικής ανάλυσης, της ανάλυσης περίπτωσης χρήσης, της ανάλυσης κλάσης, και ανάλυσης πακέτου. Στην ανάλυση χρειάζεται να εξετάσουμε τις αρχιτεκτονικά σημαντικές. Αυτό το μέγεθος είναι συνήθως μικρότερο από το 10% του όγκου των περιπτώσεων χρήσης. Επίσης αναλύουμε τις περιπτώσεις χρήσης για να τις κατανοήσουμε καλύτερα και να διακρίνουμε τις αναμεταξύ τους παρεμβάσεις. Στο σύνολο μπορεί να χρειαστεί να δούμε το 50% του όγκου των περιπτώσεων χρήσης που έχουμε περιγράψει με λεπτομέρειες.

Στη φάση της σύλληψης προχωρήσαμε την αρχιτεκτονική ανάλυση ως το σημείο που χρειαζόταν για να δείξουμε ότι υπάρχει μια εφικτή αρχιτεκτονική. Αυτό συνήθως δεν είναι πολύ προχωρημένο. Τώρα στη φάση της επεξεργασίας έχουμε να προχωρήσουμε κι άλλο την αρχιτεκτονική ανάλυση ως το σημείο που μπορεί να υποστηρίξει μία εκτελέσιμη αρχιτεκτονική. Για αυτό το λόγο κάνουμε ένα αρχικό, υψηλού επιπέδου, διαμελισμό του συστήματος σε πακέτα ανάλυσης, βασιζόμενοι στην αρχιτεκτονική όψη του μοντέλου περιπτώσεων χρήσης, τις σχετικές απαιτήσεις, το λεξιλόγιο και την υπάρχουσα γνώση του πεδίου από το επιχειρηματικό μοντέλο. Μπορεί να εφαρμόσουμε μία αρχιτεκτονική με επίπεδα.

Πολλές περιπτώσεις χρήσης έτσι όπως περιγράφονται στο μοντέλο περιπτώσεων χρήσης δεν θα γίνουν ξεκάθαρα κατανοητές. Οι περιπτώσεις χρήσης πρέπει να βελτιωθούν ως προς την ανάλυση κλάσεων που υπάρχουν στις απαιτήσεις άλλα δεν χρειάζεται να υλοποιηθούν. Η ανάγκη για βελτίωση είναι έντονη για τις πολύπλοκες περιπτώσεις χρήσης και αυτών που θεωρούνται ότι έχουν επίπτωση η μία στην άλλη, δηλαδή περιπτώσεις χρήσης όπου η μία εξαρτάται από την άλλη. Για παράδειγμα μία περίπτωση χρήσης για να είναι σε θέση να προσπελάσει κάποια πληροφορία πρέπει κάποιες άλλες περιπτώσεις χρήσης να έχουν παράσχει αυτή τη πληροφορία. Συνεπώς οι αρχιτεκτονικά σημαντικές περιπτώσεις χρήσης και αυτές που έχουν σημασία να κατανοηθούν πρέπει να βελτιωθούν ως προς την ανάλυση κλάσεων. Οι άλλες περιπτώσεις χρήσης που δεν είναι σημαντικές από πλευράς αρχιτεκτονικής ή κατανόησης του συστήματος δεν βελτιώνονται ή αναλύονται. Για αυτές τις περιπτώσεις χρήσης χρειάζεται απλά να κατανοήσουμε τι είναι και ότι δεν έχουν καμία επίπτωση. Θα καταλάβουμε πώς να τις χειριστούμε όταν θα έρθει η ώρα να τις πραγματοποιήσουμε κατά τη φάση της κατασκευής. Οι σημαντικές ή πολύπλοκες περιπτώσεις χρήσης δεν χρειάζεται να περιγράφουμε με πολλές λεπτομέρειες, άλλα μέχρι μόνο το βαθμό όπου μπορούμε να καταλάβουμε την αποστολή που περιγράφουν, σχετικά με τον κορμό αρχιτεκτονικής και το επιχειρηματικό πλάνο. Αν κοιτάξαμε το 80% των περιπτώσεων χρήσης για να καταλάβουμε το ρόλο τους στο σύστημα, και περιγράψαμε λιγότερο από το 40% του όγκου των περιπτώσεων χρήσης, τότε συνήθως μπορούμε να πραγματοποιήσουμε την ανάλυση γιατί μερικές από αυτές τις περιπτώσεις χρήσης δεν θα έχουν καμία επίπτωση

στο επιχειρηματικό πλάνο. Έπειτα ξεκινάμε να βρούμε τις κλάσεις ανάλυσης που πραγματοποιούν τις περιπτώσεις χρήσης. Χρησιμοποιούμε τις αρχιτεκτονικά σημαντικές κλάσεις που περιγράφηκαν. Μετά κάνουμε ανάθεση ευθυνών σε αυτές τις κλάσεις. Αρκετή από την εργασία της ανάλυσης περιπτώσεων χρήσης είναι να ελέγξεις μία προς μία τις περιπτώσεις χρήσης του μοντέλου περιπτώσεων χρήσης και να συμπληρώσεις περισσότερες λεπτομέρειες σε σχέση με τις κλάσεις και τις ευθύνες τους. Επίσης θα αναδείξουμε και τις συσχετίσεις μεταξύ των κλάσεων και των χαρακτηριστικών τους. Με βάση αυτή την ανάλυση περίπτωσης χρήσης επιλέγουμε τις κλάσεις που είναι αρχιτεκτονικά σημαντικές και αυτές οι κλάσεις θα είναι η βάση της αρχιτεκτονικής όψης του μοντέλου ανάλυσης.

#### ΑΝΑΛΥΣΗ ΚΛΑΣΗΣ

Για αρχή κάνουμε μία αναθεώρηση των κλάσεων που αναγνωρίστηκαν στα προηγούμενα βήματα. Συγχωνεύουμε τις ευθύνες που ανατέθηκαν σε αυτές από διαφορετικές περιπτώσεις χρήσης. Επίσης αναγνωρίζουμε τους μηχανισμούς ανάλυσης που υπάρχουν και βρίσκουμε πως χρησιμοποιούνται από κάθε κλάση.

#### ΑΝΑΛΥΣΗ ΠΑΚΕΤΟΥ

Έχοντας γίνει η ομαδοποίηση κλάσεων με παρόμοιες λειτουργίες σε πακέτα λειτουργιών σε αυτό το στάδιο η εργασία μας είναι η αναθεώρηση και συντήρηση τους.

#### 4.2.6.3 ΣΧΕΔΙΑΣΗ

Σε αυτή τη φάση συνήθως σχεδιάζουμε και υλοποιούμε λιγότερο από το 10% του όγκου των περιπτώσεων χρήσης. Αυτό το μικρό ποσοστό είναι μόνο ένα κομμάτι του όγκου των περιπτώσεων χρήσης που αναγνωρίσαμε σε αυτή τη φάση. Στη φάση της επεξεργασίας σχεδιάζουμε σε αρχιτεκτονικό επίπεδο. Αυτό σημαίνει ότι σχεδιάζουμε τις αρχιτεκτονικά σημαντικές περιπτώσεις χρήσης, κλάσεις και υποσυστήματα. Τα πακέτα στην ανάλυση και τα υποσυστήματα στη σχεδίαση είναι κρίσιμα για τον προσδιορισμό των αρχιτεκτονικών όψεων.

Η αρχιτεκτονική όψη του μοντέλου σχεδίασης περιλαμβάνει υποσυστήματα, κλάσεις, διασυνδέσεις, και υλοποιήσεις των αρχιτεκτονικά σημαντικών περιπτώσεων χρήσης, που περιέχονται στο μοντέλο περιπτώσεων χρήσης. Αναγνωρίζουμε την αρχιτεκτονική των επιπέδων, περιλαμβανομένων των γενικών σχεδιαστικών μηχανισμών, τα υποσυστήματα και τις διασυνδέσεις τους, τις αρχιτεκτονικά σημαντικές κλάσεις σχεδίασης, και τη σχετική διάταξη των κόμβων σε περίπτωση καταναμημένου συστήματος.

Αναγνώριση της αρχιτεκτονικής των επιπέδων. Συνεχίζουμε την εργασία που ξεκίνησε στη φάση της σύλληψης και σχεδιάζουμε την αρχιτεκτονική των επιπέδων.

Αναγνώριση υποσυστημάτων και των διασυνδέσεων τους. Μετά συνεχίζουμε και εργαζόμαστε με τα υψηλότερα επίπεδα της αρχιτεκτονικής, κοντά στο επίπεδο εφαρμογής. Έτσι βασιζόμενοι στα πακέτα του μοντέλου ανάλυσης αναγνωρίζουμε

αντίστοιχα υποσυστήματα που θα περιληφθούν στο μοντέλο σχεδίασης. Συνήθως προσπαθούμε να κάνουμε κάθε πακέτο υπηρεσιών του μοντέλου ανάλυσης ένα υποσύστημα υπηρεσιών στο μοντέλο ανάλυσης. Υψηλότερου επιπέδου πακέτα ανάλυσης μετατρέπονται σε υποσυστήματα στο μοντέλο σχεδίασης. Αυτή η προσέγγιση είναι επιτυχής σε αρκετές περιπτώσεις, αλλά σε κάποιες άλλες έρχεται στην επιφάνεια η αντίθεση μεταξύ της ανάλυσης και της σχεδίασης. Σε άλλες περιπτώσεις μπορούμε να επιλέξουμε επαναχρησιμοποιήσιμα κατασκευαστικά κομμάτια, που είτε παρήχθησαν εσωτερικά ή από εξωτερικούς παράγοντες. Αυτά τα κομμάτια μπορεί να μην ταιριάζουν ακριβώς με τη δομή των πακέτων που παρουσιάζει το μοντέλο ανάλυσης ως πρόταση, οπότε θα επιλέξουμε μια δομή υποσυστήματος για την αρχιτεκτονική σχεδίαση κάπως διαφορετική από αυτήν που επιλέχθηκε στην αρχιτεκτονική ανάλυση.

Αναγνώριση των αρχιτεκτονικά σημαντικών σχεδιαστικών κλάσεων. Μετατρέπουμε τις αρχιτεκτονικά σημαντικές κλάσεις ανάλυσης σε κλάσεις σχεδίασης. Καθώς όλο και περισσότερο κλάσεις σχεδίασης δημιουργούνται επιλέγουμε αυτές που έχουν το μεγαλύτερο ενδιαφέρον από πλευράς αρχιτεκτονικής και τις περιλαμβάνουμε στην αρχιτεκτονική περιγραφή.

Αναγνώριση των κόμβων και της διάταξης τους σε περίπτωση που έχουμε ένα καταναμημένο σύστημα. Μελετάμε την συνεργασία και κατανομή που απαιτείται από το σύστημα εξερευνώντας τα νήματα και τις διαδικασίες που απαιτούνται και του φυσικού δικτύου των επεξεργαστών και άλλων συσκευών. Οι περιπτώσεις χρήσης που έχουν ήδη σχεδιαστεί, και συγκεκριμένα αυτές που παρουσιάζονται στα μοντέλα αλληλεπίδρασης, αποτελούν βασικό στοιχείο πάνω στο οποίο θα στηρίξουμε την εργασίας μας. Κατανέμουμε τα αντικείμενα που χρησιμοποιούνται στα μοντέλα αλληλεπίδρασης σε ενεργές κλάσεις και αυτές με τη σειρά τους αναθέτονται σε επεξεργαστές και άλλες συσκευές. Αυτό το βήμα προσφέρει λειτουργικότητα τόσο σε λογικό όσο και σε φυσικό επίπεδο. Στη συνέχεια ετοιμάζουμε μια καινούργια έκδοση της αρχιτεκτονικής όψης του μοντέλου σχεδίασης και μια καινούργια όψη του μοντέλου διάταξης, όπου και τα δύο περιλαμβάνονται στην αρχιτεκτονική περιγραφή.

#### ΣΧΕΔΙΑΣΗ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ

Οι αρχιτεκτονικά σημαντικές περιπτώσεις χρήσης τώρα σχεδιάζονται όσον αφορά τη σχεδίαση υποσυστημάτων, υποσυστημάτων λειτουργιών, ή κλάσεων σχεδίασης. Οι υπόλοιπες περιπτώσεις χρήσης που αναγνωρίστηκαν, συμπληρώθηκαν λεπτομερώς και αναλύθηκαν δεν περνάνε στη φάση της σχεδίασης σε αυτή τη φάση της διαδικασίας. Αυτή η εργασία είναι παρόμοια με την εργασία της ανάλυσης περίπτωσης χρήσης άλλα με λίγες σημαντικές διαφορές. Στην ανάλυση θέλαμε να αναλύσουμε και να βελτιώσουμε τις περιπτώσεις χρήσης ώστε να φτάσουν στο σημείο να είναι καλοφτιαγμένες, ελαστικές σε τυχόν μελλοντικές αλλαγές και επαναχρησιμοποιήσιμες. Επίσης εργαστήκαμε για να βρούμε τις ευθύνες των αναγνωρισμένων κλάσεων ανάλυσης. Στη σχεδίαση προχωράμε σε περισσότερες λεπτομέρειες. Προχωρώντας από την ανάλυση στη σχεδίαση πρέπει να προσαρμόσουμε το μοντέλο ανάλυσης για να φτάσουμε σε ένα εφαρμόσιμο μοντέλο σχεδίασης που να συμφωνεί με τους μηχανισμούς σχεδίασης. Τα πακέτα ανάλυσης και οι κλάσεις ανάλυσης μας παρέχουν έναν οδηγό για να βρούμε τα υποσυστήματα και τις κλάσεις σχεδίασης. Μόλις αυτά βρεθούν

περιγράφουμε όχι μόνο τις λειτουργίες που έχουν άλλα και τις αλληλεπιδράσεις μεταξύ τους. Στην ανάλυση δείξαμε πως η εστίαση μεταφέρεται από ένα στοιχείο σε άλλα στην εκτέλεση μιας περίπτωσης χρήσης. Χρησιμοποιήσαμε τα μοντέλα αλληλεπίδρασης για να δείξουμε αυτή τη μετακίνηση. Στη σχεδίαση επίσης ορίζουμε τις λειτουργίες που χρησιμοποιούνται για την επικοινωνία. Επιπλέον στη σχεδίαση πρέπει να λάβουμε υπόψη μας ποια επαναχρησιμοποιήσιμα υποσυστήματα θα χρησιμοποιήσουμε και ποιες λειτουργίες θα παρέχουν. Αν είναι δύσκολο να κατανοηθούν τότε και η σχεδίαση θα είναι δύσκολη να κατανοηθεί. Το αποτέλεσμα αυτής της εργασίας είναι ένα σύνολο σχεδιαστικών πραγματοποιήσεων περιπτώσεων χρήσης για κάθε μία αρχιτεκτονικά σημαντική περίπτωση χρήσης.

#### ΣΧΕΔΙΑΣΗ ΚΛΑΣΗΣ

Σχεδιάζουμε τις κλάσεις που μετείχαν στις πραγματοποιήσεις των περιπτώσεων χρήσης του προηγούμενου βήματος. Πρέπει να σημειώσουμε ότι οι κλάσεις δεν είναι ακόμα ολοκληρωμένες. Θα μετέχουν σε περισσότερες πραγματοποιήσεις περιπτώσεων χρήσης σε επόμενες επαναλήψεις. Σε αυτό το βήμα ενοποιούμε τους διαφόρους ρόλους κάθε κλάση σε μία ομοιόμορφη κλάση.

#### ΣΧΕΔΙΑΣΗ ΥΠΟΣΥΣΤΗΜΑΤΟΣ

Σχεδιάζουμε τα υποσυστήματα που παράχθηκαν από την αρχιτεκτονική σχεδίαση. Κατά τη διάρκεια αυτών των βημάτων ενημερώνουμε και την αρχιτεκτονική όψη του μοντέλου σχεδίασης όπως απαιτείται.

### 4.2.6.4 ΥΛΟΠΟΙΗΣΗ

Αυτή η σειρά εργασιών υλοποιεί και ελέγχει τα σημαντικά αρχιτεκτονικά συστατικά, ξεκινώντας από τα σημαντικά αρχιτεκτονικά στοιχεία σχεδίασης. Το αποτέλεσμα είναι ο κορμός της αρχιτεκτονικής, υλοποιημένος με λιγότερο από το 10% του όγκου των περιπτώσεων χρήσης. Στη συνέχεια θα καλύψουμε τις εργασίες της αρχιτεκτονικής υλοποίησης, υλοποίησης κλάσης και υλοποίησης υποσυστήματος και ενοποίησης συστήματος.

#### ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΛΟΠΟΙΗΣΗ

Βασίζόμενοι στην αρχιτεκτονική όψη του μοντέλου σχεδίασης και στην αρχιτεκτονική όψη του μοντέλου διάταξης, αναγνωρίζουμε τα συστατικά που απαιτούνται για να υλοποιήσουμε τα υποσυστήματα λειτουργιών. Τα εκτελέσιμα συστατικά αντιστοιχούνται με τους κόμβους μέσα στο δίκτυο υπολογιστών στους οποίους θα εκτελεστούν. Μετά αυτό παρουσιάζεται στην αρχιτεκτονική όψη του μοντέλου υλοποίησης.

#### ΥΛΟΠΟΙΗΣΗ ΚΛΑΣΗΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΥΠΟΣΥΣΤΗΜΑΤΟΣ

Στη σειρά εργασιών της σχεδίασης σχεδιάσαμε έναν αριθμό κλάσεων αναγκαίων για τη δημιουργία του κορμού της αρχιτεκτονικής. Αυτός ο κορμός θα είναι μια πρώιμη εκτελέσιμη έκδοση του συστήματος που θα κατασκευάσουμε. Σε αυτό το βήμα υλοποιούμε αυτές τις κλάσεις.

#### ΕΝΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

Με βάση ένα μικρό ποσοστό περιπτώσεων χρήσης που είναι να υλοποιηθούν σε αυτήν την επανάληψη δημιουργούμε ένα πλάνο ενοποίησης και μετά σταδιακά ενοποιούμε τα υποσυστήματα και τα αντίστοιχα τους συστατικά στον εκτελέσιμο κορμό της αρχιτεκτονικής.

#### 4.2.6.5 ΔΟΚΙΜΗ

Η εστίαση εδώ είναι να εξακριβώσουμε ότι όλα τα υποσυστήματα με τις διασυνδέσεις τους σε όλα τα επίπεδα δουλεύουν κανονικά. Ξεκινώντας από τα χαμηλά αρχιτεκτονικά στρώματα, σημαίνει τη δοκιμή της διανομής των αντικειμένων, της αποθήκευσης των αντικειμένων, η ανάκτηση αντικειμένων και άλλων μηχανισμών των χαμηλών επιπέδων του συστήματος. Δεν περιλαμβάνει μόνο τον έλεγχο για λειτουργικότητα αλλά και για αποδεκτή απόδοση.

#### ΠΛΑΝΟ ΔΟΚΙΜΗΣ

Επιλέγουμε στόχους που θα αξιολογήσουν τον κορμό της αρχιτεκτονικής. Ένας στόχος για παράδειγμα μπορεί να είναι να εκτελεστεί ένα σενάριο μίας περίπτωσης χρήσης μέσα σε ένα συγκεκριμένο χρόνο και με ένα συγκεκριμένο φόρτο.

#### ΕΛΕΓΧΟΣ ΣΧΕΔΙΑΣΗΣ

Με βάση αυτούς τους στόχους ορίζουμε τις περιπτώσεις δοκιμών που χρειάζονται και ετοιμάζουμε τις διαδικασίες ελέγχου για να δοκιμάσουμε τις επακόλουθες ενοποιήσεις υποσυστημάτων και αργότερα όλο το κορμό της αρχιτεκτονικής.

#### ΕΚΤΕΛΕΣΗ ΔΟΚΙΜΗΣ ΕΝΟΠΟΙΗΣΗΣ

Αφότου τα συστατικά δοκιμάζονται, γίνονται έτοιμα για δοκιμές ενοποίησης. Δοκιμάζουμε κάθε μία κατασκευή.

#### ΕΚΤΕΛΕΣΗ ΔΟΚΙΜΗΣ ΣΥΣΤΗΜΑΤΟΣ

Αφότου το σύστημα, έτσι όπως έχει καθοριστεί από τις περιπτώσεις χρήσης που είναι αρχιτεκτονικά σημαντικές, έχει ολοκληρωθεί, το δοκιμάζουμε. Αυτό το σύστημα, που είναι μια έκδοση του τελικού συστήματος, είναι ο κορμός της αρχιτεκτονικής. Μέσα από τις δοκιμές θα φέρουμε στην επιφάνεια τυχόν σφάλματα προς διόρθωση. Στο τέλος εξετάζουμε τα τελικά αποτελέσματα και εξακριβώνουμε αν πετύχαμε τους αρχικούς στόχους ή χρειάζεται να αποφασίσουμε πώς οι περιπτώσεις ελέγχων πρέπει να αλλάξουν για να πετύχουν αυτούς τους στόχους.

#### 4.2.7 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΣΤΗ ΦΑΣΗ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ

Για κάθε επανάληψη γίνεται η αποτίμηση στο τέλος της με βάση τα κριτήρια που επιλέχθηκαν πριν ξεκινήσει η επανάληψη. Επίσης εξετάζουμε τα αποτελέσματα κάθε επανάληψης για να επιβεβαιώσουμε ότι ο κορμός της αρχιτεκτονικής αντιπροσωπεύει την αρχιτεκτονική που θα υποστηρίξει την επίτευξη των γενικών στόχων και θα

αντιμετωπίσει τους κινδύνους. Αν είναι να γίνουν πολλές επαναλήψεις τότε το αποτέλεσμα της πρώτης επανάληψης μπορεί να είναι μόνο μία πρώτη μορφή της αρχιτεκτονικής. Το αποτέλεσμα της δεύτερης επανάληψης θα είναι μία δεύτερη μορφή της αρχιτεκτονικής, εξέλιξη της πρώτης. Στην τελευταία επανάληψη θα δημιουργηθεί η τελική μορφή του κορμού της αρχιτεκτονικής. Επιπλέον στο τέλος κάθε επανάληψης αξιολογούμε το τι έχει επιτευχθεί σε σχέση με τα κριτήρια και αν δεν έχουν επιτευχθεί κάποια από αυτά τότε μεταφέρουμε τις ανολοκλήρωτες εργασίες σε επόμενες επαναλήψεις. Στο τέλος της φάσης της επεξεργασίας έχουμε πείσει του εμπλεκόμενους ότι οι σημαντικοί κίνδυνοι έχουν αντιμετωπιστεί και έχει κατασκευαστεί μία σταθερή αρχιτεκτονική. Έτσι τους πείθουμε ότι το σύστημα μπορεί να κατασκευαστεί σύμφωνα με το πλάνο του έργου.

#### **4.2.8 ΔΗΜΙΟΥΡΓΙΑ ΠΛΑΝΟΥ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ**

Προς το τέλος της φάσης της επεξεργασίας ξεκινάμε να σχεδιάζουμε την πρώτη επανάληψη της φάσης της κατασκευής με λεπτομέρειες και τις υπόλοιπες με πιο γενικό τρόπο. Ο αριθμός των επαναλήψεων που απαιτείται εξαρτάται από το μέγεθος και την πολυπλοκότητα του έργου. Σε κάθε επανάληψη δημιουργείται μια κατασκευή η οποία προστίθεται ως ένα σχετικά μικρό κομμάτι σε αυτό που έχει ήδη κατασκευαστεί. Συνεχίζουν να υφίστανται αρκετοί κίνδυνοι στη λίστα κινδύνων. Βάζουμε αυτούς του κινδύνους σε μια σειρά διερεύνησης τους, έτσι ώστε να προλάβουμε να τους αντιμετωπίσουμε πριν την εμφάνιση τους σε κάποια επανάληψη. Η αρχή παραμένει η ίδια. Αντιμετώπιση των κινδύνων πριν εμφανιστούν και διακόψουν την ανάπτυξη του έργου. Ορίζουμε τη σειρά της διερεύνησης των υπόλοιπων περιπτώσεων χρήσης και των σεναρίων τους και τη σειρά με την οποία θα ολοκληρώσουμε το συμπλήρωμα των μοντέλων. Σε μεγάλα έργα για να μειώσουμε τον ολικό χρόνο του έργου μπορούμε να προσλάβουμε περισσότερα άτομα ώστε να μοιράσουμε κάποιες εργασίες που μπορούν να γίνουν παράλληλα. Η ανάπτυξη μεγάλων βιομηχανικών συστημάτων σημαίνει ότι πρέπει να βρούμε εργασίες που μπορούν να γίνουν παράλληλα γιατί τέτοια έργα συνήθως έχουν χρονικούς περιορισμούς. Αυτός ο τρόπος βασίζεται στα υποσυστήματα που δημιουργήθηκαν στον κορμό της αρχιτεκτονικής. Στην εργασία της σχεδίασης εμπνευστήκαμε από τα πακέτα ανάλυσης για να βρούμε υποσυστήματα σε διάφορα επίπεδα. Τα υποσυστήματα έχουν διασυνδέσεις, και ένα υψηλού επιπέδου στόχος ήταν η αναγνώριση και καθορισμός αυτών των διασυνδέσεων. Οι διασυνδέσεις είναι ο πυρήνας της αρχιτεκτονικής. Με τα υποσυστήματα και τις διασυνδέσεις είμαστε προετοιμασμένοι να εργαστούμε παράλληλα. Αν οι ομάδες που θα εργαστούν παράλληλα θα είναι ανεξάρτητες πρέπει οι διασυνδέσεις, που ορίζουν τα όρια της εργασίας τους, πρέπει να είναι ξεκάθαρες. Για να δώσουμε έμφαση στη σημασία αυτού του καθορισμού των διασυνδέσεων αναφέρουμε ότι μερικές φορές ονομάζονται και συμβόλαια. Ένα συμβόλαιο δεσμεύει τους τωρινούς κατασκευαστές άλλα και αυτών των επόμενων κύκλων σε αυτή τη διασύνδεση. Είναι γεγονός ότι μια διασύνδεση είναι που κάνει την προσθήκη στην αρχιτεκτονική δυνατή. Αργότερα οι κατασκευαστές μπορούν να αντικαταστήσουν ένα υποσύστημα με ένα άλλο υποσύστημα, αρκεί να μην σπάσουν το συμβόλαιο της διασύνδεσης. Η κατασκευή υποσυστημάτων που διασυνδέονται μέσα από

ένα συμβόλαιο διασύνδεσης είναι ως αρχή περίπου το ίδιο με την κατασκευή συστημάτων από συστήματα.

#### 4.2.9 ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ

Τα παραγόμενα της φάσης της επεξεργασίας είναι τα εξής:

- Ένα ολοκληρωμένο επιχειρηματικό μοντέλο που περιγράφει το περιβάλλον του συστήματος.
- Μια καινούργια έκδοση όλων των μοντέλων: περιπτώσεων χρήσης, ανάλυσης, σχεδίασης, διάταξης, και υλοποίησης. Στο τέλος της φάσης της επεξεργασίας αυτά τα μοντέλα θα είναι συμπληρωμένα έως το 10% περίπου εκτός από τα μοντέλα περιπτώσεων χρήσης και ανάλυσης μπορεί να περιέχουν περισσότερες περιπτώσεις χρήσης για να επιβεβαιώσουν ότι οι απαιτήσεις έχουν κατανοηθεί. Η πλειονότητα των περιπτώσεων χρήσης έχει κατανοηθεί για να είμαστε σίγουροι ότι δεν έχουν αγνοηθεί αρχιτεκτονικά σημαντικές περιπτώσεις χρήσης.
- Ένας εκτελέσιμος κορμός της αρχιτεκτονικής.
- Μια αρχιτεκτονική περιγραφή που περιλαμβάνει όψεις των μοντέλων περιπτώσεων χρήσης, ανάλυσης, σχεδίασης, διάταξης, και υλοποίησης.
- Ενημερωμένη λίστα κινδύνων.
- Πλάνο έργου για τις φάσεις της κατασκευής και μετάβασης.
- Ένα προκαταρκτικό εγχειρίδιο χρήσης.
- Ολοκληρωμένο επιχειρηματικό πλάνο.

#### 4.3 ΦΑΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ

Ο κυρίαρχος στόχος αυτής της φάσης είναι να δημιουργήσει ένα προϊόν λογισμικού που θα αποτελεί μια αρχική λειτουργική έκδοση, που πολλές φορές ονομάζεται και δοκιμαστική έκδοση (beta release). Το προϊόν αυτό πρέπει να είναι ικανοποιητικής ποιότητας και να ικανοποιεί τις απαιτήσεις. Η διάρκεια κατασκευής πρέπει να είναι μέσα στα όρια του επιχειρηματικού πλάνου. Στο τέλος της φάσης της επεξεργασίας φέραμε το προτεινόμενο σύστημα σε μια κατάσταση όπου είχαμε έναν εκτελέσιμο κορμό αρχιτεκτονικής. Οι προηγούμενες φάσεις έχουν μειώσει τους κρίσιμους και σημαντικούς κινδύνους σε αποδεκτά επίπεδα, ώστε να είναι αντιμετωπίσιμοι κατά τη φάση της κατασκευής. Κατά τη φάση της επεξεργασίας θέσαμε τα θεμέλια των αρχιτεκτονικά σημαντικών στοιχείων των μοντέλων της σχεδίασης και διάταξης. Αυτά τα θεμέλια περιείχαν τα υποσυστήματα, τις κλάσεις, και τα συστατικά και τις διασυνδέσεις μεταξύ τους. Επίσης περιείχαν τις πραγματοποιήσεις των σημαντικών περιπτώσεων χρήσης. Το πετύχαμε αυτό περιγράφοντας λεπτομερώς περίπου το 10% του όγκου των περιπτώσεων χρήσης. Να υπενθυμίσουμε ότι περίπου το 80% των περιπτώσεων χρήσης καταγράφηκαν άλλα δεν συμπληρώθηκαν λεπτομερώς για να πετύχουμε τους στόχους της φάσης της επεξεργασίας. Αυτό είναι που θα κάνουμε στη φάση της κατασκευής.

Ξεκινώντας από ένα εκτελέσιμο κορμό αρχιτεκτονικής και εργαζόμενοι μέσα από μια σειρά από επαναλήψεις και σταδιακές προσθήσεις, αναπτύσσουμε ένα προϊόν λογισμικού

που είναι έτοιμο για αρχική λειτουργία στο περιβάλλον του χρήστη, και συχνά ονομάζεται δοκιμαστική έκδοση. Περιγράφουμε λεπτομερώς τις εναπομείναντες περιπτώσεις χρήσης και σενάρια, τροποποιούμε την αρχιτεκτονική περιγραφή αν χρειάζεται, και συνεχίζουμε τις εργασίες μέσω επιπλέον επαναλήψεων συμπληρώνοντας τα μοντέλα ανάλυσης, σχεδίασης, και υλοποίησης. Ολοκληρώνουμε τα υποσυστήματα και τα δοκιμάζουμε και μετά ολοκληρώνουμε ολόκληρο το σύστημα και το δοκιμάζουμε. Καθώς προχωράμε από τη φάση της επεξεργασίας στη φάση της κατασκευής γίνεται μια αλλαγή του σημείου έμφασης μας. Ενώ οι φάσεις της σύλληψης και επεξεργασίας είχαν το χαρακτηριστικό της έρευνας, η φάση της κατασκευής έχει το χαρακτηριστικό της κατασκευής. Η εστίαση μετατοπίζεται από την προσπάθεια συγκέντρωσης της γνώσης που χρειάζεται για την κατασκευή του έργου, στην πραγματική κατασκευή του έργου με παραμέτρους κόστους, προσπάθειας, και χρονοδιαγράμματος. Κατά τη φάση της κατασκευής επιβεβαιώνουμε ότι οι περιπτώσεις χρήσης έχουν μπει σε σειρά προτεραιότητας, έχουν μοιραστεί σε επαναλήψεις, και με τέτοιο τρόπο ώστε να αποφευχθούν οι επαναλήψεις εργασιών. Διατηρούμε τη λίστα κινδύνων ενημερωμένη βελτιώνοντας την συνεχώς, ώστε να αντικατοπτρίζει κάθε στιγμή τους τρέχοντες κινδύνους. Στόχος μας είναι να τελειώσουμε αυτή τη φάση έχοντας αντιμετωπίσει όλους τους κινδύνους, εκτός βέβαια εκείνους που θα εμφανιστούν κατά τη λειτουργία και θα αντιμετωπιστούν στη φάση της μετάβασης. Ελέγχουμε ότι η κατασκευή προσκολλάται στην αρχιτεκτονική, και όταν χρειάζεται τροποποιούμε την αρχιτεκτονική ώστε να ενσωματώσουμε τις αλλαγές που προκύπτουν κατά τη φάση της κατασκευής.

#### **4.3.1 ΑΡΧΙΚΑ ΣΤΗ ΦΑΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ**

Σχεδιάσαμε τη φάση της κατασκευής στο τέλος της φάσης της επεξεργασίας. Μπορεί να χρειαστεί να τροποποιήσουμε το πλάνο ανάλογα με συνθήκες που μπορεί να έχουν αλλάξει. Συγκεκριμένα αναφέρουμε δύο συνθήκες που συχνά συμβαίνουν. Η μία είναι το πιθανό χρονικό κενό ανάμεσα στην φάση της επεξεργασίας και στη φάση της κατασκευής. Η επενδυτική ομάδα του έργου μπορεί να εγκρίνει κατευθείαν την έναρξη της φάσης της κατασκευής εξουσιοδοτώντας μας να συνεχίσουμε χωρίς διακοπή και να διατηρήσουμε τη λεπτομερή γνώση του έργου. Δυστυχώς, μπορεί να προκύψει ένα χρονικό κενό, όπως κάποιους μήνες, μέχρι την έγκριση της συνέχισης του έργου. Η δεύτερη συνθήκη που μπορεί να συμβεί είναι ότι η επένδυση και το χρονοδιάγραμμα είναι μικρότερα από αυτά που σχεδιάστηκαν στη φάση της επεξεργασίας. Η εμβέλεια μπορεί να έχει μειωθεί για να ταιριάζει με τη χρηματοδότηση και το χρονοδιάγραμμα ή όχι. Αυτό για το οποίο πρέπει να προετοιμαστούμε είναι ότι στην έναρξη της φάσης της κατασκευής οι συνθήκες μπορεί να είναι διαφορετικές, λίγο ή πολύ, από εκείνες υπό τις οποίες κάναμε το πλάνο στο τέλος της φάσης της επεξεργασίας. Πρέπει να ξανακάνουμε το πλάνο έως κάποιο βαθμό. Στις περισσότερες περιπτώσεις θα πρέπει να προσαρμόσουμε το πλάνο από τη φάση της επεξεργασίας για να συμφωνήσουμε με τους πόρους που διαθέτουν οι επενδυτές και χρονοδιάγραμμα που θέτουν.



### 4.3.2 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ

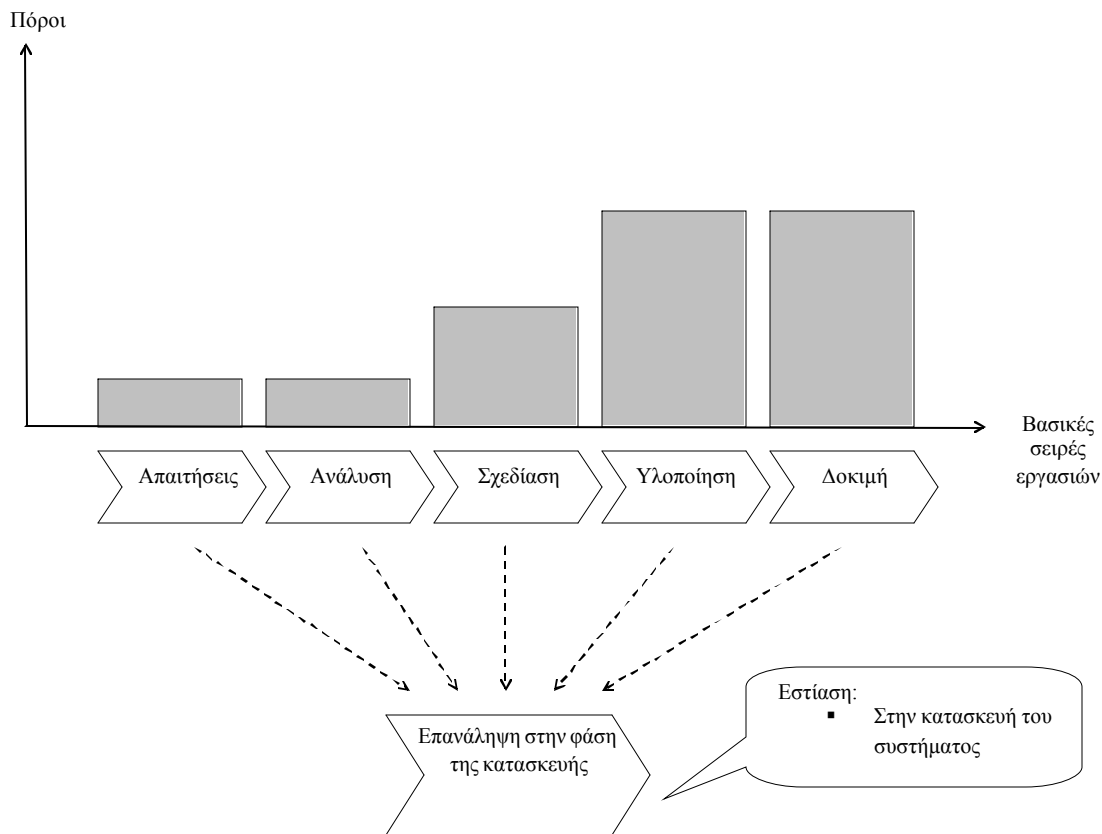
Τα συγκεκριμένα κριτήρια που πρέπει να επιτευχθούν από μια επανάληψη ή από ολόκληρη τη φάση της κατασκευής είναι μοναδικά για κάθε έργο. Καθορίστηκαν πρακτικά κατά την εργασία της ανάπτυξης των περιπτώσεων χρήσης. Όπως έχουμε πει οι περιπτώσεις χρήσης αντιπροσωπεύουν λειτουργικές απαιτήσεις. Επίσης μπορεί να έχουν μη λειτουργικές απαιτήσεις, όπως η απόδοση, προσαρτημένες σε αυτές. Το κριτήριο αξιολόγησης για αυτές τις περιπτώσεις χρήσης βασίζεται στις λειτουργικές και μη λειτουργικές απαιτήσεις που σχετίζονται με τις συγκεκριμένες περιπτώσεις χρήσης. Αυτά τα κριτήρια αξιολόγησης, σχετικά με τις περιπτώσεις χρήσης, μας καθιστούν ικανούς να δούμε ξεκάθαρα πότε έχουμε ολοκληρώσει μια επανάληψη. Επίσης, επιπλέον υλικό ετοιμάζεται κατά τη φάση της κατασκευής για το οποίο χρειάζονται κριτήρια αξιολόγησης. Για παράδειγμα:

- Υλικό για τον τελικό χρήστη. Ετοιμάζεται μια πρώτη μορφή του γραπτού υλικού για την υποστήριξη των τελικών χρηστών, όπως εγχειρίδια χρήσης και πληροφορίες έκδοσης. Το κριτήριο αξιολόγησης είναι το αν είναι επαρκή ώστε να υποστηρίξουν του χρήστες στη φάση της μετάβασης.
- Μια σειρά προϊόντων. Μια αρχική περιγραφή μιας σειράς προϊόντων για την υποστήριξη των τελικών χρηστών, όπως προγράμματα εκμάθησης και παραδείγματα, τα οποία θα ετοιμαστούν αρχικά σε αυτή τη φάση. Το κριτήριο αξιολόγησης είναι επίσης αν θα είναι ικανά να υποστηρίξουν τους χρήστες στη φάση της μετάβασης.

Για τη φάση της κατασκευής σαν σύνολο, το κριτήριο αξιολόγησης είναι αν η αρχική λειτουργική ικανότητα του συστήματος είναι αρκετά ώριμη και σταθερή ώστε να διατεθεί η δοκιμαστική έκδοση στην κοινότητα των χρηστών χωρίς να εκτεθούμε εμείς ή κοινότητα των χρηστών σε αναπάντεχους κινδύνους.

### 4.3.3 Η ΓΕΝΙΚΗ ΕΠΑΝΑΛΗΨΗ ΣΤΗ ΦΑΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ

Μία γενική επανάληψη στη φάση της κατασκευής περιλαμβάνει τις πέντε βασικές σειρές εργασιών, όπως φαίνονται στην εικόνα 13.



Εικ. 13 Γενική επανάληψη στη φάση της κατασκευής

Όπως και στη φάση της επεξεργασίας έτσι και κατά τη φάση της κατασκευής, εκτελούμε ως κάποιο βαθμό παράλληλα τέσσερα σύνολα εργασιών. Το πρώτο είναι οι βασικές σειρές εργασιών. Το δεύτερο είναι η δημιουργία των πλάνων των επαναλήψεων. Το τρίτο η δημιουργία του επιχειρηματικού πλάνου. Και το τέταρτο είναι η αποτίμηση. Στις πρώτες επαναλήψεις της φάσης της κατασκευής δίνουμε έμφαση περισσότερο στις αρχικές βασικές σειρές εργασιών, ενώ στις τελευταίες λιγότερο. Η έμφαση μετατοπίζεται κατά την διαδοχή των επαναλήψεων.

Μέχρι τώρα οι απαιτήσεις και η αρχιτεκτονική θα έχουν σταθεροποιηθεί. Η έμφαση τώρα είναι στην πραγματοποίηση όλων των περιπτώσεων χρήσης, σχεδίαση των απαραίτητων υποσυστημάτων και κλάσεων, η υλοποίηση τους ως συστατικά, και ο έλεγχος τους ατομικά και μέσα σε κατασκευές. Η καθοδηγούμενη από τις περιπτώσεις χρήσης, με κεντρικό άξονα την αρχιτεκτονική και επαναληπτική ανάπτυξη, διαδικασία κατασκευάζει μικρές κατασκευές, όπου κάθε κατασκευή προστίθεται στις προηγούμενες με τέτοιο τρόπο ώστε κάθε φορά να συντηρείται μια εκτελέσιμη κατασκευή.

Κατασκευάζοντας το λογισμικό με μικρές κάθε φορά προσαυξήσεις κάνει το έργο πιο εύκολο στη διαχείριση του. Μειώνει το εύρος των εργασιών της ανάλυσης, της σχεδίασης, της υλοποίησης και του ελέγχου στο μικρότερο μέγεθος θεμάτων και προβλημάτων που υπάρχουν μέσα σε μια επανάληψη. Σε μεγάλο βαθμό απομονώνει τους κινδύνους, τις ατέλειες, και σφάλματα, μέσα σε μια μικρή έκταση μιας μεμονωμένης κατασκευής καθιστώντας τα πιο εύκολα στο να βρεθούν και να αντιμετωπιστούν. Οι προηγούμενες φάσεις έχουν διερευνήσει και αντιμετωπίσει κρίσιμους και σημαντικούς κινδύνους, αλλά το έργο κουβαλάει ακόμα αρκετούς απλούς κινδύνους στη λίστα κινδύνων. Επιπλέον καινούργιοι κίνδυνοι μπορεί να εμφανιστούν καθώς προχωράμε με τις επαναλήψεις και καθώς οι χρήστες δοκιμάζουν τις προσαυξήσεις.

#### **4.3.4 ΕΚΤΕΛΕΣΗ ΤΩΝ ΒΑΣΙΚΩΝ ΣΕΙΡΩΝ ΕΡΓΑΣΙΩΝ**

Προηγουμένως περιγράψαμε τον γενικό σκοπό της φάσης της κατασκευής. Στη συνέχεια θα παρουσιάσουμε τις εργασίες που πραγματοποιούμε με λεπτομέρειες. Όπως και στις προηγούμενες φάσεις, έτσι και σε αυτήν τη φάση θα παρουσιάσουμε τις πέντε βασικές σειρές εργασιών, που αν και φαίνονται διαδοχικές στην πραγματικότητα πραγματοποιούνται παράλληλα. Στη φάση της επεξεργασίας μπορεί να έχουμε σχεδιάσει και υλοποιήσει έως το 10% του όγκου των περιπτώσεων χρήσης, που είναι αρκετές για να δημιουργήσουμε τον κορμό της αρχιτεκτονικής. Τώρα στη φάση της κατασκευής το έργο μας είναι να προσθέσουμε μύες στον αρχιτεκτονικό σκελετό. Δεν είναι κακό να ξαναεπισημάνουμε ότι οι πέντε βασικές σειρές εργασιών επαναλαμβάνονται σε κάθε επανάληψη.

##### **4.3.4.1 ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ**

Βρίσκουμε τις περιπτώσεις χρήσης και του χρήστες, δημιουργούμε πρωτότυπα των διασυνδέσεων χρήστη, συμπληρώνουμε με λεπτομέρειες τις περιπτώσεις χρήσης, και συνθέτουμε τις περιπτώσεις χρήσης. Στη φάση της επεξεργασίας αναγνωρίσαμε σχεδόν όλες τις περιπτώσεις χρήσης και τους χρήστες, κατανοήσαμε περίπου το 80% του όγκου των περιπτώσεων χρήσης, άλλα περιγράψαμε με λεπτομέρεια κάπου το 20% και από αυτό επιλέξαμε ένα 10% που χρειαζόμασταν για να δημιουργήσουμε τον κορμό της αρχιτεκτονικής. Στη φάση της κατασκευής φυσικά θα φτάσουμε έως ένα αρχικό λειτουργικό σύστημα, οπότε πρέπει να προχωρήσουμε την καταγραφή των απαιτήσεων ως το τέλος, δηλαδή να αναγνωρίσουμε και να συμπληρώσουμε με λεπτομέρειες το 100% αυτών.

- Εύρεση περιπτώσεων χρήσης και χρηστών: Συνήθως ένα μικρό μέρος των περιπτώσεων και των χρηστών απομένει να αναγνωριστούν στη φάση της κατασκευής, μικρότερο του 10%. Όταν χρειάζεται ενημερώνουμε τις περιπτώσεις χρήσης και τους χρήστες στο μοντέλο περιπτώσεων χρήσης.
- Δημιουργία πρωτότυπων των διασυνδέσεων χρήστη: Κατά γενικό κανόνα στη φάση της επεξεργασίας δεν προχωρούσαμε στη δημιουργία πρωτότυπων των διασυνδέσεων χρήστη, εκτός και αν είχαμε κάποιο καινούργιο τύπο διασύνδεσης χρήστη ή χρειαζόμασταν ένα πρωτότυπο για επίδειξη. Ωστόσο οι διασυνδέσεις

χρήστη πρέπει τώρα να σχεδιαστούν. Το πόσο εξαρτάται από το είδος του συστήματος που κατασκευάζουμε. Για κάποια συστήματα, ειδικά εκείνα που οι περιπτώσεις χρήσης τους απαιτούν πολύπλοκη διασύνδεση χρήστη, οι διασυνδέσεις χρήστη είναι δύσκολες στο να κατανοηθούν χωρίς την κατασκευή ενός πρωτότυπου. Οπότε κατασκευάζουμε ένα πρωτότυπο ή πολλά και βάζουμε τους χρήστες να το δοκιμάσουν. Βασιζόμενοι στις απόψεις των χρηστών μετά τη δοκιμή (feedback), τροποποιούμε το πρωτότυπο ώσπου να ικανοποιήσει τις ανάγκες των χρηστών. Η σχεδίαση των διασυνδέσεων χρήστη είναι μέρος της εργασίας των απαιτήσεων και όχι της σχεδίασης και χρειάζεται να γίνει πριν προχωρήσουμε στις επόμενες εργασίες. Το πρωτότυπο μετά γίνεται μια προδιαγραφή της διασύνδεσης χρήστη. Για συστήματα που θα πουληθούν σε ποσότητες, κατασκευάζουμε ένα πρωτότυπο χρήστη ακόμα και αν δεν είναι πολύ πολύπλοκο. Και όλα αυτά γιατί το κόστος της αντικατάστασης μιας μη ικανοποιητικής διασύνδεσης χρήστη θα ήταν πολύ μεγάλο.

- Κατάταξη των περιπτώσεων χρήσης σε προτεραιότητα: Στη φάση της επεξεργασίας κατατάξαμε τις περιπτώσεις χρήσης που χρειαζόμασταν για τον κορμό της αρχιτεκτονικής. Σε αυτή τη φάση καθώς αναγνωρίζουμε επιπλέον περιπτώσεις χρήσης τις τοποθετούμε ανάλογα στην λίστα προτεραιότητας των περιπτώσεων χρήσης.
- Συμπλήρωση λεπτομερειών περίπτωσης χρήσης: Συμπληρώνουμε με λεπτομέρεια τις εναπομείναντες περιπτώσεις χρήσης και τα σενάρια τους ανάλογα με την σειρά σημαντικότητας τους.
- Κατασκευή του μοντέλου περιπτώσεων χρήσης: Μπορεί να θέλουμε να βελτιώσουμε τη δομή του μοντέλου περιπτώσεων χρήσης. Όμως επειδή το σύστημα έχει αποκτήσει μια σταθερή αρχιτεκτονική σε αυτό το σημείο, οποιαδήποτε αλλαγή πρέπει να αφορά περιπτώσεις χρήσης με τις οποίες δεν έχουμε εργαστεί. Κάθε αλλαγμένη περίπτωση χρήσης απαιτεί μία αντίστοιχη αλλαγμένη πραγματοποίηση της περίπτωσης χρήσης στο μοντέλο ανάλυσης και σχεδίασης.

#### 4.3.4.2 ΑΝΑΛΥΣΗ

Σε αυτό το μέρος εκτελούμε τις εργασίες της αρχιτεκτονικής ανάλυσης, της ανάλυσης περίπτωσης χρήσης, της ανάλυσης κλάσης, και της ανάλυσης πακέτου που ξεκίνησαν στη φάση της επεξεργασίας. Σε εκείνη τη φάση χρειαζόταν να εργαστούμε με εκείνες τις περιπτώσεις χρήσης που ήταν αρχιτεκτονικά σημαντικές ή χρειαζόνταν για να υποστηρίξουμε το επιχειρηματικό πλάνο. Για να δώσουμε μια αίσθηση του που βρισκόμαστε αυτή τη στιγμή, λέμε ότι μπορεί να έχουμε αναλύσει το 40% του όγκου των περιπτώσεων χρήσης κατά τη φάση της επεξεργασίας. Πρέπει να επισημάνουμε ότι τα ποσοστά είναι ενδεικτικά και όχι πραγματικά. Εξαρτώνται ανάλογα με τις συνθήκες του κάθε έργου. Τώρα στη φάση της κατασκευής ασχολούμαστε με όλες τις περιπτώσεις χρήσης, άλλα δεν είναι απαραίτητο να επεκτείνουμε το μοντέλο ανάλυσης με όλες αυτές. Η ουσιώδης διαφορά μεταξύ της φάσης της επεξεργασίας και της φάσης της κατασκευής είναι ότι τώρα στη φάση της κατασκευής συμπληρώνουμε το μοντέλο ανάλυσης. Το μοντέλο ανάλυσης που είχαμε στο τέλος της φάσης της επεξεργασίας ήταν η

αρχιτεκτονική όψη, και ήταν σε μεγάλο βαθμό σχετικό με την αρχιτεκτονική. τώρα η αρχιτεκτονική όψη του μοντέλου ανάλυσης θα είναι μόνο ένα μέρος του ολόκληρου μοντέλου ανάλυσης. Στο τέλος της φάσης της κατασκευής θα έχουμε ολοκληρωμένο το μοντέλο ανάλυσης. Η αρχιτεκτονική όψη θα είναι μόνο ένα υποσύνολο του.

- Αρχιτεκτονική ανάλυση. Έχουμε ετοιμάσει την αρχιτεκτονική όψη του μοντέλου ανάλυσης έως το τέλος της φάσης της επεξεργασίας. Συνεπώς δεν υπάρχει τίποτα παραπάνω να κάνουμε στη φάση της κατασκευής, εκτός από ενημερώσεις που πρέπει να κάνουμε λόγω τυχόν αλλαγών που επηρεάζουν την αρχιτεκτονική.
- Ανάλυση περίπτωσης χρήσης. Στη φάση της επεξεργασίας χρησιμοποιήσαμε μόνο αυτές που ήταν αρχιτεκτονικά σημαντικές και τις ενσωματώσαμε στην αρχιτεκτονική όψη του μοντέλου ανάλυσης. Σε κάθε επανάληψη στη φάση της κατασκευής επεκτείνουμε το μοντέλο ανάλυσης με εκείνες τις περιπτώσεις χρήσης που περιλαμβάνονται στην επανάληψη.
- Ανάλυση κλάσης. Συνεχίζουμε την εργασία που ξεκινήσαμε στην φάση της επεξεργασίας.
- Ανάλυση πακέτου. Αναγνωρίσαμε τα πακέτα στη φάση της επεξεργασίας και τα βελτιώνουμε στη φάση της κατασκευής ανάλογα με τις καινούργιες περιπτώσεις χρήσης. Επίσης συντηρούμε τα πακέτα κατά τη φάση της κατασκευής.

#### 4.3.4.3 ΣΧΕΔΙΑΣΗ

Σε αυτή τη φάση σχεδιάζουμε και υλοποιούμε το υπόλοιπο 90% των περιπτώσεων χρήσης, αυτές που δεν θεωρήθηκαν απαραίτητες για την ανάπτυξη του κορμού της αρχιτεκτονικής. Καθώς θα εξετάσουμε την εργασία της σχεδίασης, πρέπει να δώσουμε έμφαση στο γεγονός ότι οι βασικές σειρές εργασιών επαναλαμβάνονται σε κάθε επανάληψη.

- Αρχιτεκτονική σχεδίαση. Ως γενικό κανόνα στη φάση της κατασκευής δεν θα προσθέσουμε υποσυστήματα. Αυτά τα στοιχεία υπάρχουν, έστω και σε σκελετική μορφή, στον κορμό της αρχιτεκτονικής. Μπορεί να προσθέσουμε εναλλακτικά υποσυστήματα, αρκεί βέβαια να είναι παρόμοια με αυτά που ήδη υπάρχουν. Για παράδειγμα αν υπάρχει ένα υποσύστημα για ένα πρωτόκολλο επικοινωνίας και προσθέσουμε ένα άλλο πρωτόκολλο επικοινωνίας που δεν χρειάζεται καινούργιες διασυνδέσεις, τότε είναι αποδεκτό να προσθέσουμε ένα καινούργιο υποσύστημα για εκείνο το πρωτόκολλο επικοινωνίας.

Βελτιώνουμε την αρχιτεκτονική όψη του μοντέλου σχεδίασης και διάταξης ώστε να αντικατοπτρίζουν την πρόοδο κατά τη φάση της κατασκευής. Γενικά όμως ολοκληρώσαμε την αρχιτεκτονική στο τέλος της φάσης της κατασκευής οπότε αυτό που κάνουμε σε αυτή τη φάση είναι να την ενημερώνουμε. Πρέπει να πούμε ότι η σχεδίαση είναι η κύρια εστίαση της φάσης της κατασκευής, όπως και η υλοποίηση βέβαια. Το αποτέλεσμα της είναι το μοντέλο σχεδίασης και διάταξης. Το μοντέλο σχεδίασης είναι το σχέδιο θα μπορούσαμε να πούμε του μοντέλου υλοποίησης και ίδιας της υλοποίησης.

#### 4.3.4.4 ΥΛΟΠΟΙΗΣΗ

Αυτή η σειρά εργασιών υλοποιεί και εκτελεί ελέγχους όλων των συστατικών, και βασίζεται κυρίως στο μοντέλο ανάλυσης. Το αποτέλεσμα μετά από κάποιο αριθμό επαναλήψεων, εκτός από την ενοποίηση του συστήματος και τον έλεγχο, είναι μια αρχική λειτουργική έκδοση, που αντιπροσωπεύει το 100% του όγκου των περιπτώσεων χρήσης. Στην συνέχεια καλύπτουμε τις εργασίες της αρχιτεκτονικής υλοποίησης, της υλοποίησης μιας κλάσης και της υλοποίησης ενός υποσυστήματος, του ελέγχου μιας μονάδας, και της ενοποίησης του συστήματος. Σε αυτή τη σειρά εργασιών είναι που το έργο απαιτεί την περισσότερη εργασία κατά τη φάση της κατασκευής, που είναι η κατασκευή των συστατικών. Γεμίζουμε κάθε συστατικό με όλο και περισσότερο κώδικα, επανάληψη με επανάληψη, μέχρι ώπου στο τέλος της φάσης της κατασκευής όλα τα συστατικά είναι συμπληρωμένα.

- Αρχιτεκτονική υλοποίηση. Μέχρι τώρα η αρχιτεκτονική του συστήματος θα έχει δημιουργηθεί με ακρίβεια. Εκτός από την συνεχόμενη επίβλεψη, αυτό που έχουμε να κάνουμε είναι να την ενημερώνουμε όποτε χρειάζεται.
- Υλοποίηση κλάσης και υλοποίηση υποσυστήματος. Υλοποιούμε τις κλάσεις και τα υποσυστήματα στο μοντέλο υλοποίησης.
- Εκτέλεση ελέγχου μονάδας. Έχουμε την ευθύνη της εκτέλεσης ενός ελέγχου της μονάδας ενός συστατικού που κατασκευάζουμε. Μετά διορθώνουμε την σχεδίαση και υλοποίηση του συστατικού αν χρειάζεται.
- Ενοποίηση συστήματος. Δημιουργούμε ένα πλάνο ενοποίησης των κατασκευών που περιγράφει τη σειρά των κατασκευών. Το πλάνο διατυπώνει τις περιπτώσεις χρήσης ή τα σενάρια μέσα στις περιπτώσεις χρήσης που η κατασκευή θα υλοποιήσει. Αυτές οι περιπτώσεις χρήσης και τα σενάρια είναι που οδηγούν στα υποσυστήματα και στα συστατικά.

#### 4.3.4.5 ΔΟΚΙΜΗ

Οι προσπάθειες μας να ανακαλύψουμε το τι μπορεί να δοκιμαστεί αποτελεσματικά και να επινοήσουμε περιπτώσεις δοκιμών και διαδικασίες δοκιμών έρχονται στην επιφάνεια στη φάση της κατασκευής. Είναι μια κύρια ενέργεια σε αυτή τη φάση όπως φαίνεται και στην εικόνα .

- Δημιουργία πλάνου ελέγχου. Επιλέγουμε στόχους με βάση των οποίων θα ελεγχθούν οι διαδοχικές κατασκευές και τελικώς το ολόκληρο σύστημα.
- Δημιουργία σχεδίου ελέγχου. Βρίσκουμε πώς να δοκιμάσουμε τις περιπτώσεις χρήσης που υλοποιούνται από μία κατασκευή με σκοπό να επιβεβαιώσουμε τις υπό δοκιμή περιπτώσεις χρήσης. Ετοιμάζουμε περιπτώσεις και διαδικασίες δοκιμών για αυτό το σκοπό. Από τις περιπτώσεις και διαδικασίες δοκιμών προηγούμενων κατασκευών επιλέγουμε αυτές που είναι σχετικές και τις τροποποιούμε για επαναχρησιμοποίηση σε επόμενες κατασκευές. Επιβεβαιώνουμε τα συστατικά που είναι να ελεγχθούν μαζί, όπως αρχικά είχαμε κάνει στο πλάνο ελέγχου. Ο σκοπός αυτών των ενοποιημένων ελέγχων είναι να επιβεβαιώσει τη σωστή λειτουργία των διασυνδέσεων μεταξύ των συστατικών

που υπόκεινται στον έλεγχο και το γεγονός ότι τα συστατικά δουλεύουν σωστά μεταξύ τους.

- Εκτέλεση ελέγχων ενοποίησης. Εκτελούμε τις περιπτώσεις δοκιμών και στη συνέχεια τις διαδικασίες δοκιμών. Όταν η κατασκευή ανταπεξέλθει στις δοκιμές τότε προσθέτουμε επιπλέον κατασκευές καθώς γίνονται διαθέσιμες και συνεχίζουμε να τις δοκιμάζουμε. Αν βρεθεί κάποιο σφάλμα κατά τη δοκιμή τότε σχεδιάζουμε το επόμενο βήμα. Το επόμενο βήμα μπορεί να είναι για παράδειγμα περαιτέρω εργασία μέσα στην ίδια την κατασκευή, ή στην περίπτωση σοβαρού σφάλματος ανάθεση σε ειδικευμένα άτομα να το διερευνήσουν.
- Εκτέλεση ελέγχων συστήματος. Την στιγμή που οι διαδοχικές κατασκευές φτάνουν στο τέλος μιας επανάληψης, τότε έφτασαν σε κατάσταση ενός μερικού συστήματος και εισέρχονται στη διαδικασία ελέγχου. Εκτελούμε τις περιπτώσεις δοκιμών του συστήματος, και στη συνέχεια τις διαδικασίες δοκιμών του συστήματος. Στο τέλος της τελευταίας επανάληψης της φάσης της κατασκευής, δοκιμάζουμε την αρχική λειτουργική έκδοση, και λαμβάνουμε τα αποτελέσματα τα οποία μελετάμε διεξοδικά.
- Αξιολόγηση ελέγχων. Καθώς προχωρούν οι δοκιμές της ενοποίησης και του συστήματος, εξετάζουμε τα αποτελέσματα των δοκιμών κάθε κατασκευής με βάση τους στόχους που αρχικά είχαμε θέση στο πλάνο ελέγχων, οι οποίοι όμως πιθανώς να έχουν μεταβληθεί από τις διαδοχικές επαναλήψεις. Ο σκοπός της αξιολόγησης μιας δοκιμής είναι να εγγυηθεί ότι ο έλεγχος πέτυχε τους στόχους του. Αν ένα έλεγχος δεν πετύχει τους στόχους του, τότε οι περιπτώσεις δοκιμών και οι διαδικασίες δοκιμών πρέπει να τροποποιηθούν για να πετύχουν τους στόχους του ελέγχου.

#### **4.3.5 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΚΑΙ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ**

Στο πλαίσιο της επανεξέτασης των αποτελεσμάτων των δοκιμών και άλλων κριτηρίων αξιολόγησης κάνουμε τις ακόλουθες ενέργειες:

- Εξέταση του τι έχει επιτευχθεί σε μια επανάληψη σε σχέση με αυτά που είχαν καταγραφεί στο πλάνο της επανάληψης.
- Δημιουργία πλάνου στο οποίο θα περιγράψουμε σε ποια από τις ακόλουθες επαναλήψεις θα γίνει κάποια εργασία που δεν ολοκληρώθηκε.
- Λήψη της απόφασης ότι η κατασκευή είναι έτοιμη να προχωρήσει στην επόμενη επανάληψη.
- Ενημέρωση της λίστας κινδύνων.
- Συμπλήρωση του πλάνου της επόμενης επανάληψης.
- Στο τέλος της τελευταίας επανάληψης αυτής της φάσης λήψη της απόφασης ότι το προϊόν έχει περάσει τις δοκιμές συστήματος και έχει φτάσει σε κατάσταση αρχικής λειτουργικής ικανότητας.
- Έγκριση της εισόδου στη φάση της μετάβασης.
- Ενημέρωση του πλάνου του έργου.

#### 4.3.6 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΛΑΝΟΥ ΤΗΣ ΜΕΤΑΒΑΣΗΣ

Δεν μπορούμε να περιμένουμε να δημιουργήσουμε το πλάνο της φάσης της μετάβασης με τόσες λεπτομέρειες εξαρχής, όπως κάναμε στις προηγούμενες φάσεις. Γνωρίζουμε ότι θα διαθέσουμε τις δοκιμαστικές εκδόσεις σε επιλεγμένους χρήστες για αξιολόγηση. Αυτό το μέρος της φάσης της μετάβασης, όπως η επιλογή των χρηστών-δοκιμαστών και η ετοιμασία των οδηγιών δοκιμής, είναι που μπορούμε να σχεδιάσουμε με κάποιες λεπτομέρειες. Οι απόψεις των χρηστών που λαμβάνουμε, αφού δοκιμάσουν την δοκιμαστική έκδοση, όπως κίνδυνοι, προβλήματα, σφάλματα και προτάσεις, δεν μπορούμε να τα γνωρίζουμε εξαρχής. Αν έχουμε κάποια εμπειρία σχετικά με δοκιμαστικές εκδόσεις θα γνωρίζουμε ως κάποιο βαθμό τι να περιμένουμε. Θα είμαστε σε θέση να υπολογίσουμε τον σε προσέγγιση το αριθμό των εμπειρων ατόμων που θα χρειαστούμε για να αντιμετωπίσουμε τα προβλήματα που οι χρήστες-δοκιμαστές θα φέρουν στην επιφάνεια.

#### 4.3.7 ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ

Τα παραγόμενα της φάσης της κατασκευής είναι τα εξής:

- Ένα πλάνο για την φάση της μετάβασης.
- Ένα εκτελέσιμο λογισμικό που είναι η αρχική έκδοση με λειτουργική ικανότητα. Αυτή είναι η τελική κατασκευή από την φάση της κατασκευής.
- Όλα τα μοντέλα του συστήματος.
- Συντηρημένη και ελάχιστα ενημερωμένη αρχιτεκτονική περιγραφή.
- Προκαταρκτικά εγχειρίδια χρήσης τα οποία θα είναι με τόσες λεπτομέρειες, όσες χρειάζονται για να καθοδηγηθούν οι χρήστες-δοκιμαστές.
- Επιχειρηματικό πλάνο, που αντικατοπτρίζει την κατάσταση στο τέλος της φάσης.

Πρόθεση μας είναι τα παραγόμενα που έχουν ονοματιστεί ολοκληρωμένα να είναι και στην πραγματικότητα έτσι. Τώρα η χρήση του λογισμικού από την κοινότητα των χρηστών στη φάση της μετάβασης μπορεί να αποκαλύψει ότι κάποια από αυτά δεν είναι πραγματικά σωστά. Τότε θα τροποποιηθούν για να γίνουν σωστά.

#### 4.4 ΜΕΤΑΒΑΣΗ

Καθώς το έργο εισέρχεται στη φάση της μετάβασης, το σύστημα έχει αποκτήσει μια αρχική λειτουργική ικανότητα. Θεωρούμε ότι το σύστημα είναι αξιόπιστο για να λειτουργήσει μέσα στο περιβάλλον του χρήστη, αν και χωρίς να είναι τέλειο. Για παράδειγμα κάποια προβλήματα, κίνδυνοι και ελαττώματα δεν έγιναν ορατά κατά τον έλεγχο του συστήματος στο τέλος της φάσης της κατασκευής, μπορεί να εμφανιστούν στο περιβάλλον του χρήστη. Μπορεί να υπάρχουν χαρακτηριστικά τα οποία οι χρήστες καθυστερημένα ανακάλυψαν ότι τα χρειάζονται. Αν είναι πολύ σημαντικά και ταιριάζουν με το υπάρχων προϊόν, μπορούμε να συμφωνήσουμε να τα προσθέσουμε. Ωστόσο οι αλλαγές πρέπει να είναι μικρές, ώστε να μπορούν να ενσωματωθούν χωρίς να



επηρεάσουν πολύ το πλάνο του έργου. Αν η προτεινόμενη αλλαγή επηρεάζει το χρονοδιάγραμμα, τότε η ανάγκη για αυτή πρέπει να είναι πολύ έντονη, ώστε να δικαιολογηθεί η εφαρμογή της. Στις περισσότερες περιπτώσεις, πιστεύουμε ότι είναι καλύτερο να τις προσθέσουμε στη λίστα των χαρακτηριστικών και να τις μετατοπίσουμε για τον επόμενο κύκλο ανάπτυξης, που είναι η επόμενη έκδοση του συστήματος. Οι βασικές προθέσεις αυτής της φάσης είναι οι εξής:

- Να ικανοποιήσουμε τις απαιτήσεις, όπως καταγράφηκαν στις προηγούμενες φάσεις, προς ικανοποίηση όλων των εμπλεκομένων στην ανάπτυξη του συστήματος.
- Ο χειρισμός όλων των θεμάτων που χρειάζονται για την λειτουργία του συστήματος στο περιβάλλον του χρήστη, συμπεριλαμβανομένων των διορθώσεων των ελαττωμάτων που αναφέρθηκαν από τους χρήστες δοκιμαστές.

#### **4.4.1 Η ΦΑΣΗ ΤΗΣ ΜΕΤΑΒΑΣΗΣ ΓΕΝΙΚΑ**

Αυτή η φάση εστιάζει στην εγκατάσταση του προϊόντος στο περιβάλλον λειτουργίας του. Ο τρόπος με τον οποίο πραγματοποιούμε αυτή την εστίαση ποικίλει ανάλογα με τη φύση της σχέσης του προϊόντος με την αγορά. Για παράδειγμα, αν το προϊόν είναι να βγει στην αγορά, τότε θα διανέμουμε τη δοκιμαστική έκδοση σε τυπικούς χρήστες που βρίσκονται σε αντιπροσωπευτικούς χώρους. Ενώ αν το προϊόν είναι να πάει σε ένα μόνο πελάτη τότε εγκαθιστούμε το προϊόν μόνο στο συγκεκριμένο χώρο. Παρατηρούμε τις απόψεις από τους διαφορετικούς χώρους για να:

- Βρούμε αν όντως το προϊόν κάνει αυτά που οι εταιρίες και οι χρήστες τους ζητήσανε.
- Ανακαλύψουμε μη αναμενόμενους κινδύνους.
- Σημειώσουμε άλυτα προβλήματα.
- Βρούμε σφάλματα.
- Διορθώσουμε αμφιβολίες και κενά στο εγχειρίδιο χρήσης.
- Εστιάσουμε σε σημεία που οι χρήστες φαίνονται ανεπαρκής ή χρειάζονται πληροφορίες και εκπαίδευση.

Με βάση τις επιστρεφόμενες απόψεις αυτού του είδους, τροποποιούμε το σύστημα ή τα σχετικά παραγόμενα. Ετοιμαζόμαστε για την παραγωγή του προϊόντος, την ετοιμασία των πακέτων, την διάταξη και γενικά την πρώτη παρουσίαση του προϊόντος. Σε αυτή τη φάση δεν επιζητούμε την αναμόρφωση του προϊόντος. Έχουμε ενσωματώσει μαζί με τους πελάτες σημαντικές αλλαγές στις απαιτήσεις σε προηγούμενες φάσεις. Αντιθέτως, αναζητούμε μικρές ατέλειες οι οποίες αγνοήθηκαν στη φάση της κατασκευής και οι οποίες μπορούν να διορθωθούν με την υπάρχουσα δομή. Στη σχέση μας με τον πελάτη μπορεί να παράσχουμε βοήθεια στην εγκατάσταση του κατάλληλου περιβάλλοντος για το προϊόν και στην εκπαίδευση των χρηστών για την παραγωγική χρήση του προϊόντος.

#### **4.4.2 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΛΑΝΟΥ ΤΗΣ ΜΕΤΑΒΑΣΗΣ**

Όπως είπαμε είναι δύσκολο να περιμένουμε να κάνουμε το πλάνο της φάσης της μετάβασης εξαρχής με πολλές λεπτομέρειες όπως κάναμε στην φάση της κατασκευής.

Από τη μια πλευρά ξέρουμε ότι θα διανείμουμε την δοκιμαστική έκδοση, που δημιουργήθηκε κατά τη φάση της κατασκευής, σε επιλεγμένους χρήστες. Αυτή η γνώση παρέχει τη βάση για το αρχικό πλάνο της φάσης της μετάβασης. Υπάρχει ένα γνωστό ποσό εργασίας που χρειάζεται για την παραγωγή των δοκιμαστικών εκδόσεων, για την ετοιμασία της δοκιμαστικής τεκμηρίωσης, και για την επιλογή των χρηστών-δοκιμαστών. Από την άλλη πλευρά θα υπάρχει ένα άγνωστο ποσό εργασίας, που θα προκύψει μετά την καταγραφή των απόψεων και των γνώμων των χρηστών-δοκιμαστών από τη δοκιμή του προϊόντος. Θα πρέπει να είμαστε σε ετοιμότητα για να αντιμετωπίσουμε τα θέματα που θα παρουσιαστούν. Στην δημιουργία του πλάνου της μετάβασης, περιμένουμε ότι η αρχική λειτουργική έκδοση από τη φάση της κατασκευής θα χρειαστεί λίγη αναθεώρηση, ως αποτέλεσμα των επισημάνσεων των χρηστών-δοκιμαστών. Συγκεκριμένα αν το έργο έχει διεξαχθεί με επαναληπτική λογική, τότε αυτό είναι το μεγάλο πλεονέκτημα. Αυτή η διαδικασία ανάπτυξης επιτρέπει τους κατασκευαστές να πειραματιστούν στις αρχικές επαναλήψεις, και να βρουν τα λάθη τους στη σύλληψη του συστήματος στις δοκιμές και στους ελέγχους των αρχικών επαναλήψεων και στην παρατήρηση της λειτουργίας τους. Παρομοίως, πολλά λάθη θα εντοπιστούν και θα διορθωθούν επανάληψη με επανάληψη καθώς προχωράει η διαδικασία. Εν συντομία, η πρόωγη αναθεώρηση είναι καλή. Στη φάση της μετάβασης και μέσα στο πλαίσιο της επαναληπτικής διαδικασίας, η αναθεώρηση θα είναι ελάχιστη. Αλλά θα πρέπει να υπολογίζουμε ότι δεν θα είναι και μηδαμινή. Τουλάχιστον μερικές παραλείψεις και σφάλματα θα περάσουν από τους ελέγχους. Η προκατάληψη ενός έργου να αποδοκιμάζει την πιθανότητα αναθεώρησης μπορεί να οφείλεται στα εξής:

- Στην υπερβολική πίεση από το χρονοδιάγραμμα, που θα οδηγήσει στην δημιουργία σφαλμάτων.
- Στην απουσία μια ικανοποιητικής δοκιμής συστήματος και αποτίμησης στο τέλος της φάσης της κατασκευής.
- Η αποτυχία εστίασης στην αξιοσημείωτη εργασία που απομένει στη φάση της μετάβασης.
- Το αίσθημα ότι μελετώντας την ανάγκη για αναθεώρηση θα κάνει αυτή την ανάγκη πραγματικότητα.
- Η διάθεση να εκτιμήσουμε την αναθεώρηση ως «κακή», ως παραδοχή της ανικανότητας του έργου.

Το θέμα του «αρκετά καλού» λογισμικού μπορεί να προκύψει κατά τη δημιουργία του πλάνου της φάσης της μετάβασης. Είναι γεγονός ότι στη ζωή κανένα λογισμικό προϊόν δεν είναι τέλειο. Για παράδειγμα κάποια προϊόντα παραδίδονται με κάποιο ποσοστό ατελειών, με κάποιες απαιτήσεις αναβληθείς για μια επόμενη έκδοση, ή με κάποιες ανάγκες που ανακαλύφθηκαν από τους χρήστες-δοκιμαστές και η φάση της μετάβασης αδυνατεί να τις ικανοποιήσει λόγω έλλειψης πόρων. Υπάρχουν τρεις απαντήσεις στο θέμα του «αρκετά καλού» λογισμικού. Πρώτα από όλα, οι φάσεις και οι επαναλήψεις της ενοποιημένης διεργασίας έχουν σκοπό να αναγνωρίσουν κινδύνους, να καταγράψουν ακριβείς απαιτήσεις, και να δημιουργήσουν ανάλογα το πλάνο του έργου. Στην ενοποιημένη διεργασία καταβάλουμε αυτές τις προσπάθειες σε συνεργασία με τους χρήστες και τους πελάτες. Συνεπώς, η αρχική λειτουργική έκδοση, ή δοκιμαστική έκδοση, θα πρέπει να προσεγγίζει αυτό που και οι κατασκευαστές αλλά και οι πελάτες επιθυμούν. Η δεύτερη απάντηση είναι ότι αφού ούτε οι κατασκευαστές αλλά ούτε και οι πελάτες περιμένουν η αρχική λειτουργική έκδοση να είναι χωρίς ατέλειες, έχουν

κρατήσει πόρους για τη φάση της μετάβασης. Η τελευταία απάντηση είναι ότι αφού έχουν συμφωνήσει οι κατασκευαστές και οι πελάτες για τις δύο πρώτες απαντήσεις τότε μπορεί να προκύψει μια επέκταση της φάσης της μετάβασης ή μια αναβολή της αναπάντεχης εργασίας για τον επόμενο κύκλο ανάπτυξης.

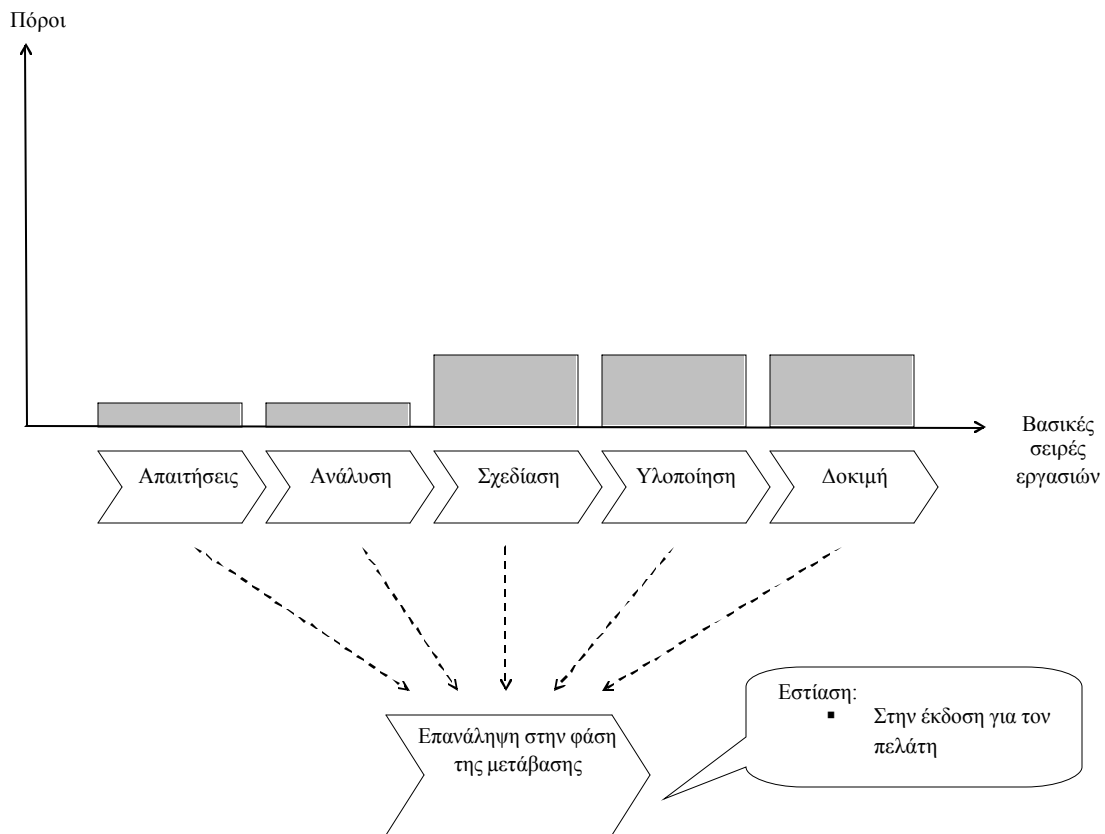
#### **4.4.3 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΚΡΙΤΗΡΙΩΝ ΑΞΙΟΛΟΓΗΣΗΣ**

Στο τέλος της φάσης της κατασκευής με τον έλεγχο του συστήματος, το προϊόν πιστεύουμε ότι αποκτά μια αρχική λειτουργική ικανότητα ή με άλλα λόγια ικανοποιεί τις περιγραφές των απαιτήσεων. Για τη φάση της μετάβασης είναι απαραίτητο να αξιολογήσουμε μόνο τα θέματα που θα προκύψουν σε αυτή τη φάση. Βασικά υπάρχουν πέντε θέματα:

1. Κάλυψαν οι χρήστες-δοκιμαστές όλες τις βασικές λειτουργίες του συστήματος, αυτές δηλαδή που περιλαμβάνονται στη πετυχημένη λειτουργία του προϊόντος;
2. Παρομοίως, πέρασε το προϊόν τις δοκιμές αποδοχής του πελάτη; Τα κριτήρια δοκιμών θέτονται από το συμβόλαιο, το οποίο έχει συμφωνηθεί και από τους κατασκευαστές και από τον πελάτη. Επίσης οι δοκιμές αποδοχής τρέχουν το λογισμικό για μια συμφωνημένη χρονική περίοδο.
3. Είναι το βοηθητικό υλικό του χρήστη αποδεκτής ποιότητας;
4. Είναι τα απαραίτητα εγχειρίδια χρήσης έτοιμα προς χρήση;
5. Είναι οι πελάτες και οι χρήστες ικανοποιημένοι από το προϊόν;

#### **4.4.4 ΟΙ ΒΑΣΙΚΕΣ ΣΕΙΡΕΣ ΕΡΓΑΣΙΩΝ ΣΤΗ ΦΑΣΗ ΤΗΣ ΜΕΤΑΒΑΣΗΣ**

Η εργασία και στις πέντε βασικές σειρές εργασιών είναι μικρή σε αυτή τη φάση όπως φαίνεται και στην εικόνα 14. Επειδή σχεδόν όλη η εργασία έγινε στη φάση της κατασκευής, το μέγεθος της εργασίας σε αυτή τη φάση είναι μικρό, και τόσο όσο χρειάζεται για να διορθωθούν τα προβλήματα που βρέθηκαν κατά τη δοκιμή του προϊόντος στο περιβάλλον λειτουργίας του. Ωστόσο αυτό δεν σημαίνει ότι δεν υπάρχει σημαντική εργασία σε αυτή τη φάση. Δεν θα πρέπει να υπάρχει καθόλου για τις σειρές εργασιών των απαιτήσεων και της ανάλυσης. Οι εργασίες της σχεδίασης τυπικά υποχωρούν κατά τη φάση της μετάβασης, και σε κάθε περίπτωση γενικά περιλαμβάνουν μικρές σχεδιαστικές βελτιώσεις, που είναι απαραίτητες για να διορθωθούν τα προβλήματα ή οι ατέλειες ή για να πραγματοποιηθούν κάποιες μικρές δευτερεύουσες προσθήκες της τελευταίας στιγμής. Η εστίαση μετατοπίζεται προς την πλευρά της διόρθωσης ατελειών για να εξαλειφθούν τα σφάλματα που εμφανίζονται κατά τη δοκιμαστική χρήση, στην επιβεβαίωση ότι οι διορθώσεις είναι οι πρόπουσες, και στην επανάληψη των δοκιμών για είμαστε σίγουροι ότι οι διορθώσεις δεν προκάλεσαν άλλα σφάλματα.



Εικ. 14 Γενική επανάληψη στη φάση της μετάβασης

Μια γενική επανάληψη στη φάση της μετάβασης αποτελείται από τις πέντε βασικές σειρές εργασιών. Γενικότερα εκτελούμε τέσσερα σύνολα εργασιών μερικώς παράλληλα. Ένα είναι οι πέντε βασικές σειρές εργασιών, το δεύτερο είναι ο σχεδιασμός των επαναλήψεων, το τρίτο είναι η περαιτέρω μελέτη του επιχειρηματικού πλάνου, και το τέταρτο είναι η αποτίμηση.

#### 4.4.5 ΕΝΕΡΓΕΙΕΣ ΣΤΗ ΦΑΣΗ ΤΗΣ ΜΕΤΑΒΑΣΗΣ

Καθώς ένα έργο περνάει από τη φάση της μετάβασης εκτελούμε τις εξής ενέργειες:

- Ετοιμάζουμε τη δοκιμαστική έκδοση από την αρχική λειτουργική έκδοση που δημιουργήθηκε στο τέλος της φάσης της κατασκευής.
- Εγκαθιστούμε αυτή την έκδοση στο χώρο που προβλέπεται, και κάνουμε κάποιες επιπλέον ενέργειες στο συγκεκριμένο χώρο, όπως τη μεταφορά των δεδομένων από το προηγούμενο σύστημα.
- Ενεργούμε αναλόγως μετά την αναφορά των προβλημάτων και σφαλμάτων από τους χρήστες-δοκιμαστές.
- Προσαρμόζουμε το διορθωμένο προϊόν στις συνθήκες του χρήστη.
- Συμπληρώνουμε τα παραγόμενα του έργου.
- Λαμβάνουμε την απόφαση πότε το έργο τελειώνει.

Αυτή η σειρά των ενεργειών θα ποικίλει ανάλογα με το αν κατασκευάζουμε ένα προϊόν για την αγορά ή για συγκεκριμένο πελάτη. Στην πρώτη περίπτωση θα υπάρξουν πολλοί πιθανοί χρήστες, και η επιλογή και καθοδήγηση των χρηστών-δοκιμαστών θα είναι μια σημαντική εργασία. Εκτός αυτού οι μεμονωμένοι χρήστες δεν ακολουθούν ένα προδιαγεγραμμένο σχέδιο δοκιμής. Χρησιμοποιούν το προϊόν όπως θέλουν και αναφέρουν ότι βρίσκουν. Στη δεύτερη περίπτωση, ο πελάτης πιθανώς να επιλέξει ένα χώρο για την αρχική εγκατάσταση και η δοκιμή αποδοχής θα ακολουθήσει μια επίσημη, συμφωνημένη, και συστηματική διαδικασία. Τα αποτελέσματα που θα αναφερθούν θα είναι πιθανώς παρεκκλίσεις από τις επίσημες προδιαγραφές. Αν προκύψουν προβλήματα που είναι πέρα από την εμβέλεια της υπάρχουσας συμφωνίας, τότε οι δύο πλευρές προβαίνουν σε μια επιπλέον συμφωνία. Επίσης η μορφή των ενεργειών αυτών θα ποικίλει ανάλογα με το αν το προϊόν λογισμικού είναι ένα καινούργιο προϊόν ή μια βελτίωση ενός υπάρχοντος προϊόντος. Είναι στην επιλογή μας να προσαρμόσουμε τη διαδικασία στις συνθήκες που αντιμετωπίζουμε.

Οι περισσότεροι από τους αρχικούς χρήστες για την χρήση του δοκιμαστικού συστήματος θα είναι έμπειροι. Οι κατασκευαστές τους τροφοδοτούν με συγκεκριμένες οδηγίες για το πώς να αναφέρουν τα ευρήματα των δοκιμών και παρατηρήσεων τους. Συμπληρώνουμε τη τεκμηρίωση του προϊόντος με συγκεκριμένες οδηγίες για τη δοκιμή του προϊόντος. Επιλέγουμε τους χρήστες-δοκιμαστές και διανέμουμε τη δοκιμαστική έκδοση και τα συνοδευτικά υλικά σε αυτούς.

Οι ενέργειες στο χώρο εγκατάστασης διαφέρουν μεταξύ των δοκιμών χρήσης και των δοκιμών αποδοχής. Πιθανόν να υπάρχει μεγάλος αριθμός χώρων δοκιμής, στους οποίους δεν παρευρίσκονται οι κατασκευαστές. Πρέπει να επισημάνουν συγκεκριμένες οδηγίες για το πώς θα εγκατασταθεί το λογισμικό, πώς θα λειτουργήσει, σε τι θέματα θα πρέπει να εστιάσουν την προσοχή τους οι χρήστες-δοκιμαστές και πώς να αναφέρουν τα ελαττώματα και τα άλλα προβλήματα που βρουν. Αν η έκδοση είναι μια αναβάθμιση ή αντικατάσταση ενός προηγούμενου λογισμικού, οι κατασκευαστές πρέπει να παράσχουν οδηγίες για το πώς οι χρήστες-δοκιμαστές θα μεταφέρουν τα δεδομένα ή θα μετατρέψουν τις βάσεις δεδομένων στην καινούργια έκδοση. Οι οδηγίες μπορεί να αναφέρονται και στο πώς θα υπάρξει παράλληλη λειτουργία και του δοκιμαστικού συστήματος άλλα και του προηγούμενου για κάποια χρονική περίοδο. Αντιθέτως, στις δοκιμές αποδοχής,

άτομα από τους κατασκευαστές θα είναι πιθανώς παρών. Θα υπάρχει ένα επίσημο κείμενο δοκιμής αποδοχής. Σφάλματα και προβλήματα θα διορθωθούν εκείνη την ώρα αν είναι εφικτό ή θα μεταφερθούν στην κατασκευαστική ομάδα.

Οι κατασκευαστές συλλέγουν και αναλύουν τα αποτελέσματα των δοκιμών για να ενεργήσουν αναλόγως. Τα αποτελέσματα είναι πιθανό να χωρίζονται σε δύο κατηγορίες. Στην πρώτη ανήκουν σχετικά δευτερεύοντα σφάλματα κώδικα που απλά θα πρέπει να εξεταστούν και να διορθωθούν, παρόλο που μπορεί να είναι δύσκολο μερικές φορές. Στην δεύτερη κατηγορία ανήκουν πιο σημαντικά προβλήματα με μεγαλύτερο εύρος.

Κάποια από τα προβλήματα που βρίσκονται κατά τη χρήση του δοκιμαστικού προϊόντος μπορεί να χρειάζονται κάτι περισσότερο από μια απλή διόρθωση σφάλματος. Για παράδειγμα, αυτό το περισσότερο μπορεί να είναι μια επιπλέον επανάληψη. Όπως έχουμε αναφέρει και πριν, σημαντικές αλλαγές οι οποίες απαιτούν παραπάνω πόρους από ότι διαθέτουμε ή καθυστερήσουν την παράδοση του έργου ή απαιτούν αλλαγή αρχιτεκτονικής θα πρέπει να μετατοπιστούν για τον επόμενο κύκλο ανάπτυξης. Παραμένει σημαντικό να συντηρούμε την αρχιτεκτονική ακεραιότητα του συστήματος ενώ διορθώνουμε προβλήματα και σφάλματα. Πρέπει να είμαστε σίγουροι ότι τα προβλήματα δεν διορθώνονται με τέτοιο τρόπο ώστε να επηρεάζεται η αρχιτεκτονική.

Η φάση της μετάβασης δεν τελειώνει μόνο όταν όλες οι εργασίες και τα παραγόμενα έχουν συμπληρωθεί, αλλά και όταν ο πελάτης είναι ικανοποιημένος. Το πότε είναι ικανοποιημένος εξαρτάται κάθε φορά από τη σχέση με την αγορά. Στην περίπτωση προϊόντων για την αγορά, συμπεραίνουμε ότι η πλειονότητα των χρηστών θα είναι ικανοποιημένοι όταν έχουμε ανταποκριθεί στις αναφορές των χρηστών-δοκιμαστών. Στην περίπτωση προϊόντος για ένα συγκεκριμένο πελάτη, συμπεραίνουμε ότι ο πελάτης θα είναι ικανοποιημένος όταν το σύστημα περάσει τις δοκιμές αποδοχής.

#### **4.4.6 ΑΠΟΤΙΜΗΣΗ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΜΕΤΑΒΑΣΗΣ**

Αυτή η αποτίμηση διαφέρει από εκείνες των προηγούμενων φάσεων για δύο λόγους κυρίως. Πρώτον γιατί είναι η τελευταία φάση, και δεν υπάρχει επόμενη φάση στην οποία θα μεταφερθούν οι εργασίες. Ωστόσο για ένα σύστημα μεγάλου μεγέθους θα υπάρχει κάποια ομάδα στην οποία θα μεταφερθούν κάποιες πολύτιμες ιδέες που αποκτήθηκαν κατά τη διάρκεια του έργου. Δεύτερον παρόλο που είναι η τελευταία φάση του τωρινού κύκλου ανάπτυξης, μπορεί πιθανώς να υπάρξουν και επόμενοι κύκλοι ανάπτυξης. Θα πρέπει να καταγράψουμε τυχόν ευρήματα που θα αποδειχθούν χρήσιμα στο μέλλον. Αυτά τα ευρήματα χωρίζονται σε δύο κατηγορίες όπως θα δούμε παρακάτω.

Από τη μια πλευρά, αν έχουμε εκτελέσει τις τρεις πρώτες φάσεις αποτελεσματικά, η φάση της μετάβασης θα εκτελεστεί ομαλά και θα ολοκληρωθεί εντός του χρονοδιαγράμματος και των πόρων που διατέθηκαν. Ο έλεγχος της δοκιμαστικής έκδοσης ανακαλύπτει συνήθως συνηθισμένα σφάλματα που εύκολα μπορούμε να διορθώσουμε. Επίσης κατά την αποτίμηση βρίσκουμε και κάποιες λίγες πληροφορίες που θα είναι πολύτιμες για τον πιθανό επόμενο κύκλο ανάπτυξης. Από την άλλη πλευρά, αν

αποτύχαμε να αναγνωρίσουμε όλους τους σημαντικούς κινδύνους, αποτύχαμε να σχεδιάσουμε την αρχιτεκτονική που θα ικανοποιεί τις απαιτήσεις, ή αποτύχαμε να υλοποιήσουμε μια σχεδίαση που θα μας παρέχει το προτεινόμενο σύστημα, τέτοιες ανεπάρκειες θα γίνουν επώδυνα φανερές στη φάση της μετάβασης. Ως αποτέλεσμα αυτών των ατελειών μπορεί να χρειαστεί να επεκτείνουμε τη φάση της μετάβασης μέχρι να φτάσουμε τουλάχιστον σε ένα ικανοποιητικό σύστημα. Αυτές οι ανεπάρκειες φυσικά συσσωρεύονται στη φάση της μετάβασης. Οπότε είναι σαν να κάνουμε αποτίμηση όλου του έργου.

#### **4.4.7 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΛΑΝΟΥ ΤΗΣ ΕΠΟΜΕΝΗΣ ΕΚΔΟΣΗΣ**

Η εμπειρία πολλών δεκαετιών έχει δείξει ότι λίγα είναι τα προϊόντα λογισμικού που διαρκούν αρκετά χωρίς να βελτιωθούν. Το υλικό και τα λειτουργικά πάνω στα οποία τρέχουν αναπτύσσονται περαιτέρω. Το επιχειρηματικό περιβάλλον μεταβάλλεται. Το λογισμικό αποκτά εφαρμογές πάνω σε όλο και περισσότερα μέρη πεδίων, όπως της βιομηχανίας, της έρευνας και της εκπαίδευσης. Σχεδόν κάθε προϊόν λογισμικού εισέρχεται πολύ σύντομα σε καινούργιο κύκλο ανάπτυξης. Ο καινούργιος κύκλος επαναλαμβάνει τις φάσεις της σύλληψης, επεξεργασίας, κατασκευής και μετάβασης.

#### **4.4.8 ΠΑΡΑΓΟΜΕΝΑ ΤΗΣ ΦΑΣΗΣ ΤΗΣ ΜΕΤΑΒΑΣΗΣ**

Τα παραγόμενα αυτής της φάσης είναι περίπου παρόμοια με αυτά της φάσης της κατασκευής, αλλά έχουν διορθωθεί και έχουν συμπληρωθεί:

- Το εκτελέσιμο λογισμικό, περιλαμβανομένου του λογισμικού εγκατάστασης.
- Έγγραφα όπως συμβόλαια, άδειες χρήσης και εγγυήσεις.
- Ολοκληρωμένη και διορθωμένη έκδοση περιλαμβανομένων όλων των μοντέλων του συστήματος.
- Ολοκληρωμένη και ενημερωμένη αρχιτεκτονική περιγραφή.
- Εγχειρίδια χρήσης και εκπαιδευτικό υλικό.
- Παραπομπές βοήθειας για τους πελάτες και παραπομπές για βοήθεια στο διαδίκτυο, όπου εκεί μπορούν να βρεθούν περισσότερες πληροφορίες, να γίνει αναφορά ελαττωμάτων, να βρεθούν πληροφορίες πιθανόν αναβαθμίσεων.

## **B ΕΝΟΠΟΙΗΜΕΝΗ ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ**

### **1 ΓΕΝΙΚΑ**

Η μοντελοποίηση έργων υπάρχει σε όλο το φάσμα των κατασκευαστικών δραστηριοτήτων του ανθρώπου. Θα ήταν αδιανόητη η κατασκευή μιας ηλεκτρικής συσκευής ή ενός αεροπλάνου χωρίς πριν να δημιουργηθούν κάποια μοντέλα του έργου. Η δημιουργία μοντέλων είναι απαραίτητη ώστε τα άτομα που εμπλέκονται στο έργο να μπορέσουν να καταλάβουν καλύτερα το έργο που θα αναπτύξουν, να επικοινωνήσουν και να ανταλλάξουν τις ιδέες τους με άλλα άτομα, και να επιβεβαιώσουν τις θεωρίες τους και να δοκιμάσουν καινούργιες με το μικρότερο ρίσκο. Με άλλα λόγια ένα μοντέλο είναι μια απλοποίηση της πραγματικότητας και επιτυγχάνονται τα εξής:

- Η οπτικοποίηση του συστήματος όπως είναι ή όπως θα θέλαμε να είναι.
- Ο καθορισμός της δομής και της συμπεριφοράς του συστήματος.
- Η δημιουργία ενός προτύπου που μας καθοδηγεί στην κατασκευή του συστήματος.
- Στην καταγραφή των αποφάσεων που έχουμε πάρει.

Έτσι και στον κόσμο του λογισμικού είναι απαραίτητη η δημιουργία μοντέλων για την ανάπτυξη λογισμικού. Η ενοποιημένη γλώσσα μοντελοποίησης UML αποτελεί μια γλώσσα απεικόνισης ή μοντελοποίησης ενός πληροφοριακού συστήματος βασισμένου σε αντικείμενα (αντικειμενοστραφούς συστήματος). Η UML είναι γλώσσα μοντελοποίησης και όχι μέθοδος. Οι περισσότερες μέθοδοι αποτελούνται, τουλάχιστον θεωρητικά, από μια διαδικασία και από μια γλώσσα μοντελοποίησης. Η διαδικασία είναι ο συμβολισμός της μεθόδου για τα βήματα που πρέπει να ακολουθηθούν στην υλοποίηση ενός σχεδίου ενώ η γλώσσα μοντελοποίησης είναι ο κυρίως γραφικός συμβολισμός που οι μέθοδοι χρησιμοποιούν ως μέσο έκφρασης των σχεδίων και είναι το πιο σημαντικό μέρος μιας μεθόδου. Είναι οπωσδήποτε το κλειδί για την επικοινωνία. Αν θέλετε να εξετάσετε το σχέδιο σας με κάποιον άλλον, αυτό που χρειάζεται να κατανοείται και οι δύο είναι η γλώσσα μοντελοποίησης. Χρησιμοποιούμε την UML γιατί μας επιτρέπει να μεταφέρουμε κάποιες έννοιες πιο ξεκάθαρα από τις άλλες εναλλακτικές μεθόδους. Η φυσική γλώσσα είναι πολύ ανακριβής και προκαλεί σύγχυση καθώς φτάνει κανείς σε πιο πολύπλοκες έννοιες. Ο κώδικας είναι πολύ ακριβής αλλά και πολύ λεπτομερής. Όπως σε όλα τα σύνθετα έργα, έτσι και στα έργα πληροφορικής η ανάγκη της μοντελοποίησης πριν την κατασκευή του συστήματος είναι επιτακτική. Η μοντελοποίηση ενός συστήματος παρέχει τη δυνατότητα της αφαίρεσης των ασήμαντων για αυτό λεπτομερειών και της εστίασης στις σημαντικές λεπτομέρειες του συστήματος που είναι απαραίτητο να κατανοηθούν πριν την κατασκευή του. Πρέπει να επισημάνουμε πως με την αφαίρεση δεν εννοούμε πως είμαστε ασαφείς ή ανούσιοι. Αντίθετα είμαστε πολύ σαφείς και ουσιαστικοί χωρίς να προσπαθούμε να μοντελοποιήσουμε ασήμαντα ή τετριμμένα σημεία του συστήματος. Προσπαθούμε να μοντελοποιήσουμε αρκετά ώστε να είμαστε σε θέση αρχικά να κατανοήσουμε και στη συνέχεια να κατασκευάσουμε το σύστημα. Προσπαθούμε να αποφύγουμε τις λεπτομέρειες και κυρίως να αναδείξουμε τις σημαντικές λεπτομέρειες. Επίσης, δίνει τη δυνατότητα του πειραματισμού με διαφορετικές λύσεις ή προσεγγίσεις για το ίδιο πρόβλημα. Καθιστά εφικτή τη



δυνατότητα ανάλυσης, σχεδίασης, καταγραφής και παρακολούθησης της προόδου ενός έργου πληροφορικής. Τέλος, προσφέρει μια κοινή γλώσσα για την επικοινωνία όσων εμπλέκονται στην κατασκευή του συστήματος. Χωρίς ένα μοντέλο δεν είναι δυνατόν να προσεγγίσει κανείς την πολυπλοκότητα των σύγχρονων πληροφοριακών συστημάτων. Η ενοποιημένη γλώσσα μοντελοποίησης (unified modeling language – UML) αποτελεί την πλέον δημοφιλή και πρότυπη γλώσσα για την οπτικοποίηση, προσδιορισμό και τεκμηρίωση συστημάτων λογισμικού. Σήμερα χιλιάδες εταιρείες χρησιμοποιούν τη γλώσσα UML στην ανάπτυξη των συστημάτων τους.

Φυσικά, η μοντελοποίηση των συστημάτων πληροφορικής δεν είναι κάτι νέο. Μοντέλα συστημάτων πληροφορικής κατασκευάζονται εδώ και δεκαετίες, με διαφορετικές όμως τεχνικές. Ο τρόπος με τον οποίο σχεδιάζουμε ένα σύστημα, είναι άρρηκτα συνδεδεμένος με την τεχνολογία που θα χρησιμοποιήσουμε για να το αναπτύξουμε. Έτσι, από τα διαγράμματα ροής προγράμματος περάσαμε, με την εμφάνιση του δομημένου προγραμματισμού, σε πιο προχωρημένες τεχνικές αναπαράστασης και σχεδίασης συστημάτων που περιλάμβαναν πιο σύνθετες διαγραμματικές τεχνικές για τον έλεγχο των πιο πολύπλοκων σύγχρονων συστημάτων. Σήμερα τα συστήματα που αναπτύσσονται είναι σχεδόν στο σύνολο τους αντικειμενοστραφή οπότε είναι σχεδόν αδύνατο να ασχοληθεί κανείς με την ανάπτυξη λογισμικού χωρίς να γνωρίζει να αναπτύσσει αντικειμενοστραφή συστήματα. Ως εκ τούτου είναι αναγκαία η γνώση της UML για την μοντελοποίηση αυτών των συστημάτων, μια και η UML αποτελεί την πρότυπη γλώσσα μοντελοποίησης αντικειμενοστραφών συστημάτων. Η UML αποτελεί πρότυπο του OMG (Object Management Group). Ο OMG είναι ένας διεθνής μη κερδοσκοπικός οργανισμός που παράγει και διαχειρίζεται πρότυπα για τη διαλειτουργικότητα των επιχειρηματικών εφαρμογών λογισμικού, μεταξύ των οποίων και η UML. Ο OMG έχει ως μέλη σχεδόν όλες τις μεγάλες εταιρίες παραγωγής λογισμικού και εκατοντάδες μικρότερες. Η προτυποποίηση της UML από τον OMG, η γενικότερη αποδοχή της από τη βιομηχανία λογισμικού, καθώς και το γεγονός πως έχει περάσει αρκετός χρόνος από το 1997 που υιοθετήθηκε το πρώτο πρότυπο της UML από τον OMG, προσδίδουν στη UML το σημαντικό χαρακτηριστικό της διάρκειας στο χρόνο, κάτι που είναι πολύ σημαντικό για τη βιομηχανία λογισμικού, που εξελίσσεται με ραγδαίους ρυθμούς.

## 2 UML

Η UML αποτελείται από ένα σύνολο διαγραμμάτων τα οποία είναι τα εξής:

- Διαγράμματα περιπτώσεων χρήσης.
- Διαγράμματα κλάσεων.
- Διαγράμματα καταστάσεων.
- Διαγράμματα δραστηριοτήτων.
- Διαγράμματα αλληλεπίδρασης.
- Διαγράμματα υλοποίησης.

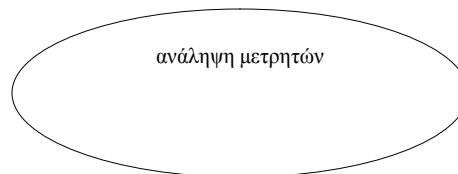
## 2.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ

Το βασικότερο θέμα στην κατασκευή ενός πληροφοριακού συστήματος είναι η κατασκευή του ορθού συστήματος, του συστήματος που ικανοποιεί τις απαιτήσεις των χρηστών του συστήματος και των άλλων συμμετεχόντων. Πολλά συστήματα πληροφορικής αποτυγχάνουν να ικανοποιήσουν αυτόν το φαινομενικά στοιχειώδη στόχο: δεν ικανοποιούν τις προδιαγραφές του χρήστη. Το μοντέλο περιπτώσεων χρήσης δίνει έμφαση στη λειτουργικότητα ενός συστήματος όπως αυτή είναι ορατή από τους εξωτερικούς χρήστες του. Μια περίπτωση χρήσης διαμοιράζει τη λειτουργικότητα ενός συστήματος σε συναλλαγές (περιπτώσεις χρήσης) που έχουν νόημα για τους χρήστες του συστήματος (χειριστές). Καταρχάς ένα σενάριο είναι μία ακολουθία βημάτων που περιγράφουν την αλληλεπίδραση μεταξύ ενός χρήστη και ενός συστήματος. Μία περίπτωση χρήσης είναι ένα σύνολο σεναρίων τα οποία συνδέονται από έναν κοινό για το χρήστη στόχο. Μία περίπτωση χρήσης έχει ένα σενάριο που όλα πάνε καλά και αρκετά εναλλακτικά σενάρια που μπορεί να περιλαμβάνουν πράγματα που πάνε στραβά ή επίσης εναλλακτικούς τρόπους που τα πράγματα πάνε καλά. Μια απλή μορφή για την αποτύπωση μιας περίπτωσης χρήσης περιλαμβάνει την περιγραφή του πρωταρχικού της σεναρίου ως μια ακολουθία από αριθμημένα βήματα και τις εναλλακτικές περιπτώσεις ως παραλλαγές αυτής της ακολουθίας.

Τα διαγράμματα περιπτώσεων χρήσης αποτελούν την διαγραμματική αναπαράσταση των περιπτώσεων χρήσης και αποτελούν μέρος της UML. Τα βασικά διαγραμματικά στοιχεία των διαγραμμάτων περιπτώσεων χρήσης είναι τα εξής:

### Περίπτωση χρήσης

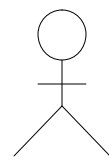
Αναπαριστά ένα στόχο για έναν εξωτερικό χειριστή (actor) τους συστήματος. Οι χειριστές ενός συστήματος μπορεί να είναι άνθρωποι άλλα ενδέχεται να είναι και εξωτερικά συστήματα, τα οποία είναι απαραίτητα για τη λειτουργία του υπό ανάπτυξη συστήματος. Το σύμβολο για μια περίπτωση χρήσης είναι η έλλειψη, μέσα στην οποία αναγράφεται το όνομα της περίπτωσης χρήσης. Μια περίπτωση χρήσης αποτελεί ένα στόχο υψηλού επιπέδου για τον χειριστή της, είναι κάτι που του αποδίδει κάποιο απτό αποτέλεσμα που έχει αξία για αυτόν. Μια περίπτωση χρήσης περιλαμβάνει συνήθως πολλά εναλλακτικά σενάρια, τα οποία ονομάζονται επεκτάσεις (extensions). Η ιδέα είναι ότι μία περίπτωση χρήσης έχει ένα κύριο σενάριο όπου όλα πάνε καλά (happy path) και αρκετές επεκτάσεις στις οποίες κάτι δεν πάει καλά ή προκύπτει μία εξαίρεση. Για παράδειγμα στην εικόνα 15 για ένα μηχάνημα ATM μιας τράπεζας η περίπτωση χρήσης «ανάληψη μετρητών» είναι κάτι που έχει αξία για έναν πελάτη.



Εικ. 15 Παράδειγμα περίπτωσης χρήσης

### χειριστής-χαρακτήρας

Ο χειριστής είναι ένας ρόλος που παίζει κάποιος χρήστης σε ότι αφορά το σύστημα. Όπως αναφέραμε και στην προηγούμενη παράγραφο, ο χειριστής ενός συστήματος μπορεί να είναι άνθρωπος ή υποσύστημα. Το σύμβολο που χρησιμοποιείται για τους χειριστές φαίνεται στην εικόνα 16, με ένα όνομα κάτω από τη φιγούρα που επισημαίνει το ρόλο που παίζει στα πλαίσια μίας περίπτωσης χρήσης. Το ίδιο σύμβολο μπορεί να χρησιμοποιηθεί και για τους χειριστές που είναι άνθρωποι αλλά και για τους χειριστές που είναι άλλα εξωτερικά συστήματα. Αν θέλουμε να δώσουμε έμφαση στην περίπτωση όπου ο χειριστής είναι ένα εξωτερικό σύστημα, μπορούμε να χρησιμοποιήσουμε ένα εναλλακτικό σύμβολο όπως φαίνεται στην εικόνα όπου το εξωτερικό σύστημα απεικονίζεται με το στερεότυπο «Actor». Θα πρέπει να επισημάνουμε το σημαντικό ρόλο που παίζουν οι χειριστές στη διαδικασία εύρεσης των προδιαγραφών: οι χειριστές (χρήστες και συστήματα) αποτελούν την πηγή από την οποία προκύπτουν οι προδιαγραφές του συστήματος. Στην ερώτηση «ποιες λειτουργίες πρέπει να προσφέρει το σύστημα» η απάντηση είναι «αυτές που χρειάζονται για την ικανοποίηση των στόχων των χειριστών του». Οι χειριστές διεκπεραιώνουν περιπτώσεις χρήσης. Ένας χειριστής μπορεί να διεκπεραιώσει πολλές περιπτώσεις χρήσης. Αντίστροφα μια περίπτωση χρήσης μπορεί να έχει πολλούς χειριστές που να την διεκπεραιώνουν. Επομένως, είναι κρίσιμο να βρούμε τους χειριστές του συστήματος (ανθρώπους, υποσυστήματα και εξωτερικά συστήματα) επειδή οι στόχοι τους θα μας υποδείξουν τις προδιαγραφές του υπό ανάπτυξη συστήματος. Αυτές ακριβώς οι προδιαγραφές αποτυπώνονται στο διάγραμμα περιπτώσεων χρήσης.



χειριστής

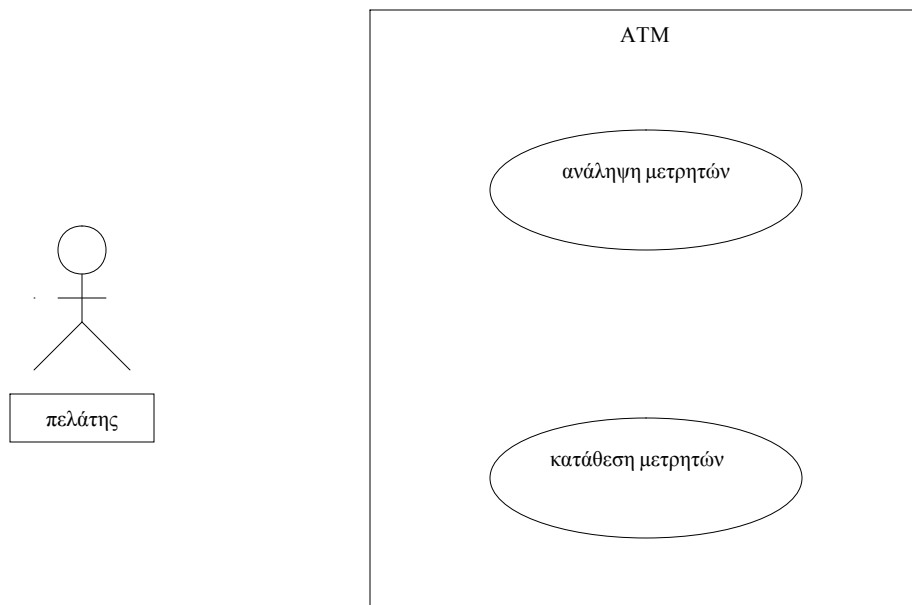


σύστημα χειριστής  
«Actor»

Εικ. 16 Παραδείγματα χειριστών συστήματος

### Σύστημα

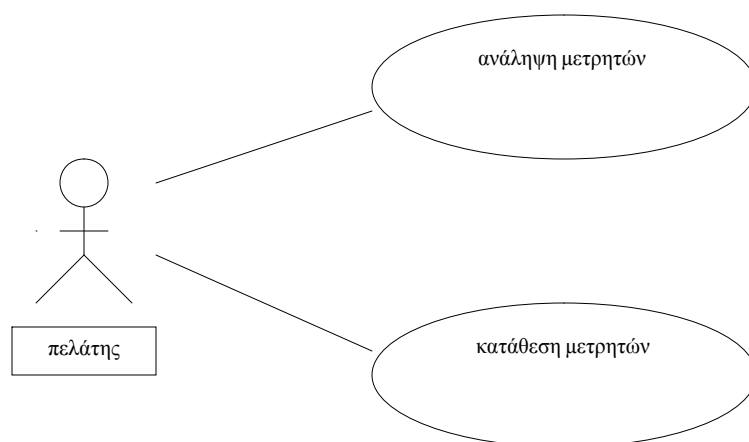
Για να διακρίνουμε τις προδιαγραφές που βρίσκονται μέσα στα πλαίσια του υπό ανάπτυξη συστήματος από τα πιθανά εξωτερικά συστήματα και τους χρήστες, περιλαμβάνουμε τις περιπτώσεις χρήσης σε ένα πλαίσιο με τίτλο το όνομα του συστήματος όπως φαίνεται και στην εικόνα 17. Μέσα στο πλαίσιο του συστήματος τοποθετούμε τις περιπτώσεις χρήσης και έξω και από αυτό τους χειριστές του συστήματος.



Εικ. 17 Παράδειγμα συστήματος και χειριστή

### Σχέση

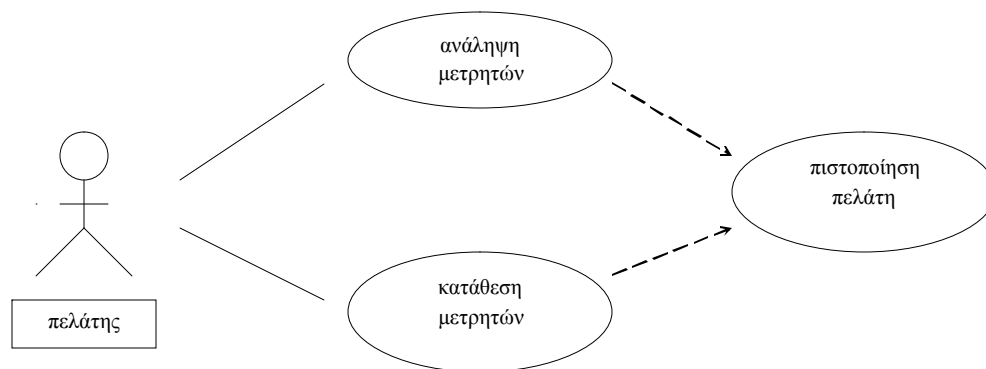
Υποδηλώνει τη σχέση ενός χειριστή με μια περίπτωση χρήσης. Ένας χειριστής, ενδέχεται να είναι ο βασικός (primary) χειριστής για μια περίπτωση χρήσης, αλλά να σχετίζονται επίσης με αυτήν και άλλοι χειριστές ή εξωτερικά συστήματα. Η σχέση συμβολίζεται με μια γραμμή που συνενώνει τον χειριστή με την περίπτωση χρήσης, όπως δείχνει και η εικόνα 18.



Εικ. 18 Παράδειγμα σχέσης συστήματος και χειριστή

### Συμπερίληψη

Η συμπερίληψη (include) είναι μια ειδική περίπτωση σχέσης, στην οποία σχετίζουμε δύο περιπτώσεις χρήσης. Η μια περίπτωση συμπεριλαμβάνει την άλλη. Η έννοια της συμπερίληψης είναι υποχρεωτική, δηλαδή πάντα η μια περίπτωση χρήσης θα συμπεριλαμβάνει την άλλη. Χρησιμοποιούμε τη συμπερίληψη όταν επαναλαμβάνονται τα ίδια σε δύο ή περισσότερες περιπτώσεις χρήσης και θέλετε να αποφύγετε την επανάληψη αυτή. Η φορά του βέλους στη συμπερίληψη είναι από την περίπτωση χρήσης που συμπεριλαμβάνει προς αυτήν που συμπεριλαμβάνεται όπως φαίνεται στην εικόνα 19. Στο παράδειγμα μας απαιτείται πιστοποίηση του πελάτη από το ΑΤΜ είτε έχουμε την περίπτωση ανάληψης μετρητών είτε την περίπτωση κατάθεσης μετρητών.

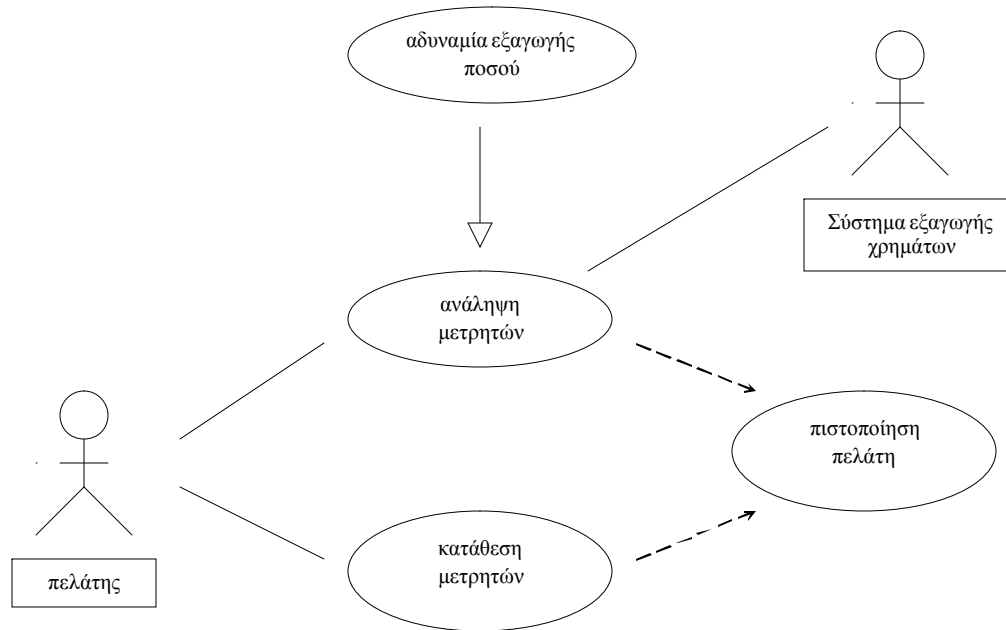


Εικ. 19 Παράδειγμα συμπερίληψης

### Επέκταση

Η επέκταση (extend) όπως και η συμπερίληψη, καθορίζει μια σχέση μεταξύ δύο περιπτώσεων χρήσης, στην οποία σχέση μια περίπτωση χρήσης επεκτείνεται προαιρετικά από μια άλλη, ανάλογα με τις επιλογές ή την κατάσταση κάποιου χειριστή, βασικού ή δευτερεύοντος. Η σχέση αυτή μπορεί να προκύψει σε συγκεκριμένα σημεία της λειτουργικότητας της βασική ροής της περίπτωσης χρήσης τα οποία ονομάζονται σημεία επέκτασης (extension points). Χρησιμοποιούμε επέκταση όταν περιγράφετε μια παραλλαγή μιας κανονικής συμπεριφοράς και θέλετε να χρησιμοποιήσετε μια πιο ελεγχόμενη μορφή, δηλώνοντας τα σημεία επέκτασης στη βασική περίπτωση χρήσης. Έστω, για παράδειγμα ότι για να γίνει η ανάληψη μετρητών στην εικόνα 20, το σύστημα θα πρέπει να επικοινωνήσει με το εξωτερικό σύστημα «σύστημα εξαγωγής χρημάτων», το οποίο είναι υπεύθυνο για την εξαγωγή χρημάτων. Το σύστημα αυτό είναι ένας από τους χειριστές της περίπτωσης χρήσης «ανάληψη μετρητών» και ενδέχεται να μην έχει τα κατάλληλα χαρτονομίσματα για την εξαγωγή του ποσού που ζήτησε ο πελάτης. Στην περίπτωση αυτή, η περίπτωση «ανάληψη μετρητών» επεκτείνεται από την περίπτωση χρήσης «αδυναμία εξαγωγής ποσού», η οποία είναι η ίδια περίπτωση χρήσης μέχρι του

σημείου όπου ο πελάτης ζητάει το ποσό και ο χειριστής «σύστημα εξαγωγής χρημάτων διαπιστώνει πως αδυνατεί να εξαγάγει αυτό το ποσό.



Εικ. 20 Παράδειγμα επέκτασης

## 2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

Τα διαγράμματα περιπτώσεων χρήσης που συζητήσαμε είναι διαγράμματα καταγραφής προδιαγραφών και είναι χρήσιμα για κάθε τύπο συστήματος. Τα αντικειμενοστραφή συστήματα όμως λειτουργούν ως μια συλλογή συνεργαζόμενων αντικειμένων. Τα αντικείμενα αποτελούν στιγμιότυπα κλάσεων, η κλάση είναι ένας τύπος από τον οποίο δημιουργούνται κατά τη διάρκεια της εκτέλεσης του προγράμματος αντικείμενα που ανήκουν στον τύπο αυτό. Είναι λοιπόν χρήσιμο να καταγράψουμε τις κλάσεις που ανήκουν σε ένα σύστημα και τις μεταξύ τους συσχετίσεις. Αυτό ακριβώς επιτυγχάνουμε με ένα διάγραμμα κλάσεων.

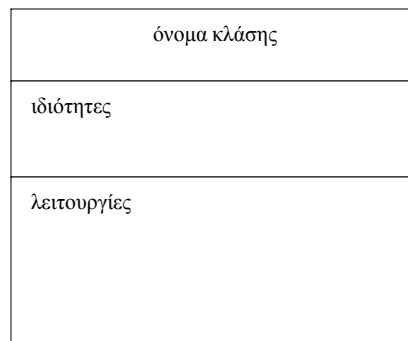
Πριν συνεχίσουμε με τις λεπτομέρειες των διαγραμμάτων κλάσεων, θα ήταν χρήσιμο να ορίσουμε την οπτική γωνία που χρησιμοποιούμε για τη σχεδίαση ενός διαγράμματος κλάσεων, επειδή υπάρχουν τουλάχιστον δύο οπτικές γωνίες. Κατά τη διαδικασία της ανάλυσης προδιαγραφών, οι κατασκευαστές αρχίζουν να αποκτούν γνώση για το πεδίο του προβλήματος του συστήματος. Αυτή η αρχική κατανόηση των εννοιών του πεδίου του προβλήματος είναι φυσικό να καταγραφεί σε ένα διάγραμμα κλάσεων, το οποίο

ονομάζεται μοντέλο του πεδίου προβλήματος (problem domain model). Στο μοντέλο αυτό καταγράφονται ως κλάσεις οι έννοιες του πεδίου του προβλήματος χωρίς να υπάρχει δέσμευση πως αυτές οι ίδιες κλάσεις θα υπάρχουν και στο λογισμικό που θα κατασκευαστεί (έστω και αν , κατά πάσα πιθανότητα, οι περισσότερες από αυτές θα υπάρχουν). Επίσης για κάθε κλάση του μοντέλου αυτού ενδεχομένως να αναγραφούν κάποιες υψηλού επιπέδου υποχρεώσεις (σαν λειτουργίες τους). Και εδώ δεν υπάρχει η υποχρέωση οι λειτουργίες αυτές να εμφανισθούν στο διάγραμμα κλάσεων του λογισμικού ως μέθοδοι των αντιστοίχων κλάσεων. Τέλος, μπορεί να εμφανίσουμε στο μοντέλο του χώρου ανάπτυξης και τις συσχετίσεις μεταξύ των κλάσεων. Κατά τη φάση της επεξεργασίας, το μοντέλο του πεδίου του προβλήματος θα αποτελέσει πηγή έμπνευσης για το επίπεδο του πεδίου προβλήματος του λογισμικού, με τις περισσότερες κλάσεις να υπάρχουν και εκεί. Φυσικά οι λειτουργίες και οι συσχετίσεις μπορεί να έχουν αλλάξει και να έχουν βελτιωθεί. Οι λειτουργίες που αναγράφονται στο μοντέλο του πεδίου προβλήματος αποτελούν υψηλού επιπέδου υποχρεώσεις των οντοτήτων του χώρου ανάπτυξης. Οι λειτουργίες των κλάσεων του λογισμικού αποτελούν μεθόδους των αντικειμένων αυτών των κλάσεων, οι οποίες καλούνται από άλλα αντικείμενα, άλλων συνεργαζόμενων κλάσεων, στο πλαίσιο της υλοποίησης ενός σεναρίου μιας περίπτωσης χρήσης. Η εύρεση των μεθόδων γίνεται με την καθοδήγηση των διαγραμμάτων αλληλεπίδρασης, τα οποία θα περιγράψουμε στην συνέχεια. Η εύρεση των υψηλού επιπέδου υποχρεώσεων γίνεται με την ανάλυση του συστήματος.

Τα διαγραμματικά στοιχεία που συναντάμε σε ένα διάγραμμα κλάσεων είναι τα εξής:

#### ΚΛΑΣΗ, ΙΔΙΟΤΗΤΕΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ

Οι κλάσεις αποτελούν τη βάση της κατασκευής οποιουδήποτε αντικειμενοστραφούς συστήματος. Ενσωματώνουν τα δεδομένα καθώς και τις λειτουργίες που επενεργούν στα δεδομένα αυτά. Συμβολίζονται με ένα παραλληλόγραμμο που έχει τρία διαμερίσματα. Στο πάνω διαμέρισμα αναγράφεται το όνομα της κλάσης, στο μεσαίο διαμέρισμα απεικονίζονται οι ιδιότητες και στο κάτω διαμέρισμα οι λειτουργίες. Ανάλογα με τη οπτική γωνία σχεδίασης που χρησιμοποιούμε, η κλάση μπορεί να είναι μία κλάση λογισμικού ή μία έννοια του πεδίου του προβλήματος. Αντίστοιχα, οι λειτουργίες μπορεί να είναι μέθοδοι ή υποχρεώσεις υψηλού επιπέδου. Στην εικόνα 21 βλέπουμε μία κλάση.



Εικ.21 Παράδειγμα σχεδίασης κλάσης

Το συνηθισμένο σε μια κλάση είναι ότι οι ιδιότητες είναι ιδιωτικές (private) και οι λειτουργίες δημόσιες (public). Ιδιωτικά είναι τα χαρακτηριστικά μιας κλάσης, ιδιότητες ή λειτουργίες, που δεν είναι προσπελάσιμα από άλλες κλάσεις, ενώ δημόσια αυτά που είναι προσπελάσιμα από άλλες κλάσεις. Το ότι τα δεδομένα είναι ιδιωτικά σημαίνει ότι αντικείμενα άλλων κλάσεων δεν έχουν τη δυνατότητα να προσπελάσουν απευθείας τις ιδιότητες μιας κλάσης, αλλά χρησιμοποιούν για αυτό τις μεθόδους της κλάσης. Η τακτική αυτή ονομάζεται αρχή απόκρυψης των δεδομένων (information hiding principle). Το σύμβολο «-» μπροστά από τις ιδιότητες και τα χαρακτηριστικά χρησιμοποιείται για να δηλώσει ιδιωτική πρόσβαση, ενώ το «+» σημαίνει δημόσια πρόσβαση. Το σύμβολο «#», που συμβολίζει την προστατευμένη πρόσβαση, έχει τη σημασία πως η ιδιότητα ή λειτουργία είναι προσπελάσιμη από την κλάση και τις τυχόν υποκλάσεις της. Η πρόσβαση σε επίπεδο πακέτου έχει την επισήμανση «~» και τη σημασία πως η ιδιότητα ή λειτουργία είναι προσπελάσιμη από την κλάση στην οποία δηλώνεται και τις άλλες κλάσεις που βρίσκονται στο ίδιο πακέτο με αυτήν.

Το συντακτικό για τη δήλωση ιδιοτήτων στη UML είναι:

*Προσδιοριστής πρόσβασης όνομα ιδιότητας:*

*τύπος[πολλαπλότητα διάταξης]=αρχική τιμή{συμβολοσειρά ιδιοτήτων}*

- Ο τύπος είναι κάποιος τύπος δεδομένων
- Η πολλαπλότητα μπορεί να είναι κάποιο διάστημα τιμών με κάτω όριο και πάνω όριο. Επίσης, μπορεί να είναι κάποια συγκεκριμένη αριθμητική τιμή.
- Η διάταξη μπορεί να μην υπάρχει ή να είναι η λέξη unordered, οπότε και θα θεωρείται πως δεν υπάρχει διάταξη, ή να είναι η λέξη ordered, οπότε υπάρχει διάταξη.
- Η αρχική τιμή είναι η τιμή που αποδίδεται στη συγκεκριμένη ιδιότητα μετά την δημιουργία ενός αντικειμένου της κλάσης. Η τιμή θα πρέπει να ανήκει στον τύπο της ιδιότητας.
- Τέλος, η συμβολοσειρά ιδιοτήτων περιέχει ιδιότητες για τη συγκεκριμένη δήλωση.

Το συντακτικό για τη δήλωση λειτουργιών στην UML είναι:

*Προσδιοριστής πρόσβασης όνομα (λίστα-παραμέτρων):*

*Τύπος επιστροφής {συμβολοσειρά ιδιοτήτων}*

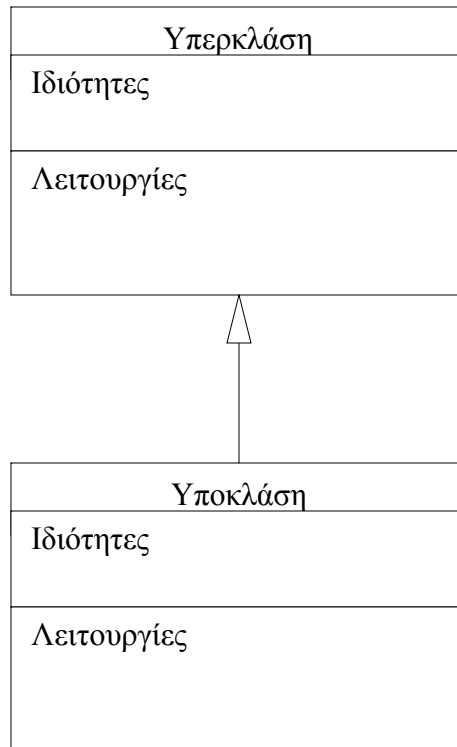
- Ο προσδιοριστής πρόσβασης είναι +, -, #, ή ~ όπως και για τις ιδιότητες.
- Το όνομα είναι το όνομα της λειτουργίας.
- Η λίστα παραμέτρων είναι μία λίστα που προσδιορίζει τις παραμέτρους της λειτουργίας, χωρισμένες με κόμματα.
- Ο τύπος επιστροφής προσδιορίζει τον τύπο των δεδομένων που επιστρέφει η συνάρτηση.
- Με την συμβολοσειρά ιδιοτήτων προσδιορίζουμε κάποιες ιδιότητες για τη λειτουργία.



## ΣΥΣΧΕΤΙΣΕΙΣ ΚΛΑΣΕΩΝ

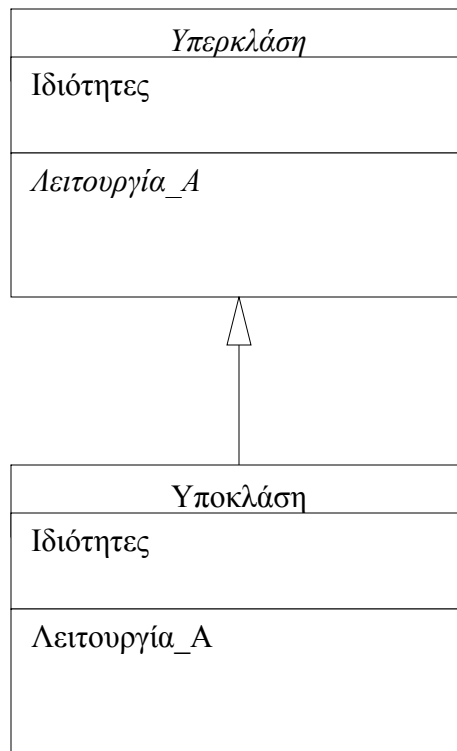
Μια συσχέτιση (association) μεταξύ δύο κλάσεων απεικονίζει μια στατική σχέση μεταξύ των δύο κλάσεων. Η ανάγκη μιας συσχέτισης προκύπτει από τη διαπίστωση πως για τη λειτουργία μιας κλάσης απαιτείται η συνεργασία της με μία ή περισσότερες άλλες κλάσεις. Αν αυτή η συνεργασία απαιτείται να είναι σε μόνιμη βάση, τότε χρησιμοποιούμε συσχέτιση, αν είναι παροδική τότε χρησιμοποιούμε εξάρτηση. Για την αναπαράσταση μια συσχέτισης χρησιμοποιούμε μία γραμμή μεταξύ των δύο κλάσεων. Κάθε συσχέτιση έχει δύο πέρατα συσχέτισης, κάθε πέρας είναι συνδεδεμένο με μία από τις τάξεις της συσχέτισης. Ένα πέρας μπορεί να ονομαστεί με μία ετικέτα. Αυτή η ετικέτα ονομάζεται όνομα ρόλου (role name) και τα πέρατα των συσχέτισεων συχνά ονομάζονται ρόλοι. Το πέρας μιας συσχέτισης έχει επίσης πολλαπλότητα (multiplicity), η οποία είναι μία ένδειξη για το πόσα αντικείμενα μπορούν να συμμετέχουν σε αυτή τη σχέση. Οι γραμμές συσχέτισης έχουν βέλη τα οποία δείχνουν τη δυνατότητα πλοήγησης (navigability). Αν μια δυνατότητα πλοήγησης υπάρχει μόνο στην μια κατεύθυνση, ονομάζουμε τη συσχέτιση μονόδρομη (unidirectional association), ενώ αν υπάρχει δυνατότητα πλοήγησης και προς τις δύο κατευθύνσεις τότε η συσχέτιση ονομάζεται αμφίδρομη. Υπάρχουν αρκετοί τρόποι ονομασίας μιας συσχέτισης. Οι περισσότεροι σχεδιαστές μοντέλων αντικειμένων προτιμούν να χρησιμοποιούν ουσιαστικά για να ονομάσουν το ρόλο στο ένα ή στο άλλο πέρας της συσχέτισης, μια και αυτό παρέχει καλύτερη αντιστοιχία με τις υποχρεώσεις και τις λειτουργίες, άλλα υπάρχει και ο παραδοσιακός τρόπος όπου μια συσχέτιση ονομάζεται χρησιμοποιώντας μια ρηματική φράση έτσι ώστε η σχέση να μπορεί να χρησιμοποιηθεί σε μια πρόταση.

Η γενίκευση είναι μία ειδική μορφή συσχέτισης, κατά την οποία μια γενική κλάση αποτελεί τη βάση για τη δήλωση μίας ή περισσότερων ειδικότερων κλάσεων. Η γενική κλάση ονομάζεται υπερκλάση και ειδικές κλάσεις ονομάζονται υποκλάσεις. Η γενίκευση στις περισσότερες γλώσσες προγραμματισμού υλοποιείται με το μηχανισμό της κληρονομικότητας ή της επέκτασης. Η ιδέα είναι πως η γενική κλάση παρέχει λειτουργίες, ιδιότητες και συσχετίσεις που είναι χρήσιμες σε όλες τις υποκλάσεις. Από την άλλη μεριά, οι υποκλάσεις επεκτείνουν τη λειτουργικότητα της υπερκλάσης και παρέχουν επιπλέον λειτουργίες όπου αυτό είναι απαραίτητο ή εξειδικεύουν τη συμπεριφορά τους. Στην εικόνα 22 φαίνεται το σύμβολο της γενίκευσης που είναι ένα βέλος που δείχνει από την ειδική στη γενική κλάση.



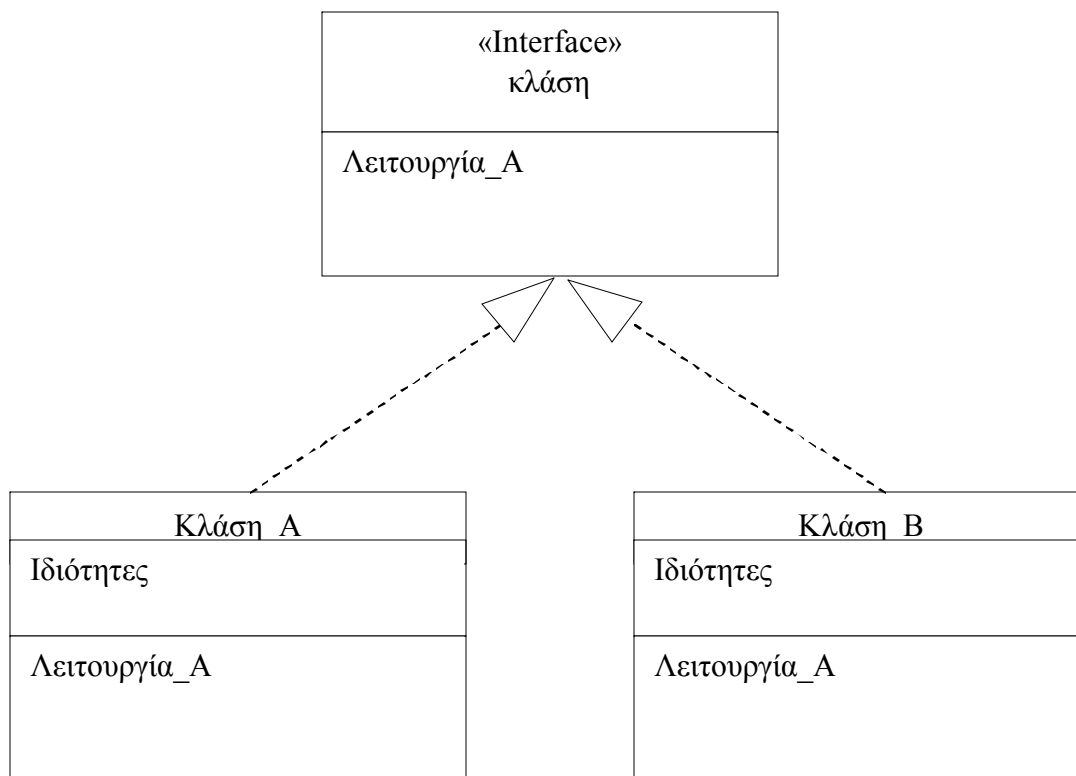
Εικ. 22 Παράδειγμα γενίκευσης

Κάποιες κλάσεις σε ένα σύστημα δηλώνονται ως αφαιρετικές (abstract). Αυτές οι κλάσεις παρέχουν κώδικα για κάποιες λειτουργίες αλλά αφήνουν άλλες λειτουργίες τους χωρίς κώδικα. Η ιδέα είναι πως κάποιες υποκλάσεις αυτών των κλάσεων θα προσδιορίσουν τις χωρίς υλοποίηση λειτουργίες, παρέχοντας την κατάλληλη για αυτές υλοποίηση. Αυτές οι υποκλάσεις ονομάζονται συγκεκριμένες (concrete). Οι συγκεκριμένες υποκλάσεις κληρονομούν τα συγκεκριμένα και αφαιρετικά στοιχεία των υπερκλάσεων τους. Ταυτόχρονα συγκεκριμενοποιούν παρέχοντας την κατάλληλη υλοποίηση τα αφαιρετικά στοιχεία. Για να επισημάνετε μία κλάση ως αφαιρετική στη UML χρησιμοποιείτε το όνομα της κλάσης με πλάγια γραφή ή επίσης μπορείτε να χρησιμοποιήσετε τον προσδιορισμό {abstract}. Επίσης με πλάγια γραφή επισημαίνονται και τα αφαιρετικά στοιχεία (οι αφαιρετικές λειτουργίες) (εικ. 23).



Εικ. 23 Παράδειγμα αφαιρετικής κλάσης

Μία διασύνδεση (interface) είναι ένας τύπος που παρέχει ένα σύνολο λειτουργιών οι οποίες είναι στο σύνολο τους αφαιρετικές. Μία διασύνδεση δεν έχει δεδομένα, κατάσταση, άλλα ούτε συσχετίσεις με πλοϊμότητα από τη διασύνδεση προς το άλλο άκρο της συσχέτισης. Αυτό αποτελεί συνέπεια της έλλειψης δεδομένων. Κάποια κλάση που πραγματώνει (realizes) τη διασύνδεση έχει την υποχρέωση να υλοποιήσει όλες τις λειτουργίες της διασύνδεσης. Το κέρδος για την κλάση είναι ότι τα αντικείμενα της θα έχουν τον τύπο της διασύνδεσης και επομένως θα μπορούν να χρησιμοποιηθούν όπου αναμένονται αντικείμενα του τύπου της διασύνδεσης. Κλάσεις που δεν έχουν καμία σχέση μεταξύ τους μπορούν να υλοποιήσουν μία διασύνδεση και να ωφεληθούν από λειτουργίες που ενεργούν σε αντικείμενα αυτής της διασύνδεσης. Μία διασύνδεση στη UML σχεδιάζεται με το σύμβολο της κλάσης και με τη λέξη κλειδί «Interface» πάνω από το όνομα της κλάσης, για να υποδηλώνεται ότι πρόκειται για διασύνδεση. Φυσικά, αφού μια διασύνδεση δεν έχει ιδιότητες, το διαμέρισμα των ιδιοτήτων μπορεί να μην υπάρχει ή, εφόσον υπάρχει, να είναι κενό. Μια κλάση που υλοποιεί τη διασύνδεση συσχετίζεται με αυτή με το σύμβολο της πραγμάτωσης (realization), που είναι ίδιο με το σύμβολο της γενίκευσης, άλλα με γραμμή που είναι διακεκομμένη. Στην εικόνα 24 μπορούμε να δούμε μια διασύνδεση που υλοποιείται με το συμβολισμό της κλάσης και δύο κλάσεις που πραγματώνουν τη διασύνδεση.

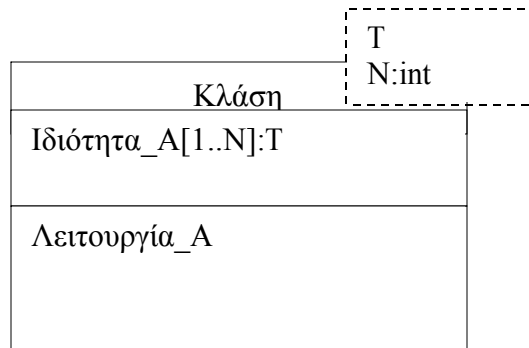


Εικ. 24 Παράδειγμα διασύνδεσης

Οι αφηρημένες τάξεις και οι διασυνδέσεις είναι όμοιες, αλλά υπάρχει μία διαφορά. Και οι δύο επιτρέπουν τη δήλωση μιας διασύνδεσης και αναβάλλουν την υλοποίηση της για αργότερα. Παρόλα αυτά οι αφηρημένες κλάσεις σας επιτρέπουν να προσθέσετε υλοποίηση για κάποιες από τις μεθόδους. Μια διασύνδεση σας εξαναγκάζει να αναβάλετε την υλοποίηση όλων των λειτουργιών.

Πολύ συχνά θέλουμε να ορίσουμε μία κλάση στην οποία κάποιος τύπος των ιδιοτήτων της κλάσης είναι παράμετρος. Ένα κλασικό παράδειγμα είναι η περίπτωση των διαφόρων κλάσεων που ορίζουν συλλογές στοιχείων. Για παράδειγμα, όταν ορίζουμε μία ουρά προτεραιότητας θα θέλαμε να μπορούμε να καθορίσουμε την ουρά χωρίς αναφορά στο συγκεκριμένο τύπο των στοιχείων που θα εισαχθούν σε αυτήν. Το πλεονέκτημα είναι πως με αυτόν τον τρόπο μπορούμε να υλοποιήσουμε μία κλάση γενικού σκοπού και να τη χρησιμοποιήσουμε με διάφορους τύπους. Μια παραμετρική κλάση στη UML είναι ένα πρότυπο (template) με ένα ή περισσότερα μη δεσμευμένα στοιχεία. Τα στοιχεία αυτά τυπικά αναπαριστούν έναν τύπο μίας ή περισσότερων ιδιοτήτων της κλάσης και μπορούν

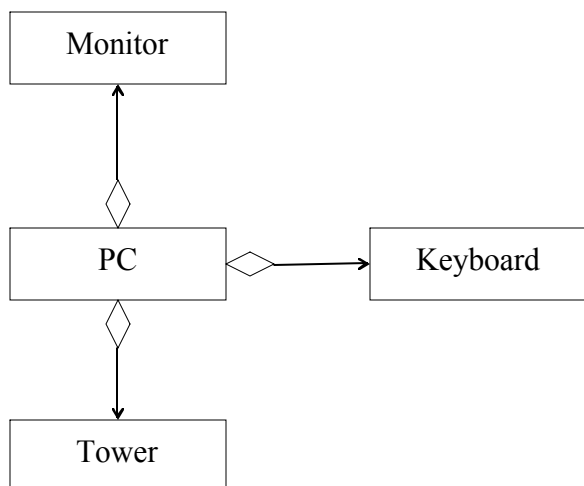
επίσης να εμφανιστούν στις λειτουργίες της κλάσης ως παράμετροι. Ο συμβολισμός της UML για τις παραμετρικές κλάσεις είναι το σύμβολο της κλάσης με ένα διακεκομμένο τετράγωνο στην πάνω δεξιά γωνία, όπου εισάγονται τα ονόματα, οι τύποι, και οι εξορισμού τιμές για τις παραμέτρους της κλάσης. Αν ο τύπος της παραμέτρου παραλειφθεί, τότε θεωρείται πως θα είναι κάποιος τύπος δεδομένων του συστήματος. Μια παραμετρική κλάση φαίνεται στην εικόνα 25.



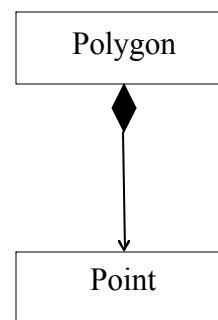
Εικ. 25 Παράδειγμα παραμετρικής κλάσης

Παραμετρικές κλάσεις δεν μπορούν να χρησιμοποιηθούν για τη δημιουργία αντικειμένων. Για να προκύψουν συγκεκριμένες κλάσεις, από τις οποίες να μπορούν να δημιουργηθούν αντικείμενα, θα πρέπει να δεσμευθούν οι παράμετροι σε συγκεκριμένους τύπους. Αυτές οι συγκεκριμένες κλάσεις ονομάζονται δεσμευμένα στοιχεία (Bound Elements).

Η συναρμολόγηση (aggregation) και η σύνθεση (composition) είναι δύο ειδικές περιπτώσεις συσχετίσεων. Και ο δύο υποδηλώνουν τη συσχέτιση μιας κλάσης με κάποια άλλη κλάση που αποτελεί μέρος της. Είναι δηλαδή και οι δύο συσχετίσεις όλου και μερών. Η συναρμολόγηση έχει μόνο έναν επιπλέον περιορισμό σε σχέση με μία συνηθισμένη συσχέτιση, δεν επιτρέπεται η κυκλική συσχέτιση του μέρους με το όλο, αλλά μόνο μία συσχέτιση από το όλο προς το μέρος. Αντιθέτως, η σύνθεση είναι μία ισχυρή μορφή συσχέτισης με δύο επιπλέον σημασιολογικά στοιχεία που τη διαφοροποιούν από τις συνηθισμένες συσχετίσεις, το πρώτο είναι πως το όλο περιέχει αποκλειστικά τα μέρη του, δεν μπορεί δηλαδή κάποιο άλλο όλο να περιέχει το ίδιο αντικείμενο. Το δεύτερο είναι πως δημιουργούνται και καταστρέφονται ταυτόχρονα με το όλο. Μια συναρμολόγηση συμβολίζεται με μία συσχέτιση από το όλο προς το μέρος, στην οποία τοποθετείται ένας άσπρος ρόμβος στην πλευρά του όλου. Η σύνθεση συμβολίζεται με μία συσχέτιση από το όλο προς το μέρος στην οποία τοποθετείται ένας μαύρος ρόμβος στην πλευρά του όλου. Σαν παράδειγμα συναρμολόγησης η εικόνα 26 απεικονίζει έναν προσωπικό υπολογιστή που αποτελείται από μία οθόνη, ένα πληκτρολόγιο και μία κεντρική μονάδα. Σαν παράδειγμα σύνθεσης η εικόνα 27 που αφορά ένα σχεδιαστικό πρόγραμμα αναπαριστά ένα πολύγωνο το οποίο συσχετίζεται μέσω σύνθεσης με τα σημεία που αποτελούν τις κορυφές του.

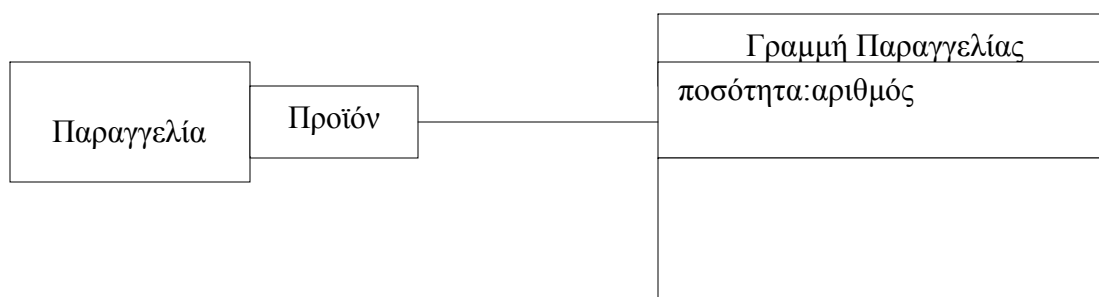


Εικ. 26 Παράδειγμα συναρμολόγησης



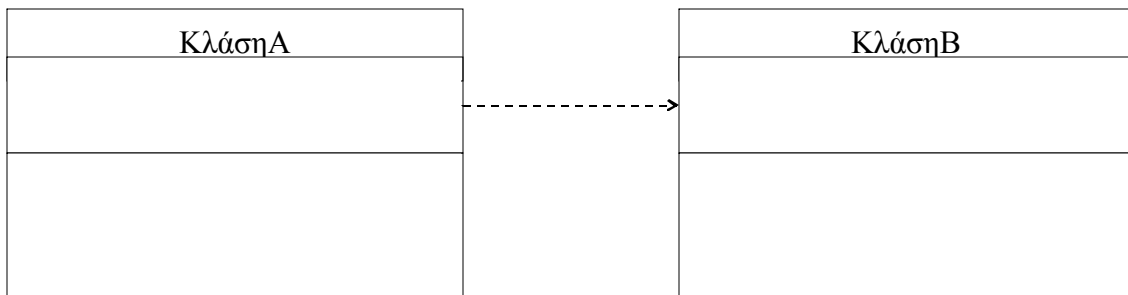
Εικ. 27 Παράδειγμα σύνθεσης

Πολλές φορές όταν υλοποιούμε συσχετίσεις μιας κλάσης με πολλαπλότητα μεγαλύτερη του 1, θέλουμε να περιορίσουμε τη συσχέτιση έτσι ώστε με τη χρήση κάποιας ιδιότητας να ταυτοποιούμε τα αντικείμενα που συμμετέχουν στη συσχέτιση στο άλλο άκρο της συσχέτισης. Αυτό επιτυγχάνεται με την προσδιορισμένη συσχέτιση. Μια προσδιορισμένη συσχέτιση της UML είναι ισοδύναμη με μία έννοια που συναντάμε στον προγραμματισμό και είναι γνωστή με διάφορους όρους όπως συσχετιστικοί πίνακες (associative arrays), χάρτες (maps) και λεξικά (dictionaries). Μια προσδιορισμένη συσχέτιση συμβολίζεται με ένα παραλληλόγραμμο στην κλάση που αποτελεί την πηγή της συσχέτισης μέσα στην οποία αναγράφεται το προσδιοριστικό. Το προσδιοριστικό χρησιμοποιείται για τον προσδιορισμό των αντικειμένων που συμμετέχουν στη συσχέτιση στο άλλο άκρο της συσχέτισης, που είναι ο στόχος. Η εικόνα 28 αναπαριστά τη συσχέτιση μεταξύ των κλάσεων παραγγελία και γραμμή παραγγελίας, που χρησιμοποιεί ένα προσδιοριστικό (qualifier). Το προσδιοριστικό λέει ότι, σε σχέση με μία παραγγελία, μπορεί να υπάρχει μόνο μία γραμμή παραγγελίας για κάθε παρουσία ενός προϊόντος. Εννοιολογικά αυτό το παράδειγμα σημαίνει ότι δεν μπορείτε να έχετε δύο γραμμές παραγγελίας μέσα στην ίδια παραγγελία για το ίδιο προϊόν.



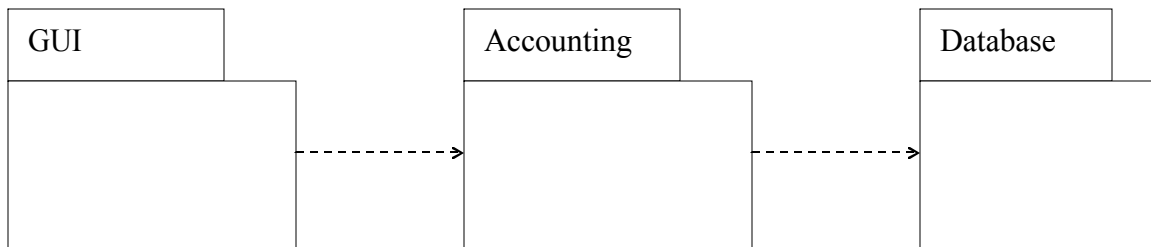
Εικ. 28 Παράδειγμα προσδιορισμένης συσχέτισης

Οι εξαρτήσεις (dependencies) συμβολίζονται με διακεκομμένα βέλη και συνδέουν ένα διαγραμματικό στοιχείο από το οποίο ξεκινούν που ονομάζεται στοιχείο προέλευση με ένα άλλο διαγραμματικό στοιχείο στο οποίο καταλήγουν που ονομάζεται στοιχείο προορισμός. Η ιδέα είναι πως αν μεταβληθεί το στοιχείο προέλευση θα πρέπει πιθανώς να αλλάξει και το στοιχείο προορισμός. Στόχος της ανάπτυξης είναι η ελαχιστοποίηση των εξαρτήσεων. Στην εικόνα 29 φαίνεται η αναπαράσταση της εξάρτησης.



Εικ. 29 Παράδειγμα εξάρτησης

Τα μοντέλα ενός μεγάλου συστήματος είναι χρήσιμο να οργανώνονται σε πακέτα. Τα πακέτα προσφέρουν τη δυνατότητα ομαδοποίησης συναφών, υπό κάποια έννοια μοντέλων μέσω της τοποθέτησης αυτών των μοντέλων στο ίδιο πακέτο. Επίσης παρέχουν τη δυνατότητα ελέγχου των εξαρτήσεων αφού μπορούμε να περιορίσουμε την πρόσβαση στα στοιχεία του πακέτου εκθέτοντας μόνο ένα υποσύνολο τους. Για παράδειγμα, τα στοιχεία με πρόσβαση πακέτου δεν είναι ορατά έξω από το πακέτο τους. Τέλος, τα πακέτα περιέχουν ένα χώρο ονομάτων, αφού είναι δυνατόν να έχουμε σε διαφορετικά πακέτα στοιχεία με το ίδιο όνομα χωρίς να προκαλείται σύγχυση. Ένα πακέτο συμβολίζεται με ένα φάκελο όπως φαίνεται και στην εικόνα 30. Το όνομα του πακέτου αναγράφεται είτε μέσα στο πακέτο, είτε στο μικρό τετράγωνο στην πάνω αριστερή γωνία. Μπορούμε να εμφανίσουμε τα περιεχόμενα του πακέτου μέσα στο πακέτο, είτε να τα παραλείψουμε. Είναι χρησιμότερο να έχουμε ένα διάγραμμα στο οποίο εμφανίζονται μόνο τα πακέτα και οι εξαρτήσεις τους και τα περιεχόμενα των πακέτων να εμφανίζονται σε διαφορετικά διαγράμματα. Τα πακέτα ενδέχεται να περιέχουν άλλα πακέτα κ.ο.κ. Οι εξαρτήσεις μεταξύ των πακέτων είναι πολύ σημαντικές για τον έλεγχο των αλλαγών σε ένα έργο. Στην εικόνα 30 βλέπουμε ότι η γραφική διασύνδεση χρήστη (GUI) εξαρτάται, δηλαδή χρησιμοποιεί, στοιχεία από το πακέτο της λογιστικής (Accounting). Επίσης το πακέτο της λογιστικής εξαρτάται από στοιχεία στο πακέτο της βάσης δεδομένων (Database). Αυτή η καταγραφή των εξαρτήσεων είναι σημαντική διότι, αν αλλάξει κάτι στο πακέτο της λογιστικής δεν χρειάζεται να ανησυχήσουμε για αλλαγές στο πακέτο της βάσης δεδομένων παρά μόνο για αλλαγές στο πακέτο της γραφικής διασύνδεσης χρήστη, επειδή το πακέτο αυτό εξαρτάται από το πακέτο της λογιστικής. Στις γλώσσες προγραμματισμού ένα πακέτο της UML θα αντιστοιχιστεί σε κάποια αντίστοιχη έννοια της γλώσσας προγραμματισμού, για παράδειγμα σε ένα package της Java.



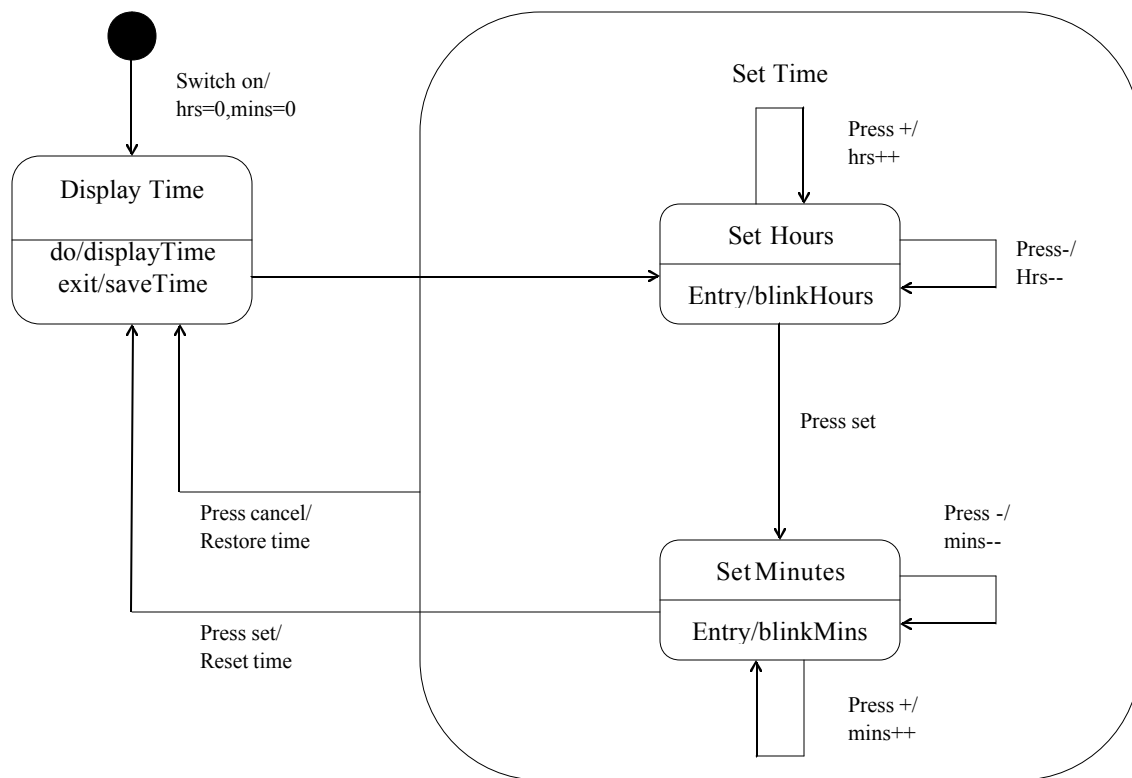
Εικ. 30 Παράδειγμα πακέτων και των μεταξύ τους εξαρτήσεων

### 2.3 ΔΙΑΓΡΑΜΜΑΤΑ ΚΑΤΑΣΤΑΣΗΣ

Ένα διάγραμμα κατάστασης (state diagram) απεικονίζει τη δυναμική συμπεριφορά των αντικειμένων μιας κλάσης και τον τρόπο που μεταβάλλεται η κατάσταση τους ως αντίδραση σε συμβάντα. Μπορείτε να χρησιμοποιήσετε τα διαγράμματα κατάστασης για να μοντελοποιήσετε τη συμπεριφορά και άλλων στοιχείων της UML, όπως π.χ. των χειριστών ενός συστήματος, αλλά τυπικά χρησιμοποιούνται για τα στιγμιότυπα μιας κλάσης. Επιπλέον, συνήθως δεν σχεδιάζουμε διαγράμματα κατάστασης για όλες τις κλάσεις σε ένα σύστημα, παρά μόνο για εκείνες που παρουσιάζουν μία έντονα δυναμική συμπεριφορά. Το διάγραμμα κατάστασης μας βοηθάει να αντιληφθούμε αυτήν τη συμπεριφορά. Ως παράδειγμα συμπεριφοράς για τα βασικά σύμβολα των διαγραμμάτων κατάστασης θα χρησιμοποιήσουμε το διάγραμμα της εικόνας 31, στο οποίο απεικονίζεται η δυναμική συμπεριφορά για ένα ρολόι. Το ρολόι βρίσκεται σε δύο δυνατές καταστάσεις, την κατάσταση εμφάνισης της ώρας («Display Time») και την κατάσταση ρύθμισης της ώρας («Set Time»). Η κατάσταση ρύθμισης της ώρας αναλύεται περαιτέρω σε δύο υποκαταστάσεις, αυτή της ρύθμισης των ωρών («Set Hours») και αυτή της ρύθμισης των λεπτών («Set Minutes»). Οι υποκαταστάσεις περιέχονται στη σύνθετη κατάσταση «Set Time». Κάθε κατάσταση συνδέεται με τις άλλες καταστάσεις με βέλη που ονομάζονται μεταβάσεις. Μία μετάβαση λαμβάνει χώρα όταν προκύπτει ένα συμβάν. Στο διάγραμμα εμφανίζεται μία ειδική αρχική ψευδοκατάσταση (ο μαύρος κύκλος). Από την αρχική κατάσταση περνάμε στην κατάσταση εμφάνισης της ώρας αφού προκύψει το συμβάν Switch on. Μετά το όνομα του συμβάντος μπορούμε να βάλουμε μία κάθετο (/) και να αναγράψουμε τυχόν ενέργειες που γίνονται κατά τη διαδικασία της μετάβασης από τη μία κατάσταση στην άλλη και πριν από την είσοδο στην επόμενη κατάσταση. Για παράδειγμα, κατά την μετάβαση από την αρχική κατάσταση στη κατάσταση «Display Time» οι ώρες και τα λεπτά παίρνουν αρχική τιμή μηδέν. Η κατάσταση συμβολίζεται με ένα παραλληλόγραμμο με στρογγυλεμένες γωνίες. Στο πάνω μέρος της κατάστασης εμφανίζεται το όνομα της. Προαιρετικά μπορούν να εμφανισθούν στο εσωτερικό της κατάστασης δραστηριότητες που λαμβάνουν χώρα κατά την είσοδο στην κατάσταση (με την ένδειξη entry), κατά την έξοδο από την κατάσταση (με την ένδειξη exit), και κατά τη διάρκεια της κατάστασης (με την ένδειξη do). Για παράδειγμα, όσο είμαστε στην κατάσταση της εμφάνισης της ώρας λαμβάνει χώρα η δραστηριότητα display time. Όταν

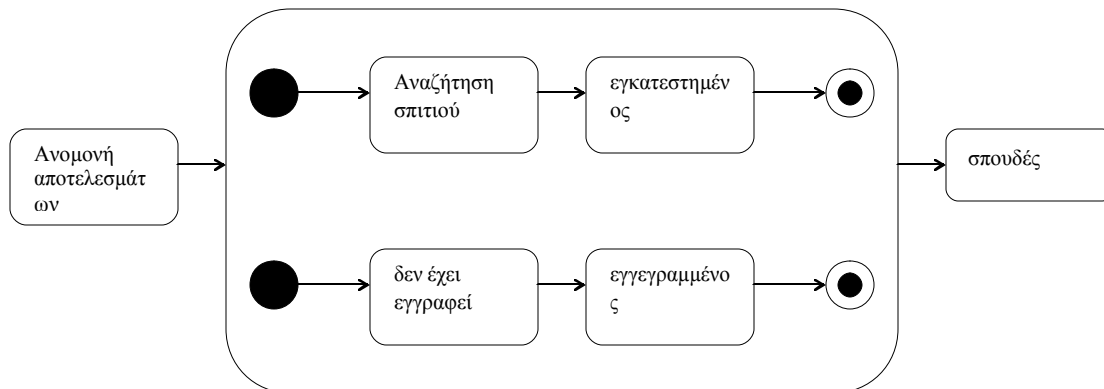


βγαίνουμε από την κατάσταση της εμφάνισης της ώρας, η ώρα αποθηκεύεται με την ενέργεια εξόδου save time. Όταν μπαίνουμε στις καταστάσεις ρύθμισης των ωρών και των λεπτών, προκαλούνται οι ενέργειες εισόδου blinkHours και blinkMinutes αντίστοιχα. Πέρα από τις μεταβάσεις από μία κατάσταση σε άλλη, μπορούμε να έχουμε και αυτομεταβάσεις. Μία αυτομετάβαση συμβολίζεται ως μία μετάβαση που ξεκινάει και καταλήγει στην ίδια κατάσταση. Παραδείγματα αυτομεταβάσεων βλέπουμε στις καταστάσεις ρύθμισης των ωρών και των λεπτών, για τα πατήματα των πλήκτρων «+» και «-» που έχουν ως αποτέλεσμα την αύξηση ή μείωση κατά 1 ωρών ή των λεπτών αντίστοιχα ενώ το ρολόι παραμένει στην ίδια κατάσταση. Οι σύνθετες καταστάσεις είναι χρήσιμες όταν έχουμε μεταβάσεις από ένα σύνολο καταστάσεων σε μία άλλη κατάσταση όταν προκύψει κάποιο συμβάν. Στο παράδειγμα αν είμαστε στην κατάσταση ρύθμισης της ώρας συνολικά, ωρών ή λεπτών, και ο χρήστης πατήσει το πλήκτρο ακύρωσης cancel, θα μεταφερθούμε στην κατάσταση εμφάνισης της ώρα και με εκτέλεση της ενέργεια restoreTime, η ώρα θα αποκατασταθεί στην τιμή που είχε όταν βγήκαμε από την κατάσταση εμφάνισης της ώρας. Θα μπορούσαμε να εμφανίσουμε τη μετάβαση για το cancel σε κάθε κατάσταση της ρύθμισης της ώρας ξεχωριστά αλλά αυτό όμως θα έκανε το διάγραμμα ακατάστατο και επομένως λιγότερο κατανοητό. Η σύνθετη κατάσταση μας βοηθάει να απεικονίσουμε την κοινή μετάβαση ως μετάβαση από τη σύνθετη κατάσταση συνολικά προ της κατάσταση εμφάνισης της ώρας. η σημασία της μετάβασης από τη σύνθετη κατάσταση προς κάποια άλλη κατάσταση είναι ότι η μετάβαση αυτή ισχύει για οποιαδήποτε υποκατάσταση της σύνθετης κατάστασης. Επιπλέον υπάρχει και η δυνατότητα μετάβασης από μία εξωτερική κατάσταση σε μία υποκατάσταση, όπως συμβαίνει για παράδειγμα με τη μετάβαση από την κατάσταση εμφάνισης της ώρας στην υποκατάσταση ρύθμισης των ωρών μόλις ο χρήστης πατήσει το πλήκτρο Set. Το σύμβολο που χρησιμοποιείται για να υποδηλώσει το τέλος ενός διαγράμματος κατάστασης είναι ένας κύκλος και στο κέντρο του ένας γεμισμένος μαύρος κύκλος.



Εικ. 31 Παράδειγμα διαγράμματος καταστάσεων

Πολλές φορές θέλουμε να απεικονίσουμε ταυτόχρονα διαγράμματα κατάστασης στα οποία ένα αντικείμενο βρίσκεται ταυτόχρονα σε δύο ή και περισσότερες καταστάσεις. Για το σκοπό αυτόν μπορούμε να χωρίσουμε μία κατάσταση σε δύο ή και περισσότερα μέρη με μία διακεκομμένη γραμμή και να δείξουμε τα ταυτόχρονα διαγράμματα κατάστασης στα διαφορετικά αυτά μέρη. Η μετάβαση από μία εξωτερική κατάσταση στη σύνθετη ταυτόχρονη κατάσταση συνεπάγεται ότι το αντικείμενο θα βρεθεί ταυτόχρονα στις αρχικές ψευδοκαταστάσεις των ταυτόχρονων καταστάσεων. Στην εικόνα 32 απεικονίζεται ένα διάγραμμα με ταυτόχρονες καταστάσεις για έναν υποψήφιο των πανελληνίων εξετάσεων.

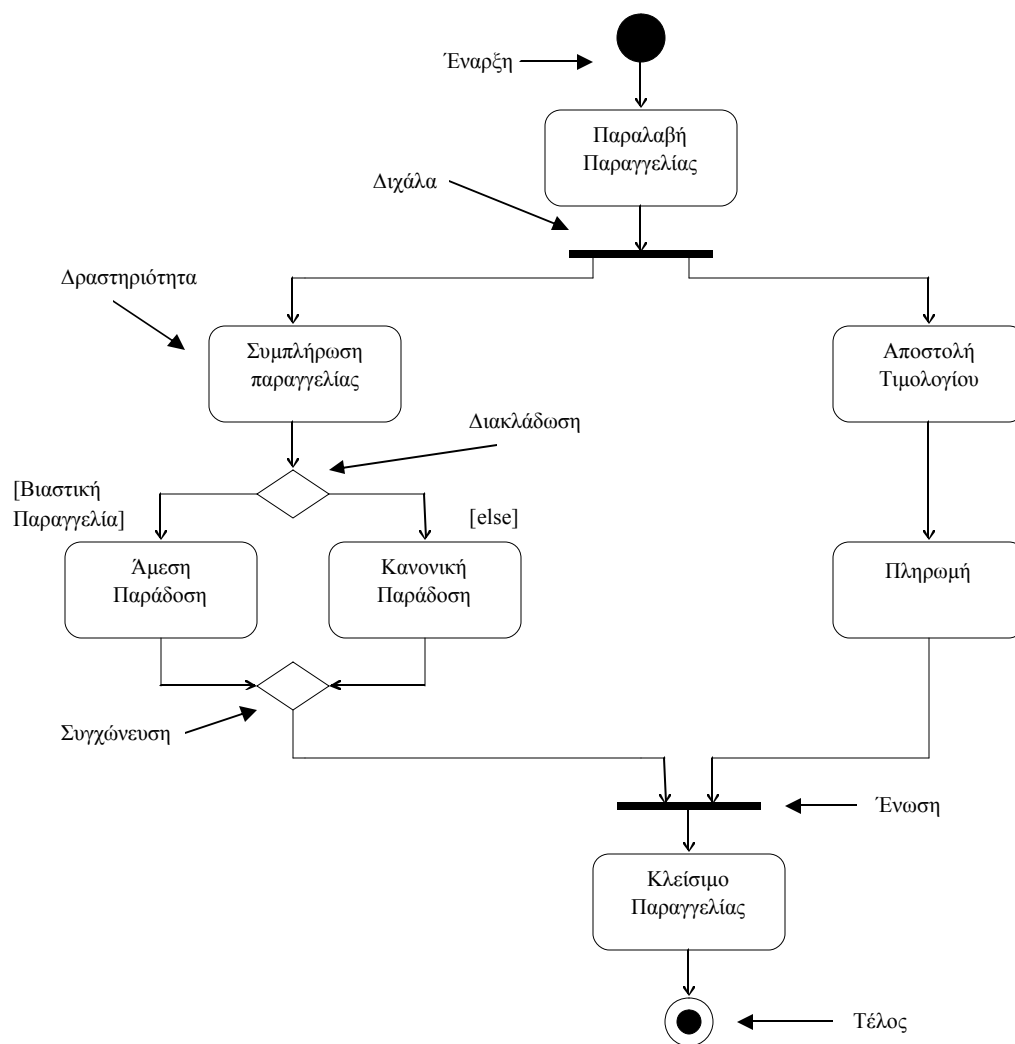


Εικ. 32 Παράδειγμα διαγράμματος καταστάσεων για αντικείμενο με ταυτόχρονες καταστάσεις

## 2.4 ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Το διάγραμμα δραστηριοτήτων (activity diagram) είναι μια παραλλαγή του διαγράμματος καταστάσεων στο οποίο οι κόμβοι αναπαριστούν καταστάσεις ενεργειών και οι μεταβάσεις λαμβάνουν χώρα με την ολοκλήρωση αυτών των ενεργειών και όχι ως συνέπεια ενός συμβάντος όπως συμβαίνει με το διάγραμμα καταστάσεων. Έτσι, το μεγαλύτερο μέρος της ορολογίας συμφωνεί με την ορολογία του διαγράμματος κατάστασης. Τυπικά χρησιμοποιούμε τα διαγράμματα δραστηριοτήτων για την περιγραφή υπολογισμών ή ροών εργασιών (work flows). Στην εικόνα 33 το βασικό σύμβολο είναι αυτό για την κατάσταση δραστηριότητας (activity state) ή απλώς δραστηριότητα (activity). Μια δραστηριότητα είναι μια κατάσταση στην οποία συμβαίνει κάτι, είτε μια διαδικασία του πραγματικού κόσμου όπως η προετοιμασία μια επιστολής είτε η εκτέλεση μιας ενότητας λογισμικού όπως η εκτέλεση μιας μεθόδου μιας κλάσης. Το διάγραμμα δραστηριότητας περιγράφει την ακολουθία των δραστηριοτήτων, και παρέχει υποστήριξη και της υπό συνθήκη αλλά και της παράλληλης συμπεριφοράς. Η παράλληλη συμπεριφορά παριστάνεται με διχάλες και ενώσεις. Μια διχάλα (fork) έχει μία εισερχόμενη μετάβαση και πολλές εξερχόμενες. Όταν συμβαίνει η εισερχόμενη μετάβαση, ακολουθούνται παράλληλα όλες οι εξερχόμενες μεταβάσεις. Έτσι στην εικόνα 33 αφού έχετε πάρει μία παραγγελία, συμπληρώνετε την παραγγελία και παράλληλα στέλνετε το τιμολόγιο. Όταν έχετε μία παράλληλη συμπεριφορά, χρειάζεται να συγχρονιστείτε. Δεν κλείνουμε την παραγγελία πριν την παράδοση της και την αποπληρωμή της. Αυτό το δείχνουμε με μία ένωση (join), πριν από τη δραστηριότητα «κλείσιμο παραγγελίας». Στο πλαίσιο μιας ένωσης, η εξερχόμενη μετάβαση ακολουθείται μόνο όταν όλες οι καταστάσεις των εισερχόμενων μεταβάσεων έχουν ολοκληρώσει τις δραστηριότητες τους. Οι διχάλες και οι ενώσεις πρέπει να ταιριάζουν.

Στην απλούστερη περίπτωση αυτό σημαίνει ότι, κάθε φορά που έχετε μία διχάλα, πρέπει να έχετε και μία ένωση που ενώνει μαζί τα νήματα εκτέλεσης τα οποία ξεκίνησαν από αυτή τη διχάλα. Η υπό συνθήκη συμπεριφορά περιγράφεται από διακλάδωσεις και συγχωνεύσεις. Μια διακλάδωση (branch) έχει μία μόνον εισερχόμενη μετάβαση και πολλές φρουρούμενες εξερχόμενες μεταβάσεις. Μόνο μία από τις εξερχόμενες μεταβάσεις μπορεί να ακολουθηθεί, και έτσι οι φρουροί πρέπει να είναι αμοιβαία αποκλειόμενες συνθήκες. Η χρήση της φράσης [else] (αλλιώς) στη θέση ενός φρουρού σημαίνει ότι η μετάβαση αλλιώς πρέπει να χρησιμοποιηθεί αν όλοι οι υπόλοιποι φρουροί σε αυτή τη διακλάδωση είναι ψευδείς. Στην εικόνα 33 αφού έχει συμπληρωθεί μία παραγγελία, υπάρχει μία διακλάδωση. Αν πρόκειται για βιαστική παραγγελία, κάνετε μία άμεση παράδοση, διαφορετικά κάνετε μία κανονική παράδοση. Μια συγχώνευση (merge) έχει πολλές εισερχόμενες μεταβάσεις και μία μόνον έξοδο. Μία συγχώνευση σημαίνει το τέλος μίας υπό συνθήκη συμπεριφοράς που ξεκίνησε με διακλάδωση.



Εικ. 33 Παράδειγμα διαγράμματος δραστηριοτήτων

## 2.5 ΔΙΑΓΡΑΜΜΑΤΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ

Τα διαγράμματα αλληλεπίδρασης (interaction diagrams) χρησιμοποιούνται για την οπτική αναπαράσταση των διαφόρων σεναρίων των περιπτώσεων χρήσης του συστήματος. Απεικονίζουν ένα σενάριο σαν στιγμιότυπα αντικειμένων που αλληλεπιδρούν μεταξύ τους ανταλλάσσοντας μηνύματα. Υπάρχουν δύο τύποι διαγραμμάτων αλληλεπίδρασης, οι οποίοι είναι ισοδύναμοι. Το διάγραμμα ακολουθίας (sequence diagram) και το διάγραμμα συνεργασίας (collaboration diagram). Το διάγραμμα ακολουθίας δίνει έμφαση στη χρονική ακολουθία των μηνυμάτων ενώ το διάγραμμα συνεργασίας απεικονίζει τους συνδέσμους μεταξύ των αντικειμένων σε αυτό και για να γίνει προφανής η χρονική σειρά των μηνυμάτων απαιτείται η αρίθμηση τους. Το διάγραμμα συνεργασίας απαιτεί λιγότερο σχεδιαστικό χώρο μια και η διάταξη των αντικειμένων είναι ελεύθερη στο χώρο και επομένως προτιμάται όταν έχουμε πολλά αντικείμενα σε μία αλληλεπίδραση ή όταν σχεδιάζουμε ένα διάγραμμα με το χέρι. Επειδή τα διαγράμματα ακολουθίας προσφέρουν καλύτερη απεικόνιση της χρονικής αλληλουχίας των μηνυμάτων, οι περισσότεροι σχεδιαστές τα προτιμούν.

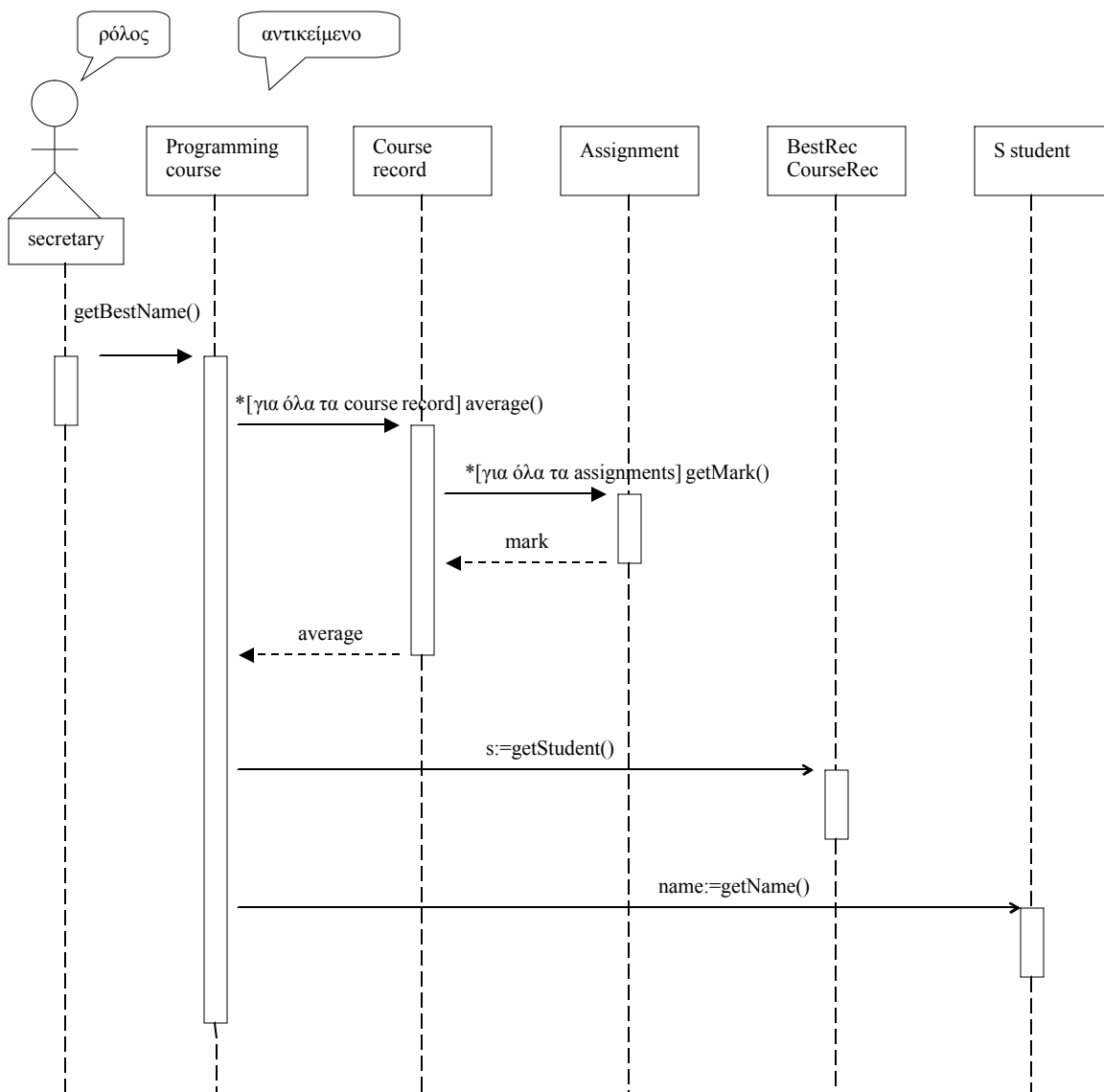
Στα διαγράμματα ακολουθίας (sequence diagrams) έχουμε την απεικόνιση μιας συνεργασίας αντικειμένων. Τα αντικείμενα συμβολίζονται με παραλληλόγραμμα στα οποία αναγράφεται το όνομα του αντικειμένου ακολουθούμενο από μία άνω κάτω τελεία και στη συνέχεια το όνομα της κλάσης. Αν δε χρειάζεται το όνομα του αντικειμένου, μπορεί να παραλειφθεί. Κάτω από κάθε αντικείμενο εκτείνεται μία διακεκομμένη γραμμή που ονομάζεται γραμμή ζωής (life line) του αντικειμένου. Αν πρέπει να δείξουμε τον τερματισμό της ζωής ενός αντικειμένου, μπορούμε να βάλουμε ένα X στο τέλος της γραμμής ζωής του. Τα αντικείμενα γράφονται από αριστερά προς τα δεξιά και στον οριζόντιο άξονα. Τα αντικείμενα ανταλλάσσουν μηνύματα, τα οποία στην επίσημη ορολογία της UML ονομάζονται ερεθίσματα (stimuli). Ερέθισμα μπορεί να είναι οτιδήποτε από τα εξής:

- Κλήση μια λειτουργίας: όταν ένα αντικείμενο καλεί μια λειτουργία ενός άλλου αντικειμένου. Πρόκειται για σύγχρονο μήνυμα, δηλαδή ο αποστολέας του μηνύματος θα πρέπει να περιμένει την ολοκλήρωση της λειτουργίας για να συνεχίσει. Συμβολίζεται με ένα βέλος από τον αποστολέα προς τον παραλήπτη. Η κεφαλή του βέλους είναι γεμισμένη με μαύρο χρώμα. Πάνω από το βέλος αναγράφεται το όνομα της λειτουργίας που καλείται, με τις ενδεχόμενες παραμέτρους σε παρενθέσεις.
- Σήμα: όταν ένα αντικείμενο αποστέλλει ένα ασύγχρονο μήνυμα σε ένα άλλο αντικείμενο. Τυπικά ασύγχρονα μηνύματα συναντάμε σε πολυνηματικές (multithreading) εφαρμογές, όπου ένα μήνυμα τοποθετείται σε κάποια ουρά ενός νήματος εκτέλεσης ενώ το ενεργό αντικείμενο παραλήπτης θα επεξεργαστεί το μήνυμα σε κάποια επόμενη χρονική στιγμή. Η διαφορά με την κλήση στο συμβολισμό είναι πως η κατάληξη είναι ένα ανοιχτό βέλος. Η σημασιολογική διάκριση είναι πως το αντικείμενο αποστολέας του ασύγχρονου μηνύματος μπορεί να συνεχίσει την επεξεργασία χωρίς να περιμένει την ολοκλήρωση της επεξεργασίας του μηνύματος που απέστειλε.

- Δημιουργία αντικειμένου: είναι ένα ερέθισμα που καταλήγει στη δημιουργία ενός νέου αντικειμένου και όχι στη γραμμή ζωής κάποιου υπάρχοντος αντικειμένου. Ένα τυπικό παράδειγμα είναι η απεικόνιση της κλήσης της μεθόδου κατασκευής ενός αντικειμένου.
- Επιστροφή κλήσης: είναι ένα διακεκομμένο βέλος το οποίο συμβολίζει την επιστροφή από μία κλήση. Θα πρέπει φυσικά να έχει προηγηθεί η κλήση και να έχουν ολοκληρωθεί και άλλες πιθανές υποκλήσεις που περιέχει η εκτέλεση της συγκεκριμένης λειτουργίας. Πάνω στο διακεκομμένο βέλος αναγράφεται η τιμή επιστροφής, αν υπάρχει. Συνήθως δεν εμφανίζουμε τις επιστροφές οι οποίες εννοούνται στην περίπτωση των κλήσεων λειτουργιών. Επιπλέον, αν θέλουμε, μπορούμε να δείξουμε την τιμή επιστροφής στην κλήση της λειτουργίας.
- Καταστροφή αντικειμένου: είναι ένα ερέθισμα που καταλήγει στο σύμβολο τερματισμού της ζωής ενός άλλου αντικειμένου. Σηματοδοτεί την καταστροφή ενός αντικειμένου ως αποτέλεσμα ενός μηνύματος από ένα άλλο αντικείμενο.

Η δημιουργία ενός νέου αντικειμένου απεικονίζεται με ένα μήνυμα που αποστέλλεται σε ένα αντικείμενο και το οποίο καταλήγει πάνω στο πλαίσιο του νέου αντικειμένου που δημιουργείται. Αν το νέο αντικείμενο που δημιουργείται καλεί αμέσως μετά τη δημιουργία του μια λειτουργία τότε αρχίζει ένα πλαίσιο ενεργοποίησης αμέσως με την αρχή της γραμμής ζωής του αντικειμένου. Η καταστροφή ενός αντικειμένου συμβολίζεται με ένα X στο τέλος της γραμμής ζωής του. Αν η καταστροφή του αντικειμένου προκαλείται από ένα άλλο αντικείμενο, αυτό μπορούμε να το δείξουμε με ένα σήμα που αποστέλλεται από ένα αντικείμενο στο άλλο και καταλήγει πάνω στο X. Όταν ένα αντικείμενο καλεί μια λειτουργία στον εαυτό του, αυτό απεικονίζεται σαν μία κλήση που ξεκινάει από το αντικείμενο και καταλήγει πάλι σε αυτό. Το πλαίσιο ενεργοποίησης εμφανίζεται ένθετο στο πλαίσιο ενεργοποίησης της λειτουργίας στην οποία γίνεται η αυτόκληση. Στο διάγραμμα της εικόνας 34 ο ρόλος Secretary (γραμματέας) αποστέλλει το μήνυμα `getBestName()`, που είναι λειτουργία της κλάσης «Course», από την οποία ξεκινάει η διαδικασία ανάκτησης του ονόματος του καλύτερου φοιτητή για ένα μάθημα. Παρατηρήστε το πλαίσιο ενεργοποίησης (activation frame) που ξεκινάει στην αρχή της κλήσης. Το πλαίσιο ενεργοποίησης συμβολίζει τη χρονική διάρκεια της εκτέλεσης μιας λειτουργίας. Η χρονική διάρκεια της εκτέλεσης της λειτουργία `getBestName()` περιέχει τις χρονικές διάρκειες εκτέλεσης των λειτουργιών `average()`, `getmark()` διότι περιέχονται σε αυτήν. Για να βρούμε ποιος είναι ο καλύτερος φοιτητής του μαθήματος «programming» καλούμε επαναληπτικά τη λειτουργία `average()` για όλες τις εγγραφές του «Course». Για να συμβολίσουμε την επανάληψη, τοποθετούμε έναν αστερίσκο στην κλήση της μεθόδου και μέσα σε αγκύλες γράφουμε τη συνθήκη της επανάληψης. Με τη σειρά της, για να υπολογίσει το μέσο όρο ενός `CourseRecord`, η λειτουργία `average()` πρέπει να ανακτήσει το βαθμό κάθε εργασίας για όλες τις εργασίες της εγγραφής. Αυτό γίνεται με την επαναληπτική κλήση της λειτουργίας `getMark()`. Παρατηρήστε ότι το πλαίσιο ενεργοποίησης της `getMark()` περιέχεται εξ ολοκλήρου στο πλαίσιο ενεργοποίησης της `average()`. Οι λειτουργίες `average()` και `getMark()` επιστρέφουν, αντίστοιχα, το μέσο όρο μιας εγγραφής και μιας εργασίας. Για να δείξουμε την επιστροφή τιμής χρησιμοποιούμε το διακεκομμένο βέλος. Αφού βρεθεί με τη διαδικασία της ανάκτησης των μέσων όρων των εργασιών η εγγραφή με τον καλύτερο βαθμό, στη συνέχεια καλείται σε αυτήν την εγγραφή η λειτουργία

getStudent(), η οποία επιστρέφει το φοιτητή της. Στη συνέχεια, καλείται σε αυτό το αντικείμενο η λειτουργία getName() για την ανάκτηση του ονόματος του φοιτητή.



Εικ. 34 Παράδειγμα διαγράμματος ακολουθίας

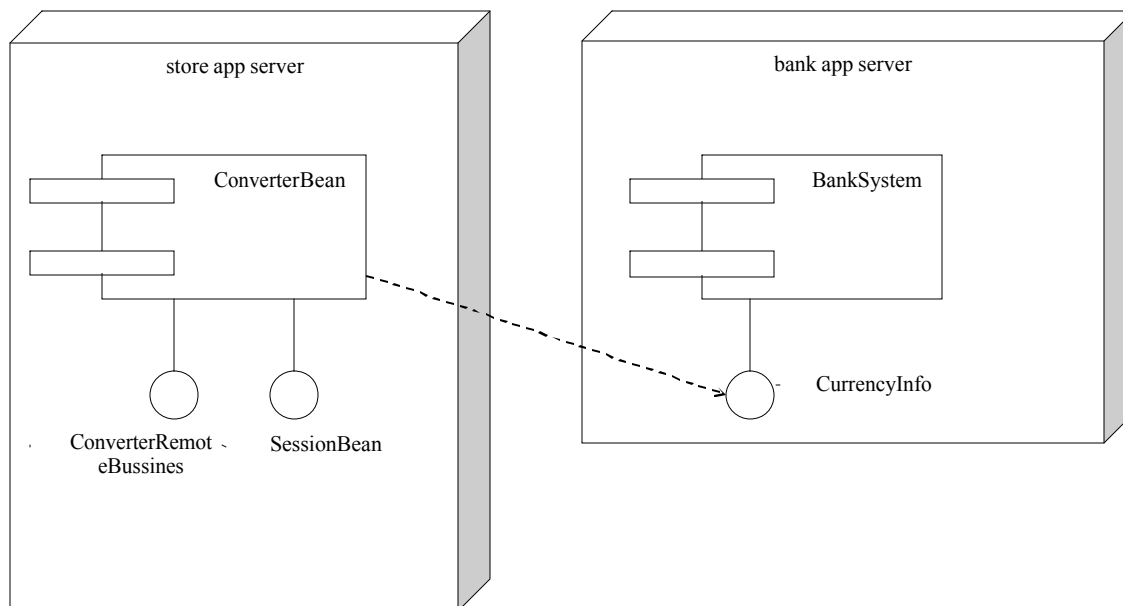
## 2.6 ΔΙΑΓΡΑΜΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Η UML παρέχει δύο είδη διαγραμμάτων υλοποίησης (deployment diagrams). Τα διαγράμματα συστατικών και τα διαγράμματα διάταξης. Συστατικό (component) είναι μία ενότητα λογισμικού που εκθέτει τις λειτουργίες της μέσω ενός συνόλου δημόσιων διασυνδέσεων και μπορεί να συνδεθεί δυναμικά με άλλα συστατικά για τη δημιουργία μεγαλύτερων συστατικών και εφαρμογών. Επομένως, ένα συστατικό έχει τρία σημαντικά χαρακτηριστικά:

- Διαθέτει τις λειτουργίες του μέσω διασυνδέσεων και άρα η εσωτερική του υλοποίηση είναι κρυφή
- Μπορεί να συνδεθεί δυναμικά με άλλα συστατικά
- Είναι επαναχρησιμοποιήσιμο και δεν έχει γίνει για μία μόνο εφαρμογή

Για να αναπαραστήσουμε τα συστατικά με τη UML χρησιμοποιούμε το διάγραμμα συστατικών (component diagram) τα συστατικά παριστάνονται σαν παραλληλόγραμμα, από την αριστερή πλευρά των οποίων εξέρχονται δύο μικρότερα παραλληλόγραμμα. Οι διασυνδέσεις που εκθέτει ένα συστατικό συμβολίζονται με μικρούς λευκούς κύκλους, οι οποίοι συνδέονται μέσω ευθύγραμμων τμημάτων με το συστατικό (συμβολισμός lollipop). Τα συστατικά μπορούν να εξαρτώνται από άλλα συστατικά με χρήση των διασυνδέσεων. Η εξάρτηση αυτή απεικονίζεται με ένα κατευθυνόμενο διακεκομμένο βέλος, με αρχή από το συστατικό που χρησιμοποιεί τη διασύνδεση και κατάληξη στη διασύνδεση του συστατικού που την παρέχει. Το διάγραμμα διάταξης μπορεί να χρησιμοποιηθεί κυρίως σε καταναμημένα συστήματα για να δείξει τη φυσική διάταξη των διαφόρων τμημάτων του λογισμικού. Δηλαδή το διάγραμμα διάταξης δείχνει τις φυσικές σχέσεις μεταξύ των συστατικών του υλικού και του λογισμικού σε ένα σύστημα. Εδώ μπορούν να παρουσιαστούν και συστατικά τα οποία είναι στιγμιότυπα των συστατικών που απεικονίζονται στο διάγραμμα συστατικών. Το διάγραμμα περιέχει κόμβους (nodes) οι οποίοι περιέχουν τις εφαρμογές και τα συστατικά που εκτελούνται σε αυτούς. Οι κόμβοι του συστήματος συμβολίζονται με κύβους, στους οποίους αναγράφεται το όνομα του κόμβου. Σε επίπεδο υλοποίησης υπάρχουν τεχνολογίες για την υλοποίηση συστατικών όπως το COM+ (Component Object Model) της εταιρίας Microsoft και τα EJB (Enterprise Java Beans). Στην εικόνα 35 είναι σχεδιασμένο το διάγραμμα συστατικών πάνω στο διάγραμμα διάταξης και απεικονίζεται ένα EJB Session Bean με το όνομα «Converter Bean», το οποίο αφορά ένα μετατροπέα νομισματικών ισοτιμιών. Το συγκεκριμένο συστατικό έχει δύο διασυνδέσεις: μία που περιέχει τις πρότυπες μεθόδους των EJB για τα session EJBs (τη διασύνδεση «Session Beans») και μία με τις εξειδικευμένες μεθόδους για τη μετατροπή των νομισματικών μονάδων (ConverterRemoteBusiness). Το συστατικό αυτό μπορεί να χρησιμοποιηθεί από πελάτες που θα δημιουργήσουν ένα στιγμιότυπο του μέσω της διασύνδεσης «SessionBean» και θα καλέσουν τις μεθόδους που παρέχονται μέσω της διασύνδεσης «ConverterRemoteBusiness» για τη μετατροπή των νομισματικών μονάδων. Το συστατικό χρησιμοποιεί ένα άλλο συστατικό, το «BankSystem» και πιο συγκεκριμένα τη διασύνδεση «CurrencyInfo» του συστατικού αυτού για να πάρει τις ισοτιμίες μεταξύ των διαφόρων νομισμάτων. Το στιγμιότυπο του συστατικού «ConverterBean» εκτελείται στο διακομιστή εφαρμογών (Application Server) του καταστήματος ενώ το στιγμιότυπο του συστατικού «BankSystem» εκτελείται στον διακομιστή εφαρμογών της τράπεζας.





Εικ. 35 Παράδειγμα διαγράμματος διάταξης

## Γ. ΠΑΡΑΔΕΙΓΜΑ

Στη συνέχεια θα αναπτύξουμε ένα παράδειγμα για να κάνουμε όσο γίνεται πιο κατανοητή τη μέθοδο της ενοποιημένης διεργασίας. Γενικά οι εφαρμογές περιλαμβάνουν στοιχεία διασύνδεσης χρήστη και εφαρμογής, στοιχεία της βασικής λογικής της εφαρμογής, στοιχεία πρόσβασης σε βάσεις δεδομένων και στοιχεία αλληλεπίδρασης με άλλα συστήματα λογισμικού. Παρόλο που η μέθοδος μπορεί να εφαρμοστεί σε όλα τα επίπεδα, εμείς θα επικεντρωθούμε στο επίπεδο της λογικής της εφαρμογής. Το αντικείμενο του παραδείγματος μας θα είναι ένα σύστημα λογισμικού για ταμειακή μηχανή, που θα χρησιμοποιείται σε ένα τυπικό κατάστημα λιανικής πώλησης. Το λογισμικό θα καταγράφει τις πωλήσεις και θα χειρίζεται τις πληρωμές.

### 1. ΣΥΛΛΗΨΗ

Επειδή το έργο είναι σχετικά μικρό, η φάση της σύλληψης μπορεί να ολοκληρωθεί σε δύο επαναλήψεις. Κατά αυτές τις επαναλήψεις θα ανακαλύψουμε τις βασικές περιπτώσεις χρήσης και θα αποκτήσουμε μια γενική εικόνα του συστήματος και του τι ακριβώς έχουμε να κάνουμε στη συνέχεια.

#### 1.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ

Η πρώτη επανάληψη είναι αυτή που θα σηματοδοτήσει και την έναρξη του έργου. Επειδή έχει προαποφασιστεί η πραγματοποίηση του έργου δεν μπαίνουμε σε διαδικασία ελέγχου αν συμφέρει να γίνει το έργο ή όχι. Επίσης κανονίζουμε συναντήσεις με όλους τους εμπλεκόμενους και κυρίως με τους μελλοντικούς χρήστες, για να μάθουμε όσα περισσότερα μπορούμε για το έργο. Από αυτές τις συναντήσεις θα καταγράψουμε τη βασική λειτουργία του συστήματος και θα καταγράψουμε ονομαστικά τις περιπτώσεις χρήσης του συστήματος. Επίσης στη συνέχεια θα προχωρήσουμε σε οριοθέτηση της εμβέλειας του συστήματος και στην αναγνώριση των χρηστών του συστήματος.

Η βασική λειτουργία του συστήματος είναι η εξής: το σύστημα θα διαχειρίζεται τις πωλήσεις και τις επιστροφές σε ένα κατάστημα λιανικής. Ο πελάτης θα παίρνει κάποια προϊόντα και θα πηγαίνει στο ταμείο, όπου βρίσκεται ο ταμίας, για να τα πληρώσει και να φύγει. Από τα προηγούμενα μπορούμε να συμπεράνουμε ότι μία θα είναι η περίπτωση χρήσης του συστήματος, η «πώληση» (εικ.36). Επίσης πραγματοποιούμε μια αναγνώριση των χρηστών του συστήματος και των στόχων τους και τους καταγράφουμε. Οι χρήστες του συστήματος μας είναι ο ταμίας και ο πελάτης (εικ.37).

Περιπτώσεις χρήσης
Πώληση

Εικ. 36 Πίνακας περιπτώσεων χρήσης

Χρήστης	Στόχοι
Ταμίας	Να διεκπεραιώσει μια πώληση
Πελάτης	Να αγοράσει προϊόντα

Εικ. 37 Πίνακας με τους χρήστες του συστήματος και τους στόχους τους

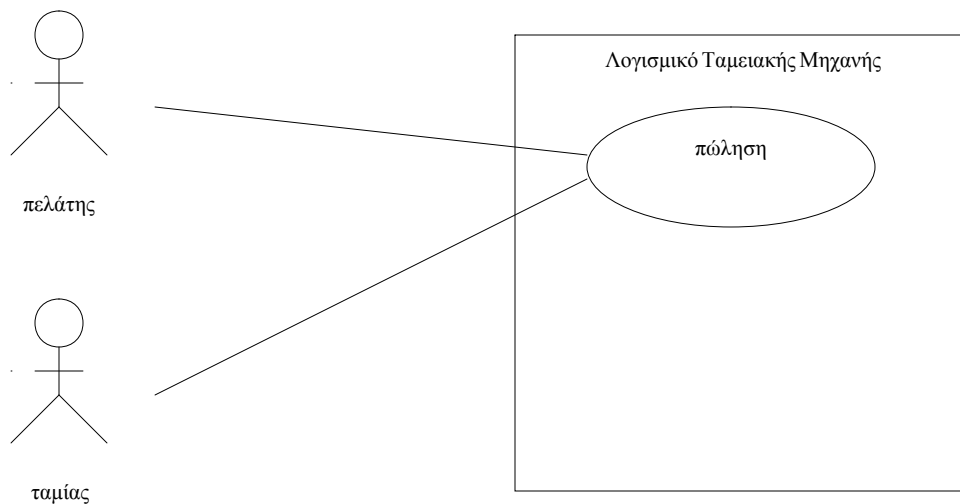
Με το τέλος της πρώτης επανάληψης θα έχουμε πραγματοποιήσει τα εξής:

- Καταγράψαμε ονομαστικά τις περιπτώσεις χρήσης του συστήματος.
- Προσδιορίσαμε τους χρήστες του συστήματος και τους στόχους τους.

## 1.2 ΔΕΥΤΕΡΗ ΕΠΑΝΑΛΗΨΗ

Στη δεύτερη επανάληψη της φάσης της σύλληψης θα προχωρήσουμε περαιτέρω και θα οριοθετήσουμε την εμβέλεια του συστήματος. Επίσης θα αναπτύξουμε το βασικό σενάριο της βασικής περίπτωσης χρήσης που έχουμε καταγράψει, δηλαδή της περίπτωσης χρήσης «πώληση». Έτσι θα αποκτήσουμε μια καλύτερη εικόνα για το σύστημα που έχουμε να αναπτύξουμε.

Οι πληροφορίες που διαθέτουμε τώρα μας δίνουν τη δυνατότητα να σχηματίσουμε ένα αρχικό διάγραμμα περιπτώσεων χρήσης, όπου θα φαίνονται, εκτός από τις περιπτώσεις χρήσης και τους χρήστες, και το πλαίσιο της εμβέλειας του συστήματος (εικ.38).



Εικ. 38 Το σύστημα και οι χρήστες του συστήματος

Η βασική περίπτωση χρήσης στο παράδειγμα μας είναι η διαδικασία μιας τυπικής αγοράς ενός προϊόντος.

Πώληση: ένας πελάτης φτάνει στο ταμείο με κάποιο προϊόν για να αγοράσει. Ο ταμίας χρησιμοποιεί το σύστημα λογισμικού για να καταγράψει το προϊόν. Το σύστημα παρουσιάζει το τρέχων ποσό πληρωμής και πληροφορίες για το συγκεκριμένο προϊόν. Ο πελάτης πληρώνει και το σύστημα καταγράφει την πώληση. Ο πελάτης φεύγει από το κατάστημα με το προϊόν. Ο τρόπος καταγραφής της περίπτωσης χρήσης θα γίνει όπως παρακάτω.

*Περίπτωση Χρήσης ΠΧ1: Πώληση*

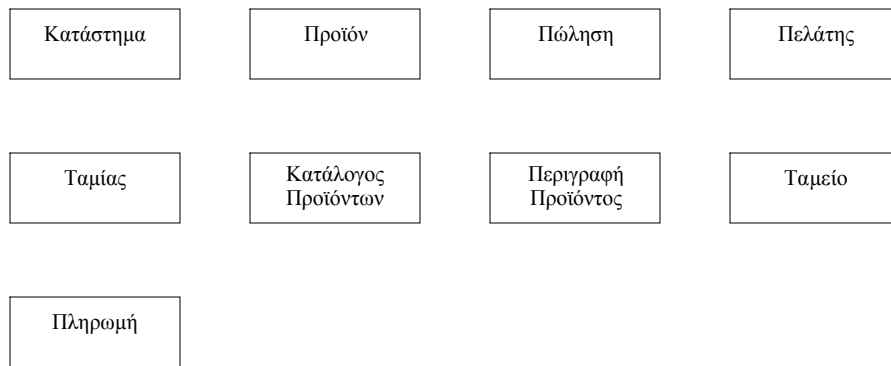
*Χρήστης: Ταμίας, Πελάτης*

*Βασικό σενάριο:*

- 1. ο πελάτης φτάνει στο ταμείο με προϊόντα για να αγοράσει.*
- 2. ο ταμίας ξεκινά μια καινούργια πώληση.*
- 3. ο ταμίας εισάγει τον κωδικό του προϊόντος.*
- 4. το σύστημα καταγράφει το προϊόν και εμφανίζει την περιγραφή του προϊόντος, την τιμή και το συνολικό ποσό.*
- 5. ο ταμίας επαναλαμβάνει τα βήματα 3 έως 4 μέχρι το τέλος των προϊόντων.*
- 6. το σύστημα παρουσιάζει το τελικό ποσό.*
- 7. ο πελάτης πληρώνει και φεύγει από το κατάστημα με το προϊόν.*

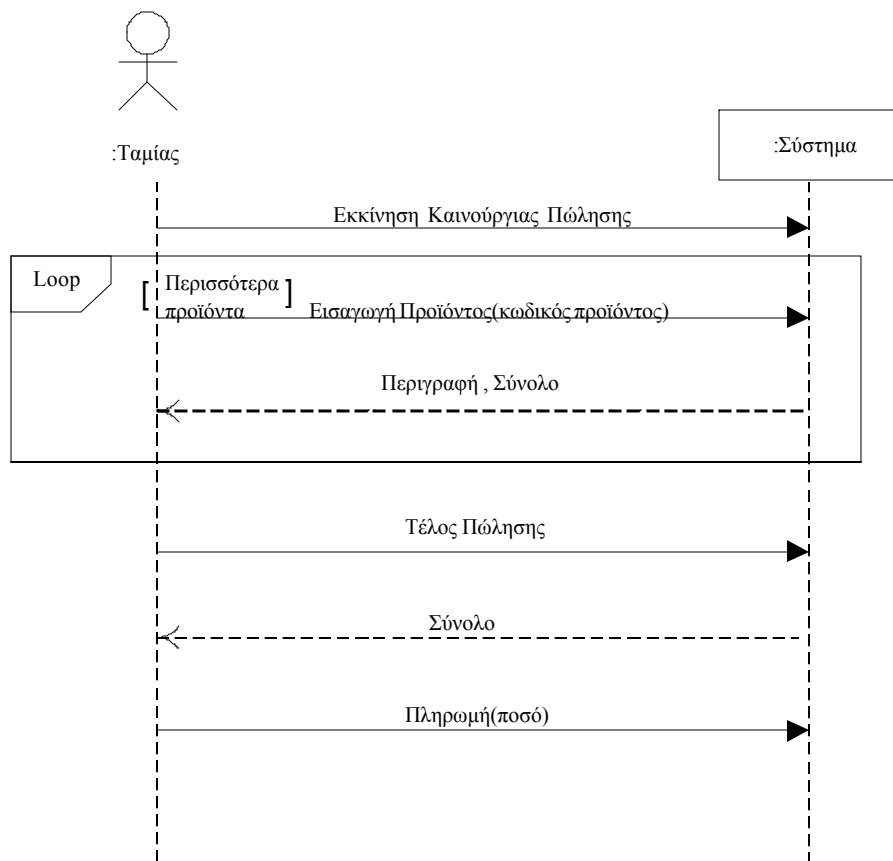
Αυτό είναι το βασικό σενάριο της περίπτωσης χρήσης «πώληση». Η περίπτωση χρήσης μπορεί να έχει και άλλα σενάρια τα οποία όμως θα ανακαλυφθούν παρακάτω σε επόμενες επαναλήψεις.

Στη συνέχεια θα προσδιορίσουμε τις αφαιρέσεις του πραγματικού κόσμου που απαιτούνται για την κατασκευή του συστήματος, δηλαδή τα κύρια εννοιολογικά αντικείμενα που πρόκειται να συμμετάσχουν σε αυτό το σύστημα. Η προσέγγιση αυτή απορρέει από τη γενικότερη αρχή δημιουργίας συστημάτων με αντικειμενοστραφή τρόπο, δηλαδή ότι το λογισμικό πρέπει να βασίζεται στα φυσικά αντικείμενα του πεδίου προβλήματος. Η αρχή αυτή βασίζεται στη θεωρία ότι ο πραγματικός κόσμος αλλάζει λιγότερο συχνά από τι οι προδιαγραφές ενός συστήματος λογισμικού. Στο παράδειγμα μας οι έννοιες τις οποίες μπορούμε να ξεχωρίσουμε και οι οποίες θα σχηματίσουν το αρχικό μοντέλο του πεδίου προβλήματος (domain model) είναι οι εξής: πώληση, ταμίας, πελάτης, κατάστημα, προϊόν, κατάλογος προϊόντων, περιγραφή προϊόντος και ταμείο (εικ. 39).



Εικ. 39 Μοντέλο του πεδίου προβλήματος στην δεύτερη επανάληψη της φάσης της σύλληψης

Στη παρούσα επανάληψη δεν δίνουμε σημασία στη καταγραφή των πεδίων ή των μεθόδων των κλάσεων του πραγματικού κόσμου και των σχέσεων μεταξύ τους, αλλά μόνο στην καταγραφή των κλάσεων. Επιπλέον ανάλυση του συστήματος που θα κάνουμε είναι με την δημιουργία του διαγράμματος αλληλεπίδρασης του συστήματος (εικ.40)



Εικ. 40 Διάγραμμα αλληλεπίδρασης του συστήματος

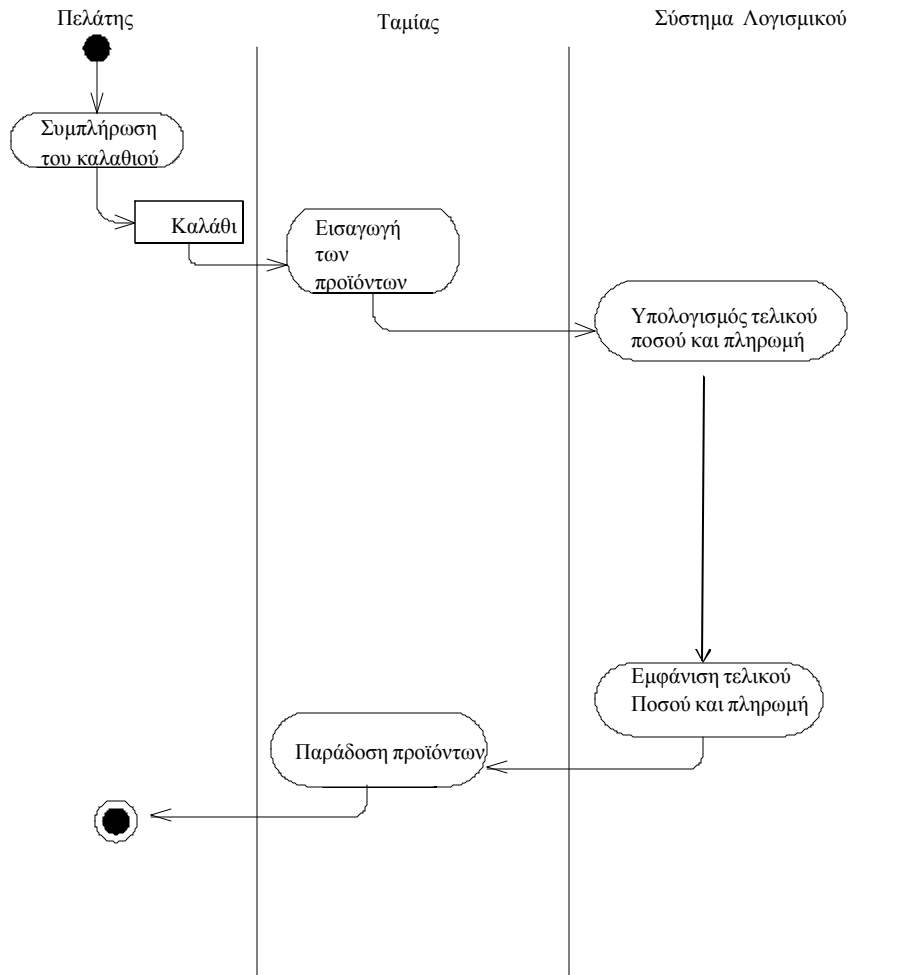
Τα μηνύματα που πηγαίνουν προς το σύστημα θα είναι τα μηνύματα που θα στέλνονται από τη διασύνδεση χρήστη (user interface) προς το σύστημα. Ενώ τα μηνύματα που φεύγουν από το σύστημα θα είναι τα μηνύματα που στέλνει το σύστημα στη διασύνδεση χρήστη.

## 2 ΕΠΕΞΕΡΓΑΣΙΑ

Για το παράδειγμα μας η φάση της επεξεργασίας θα χρειαστεί τρεις επαναλήψεις.

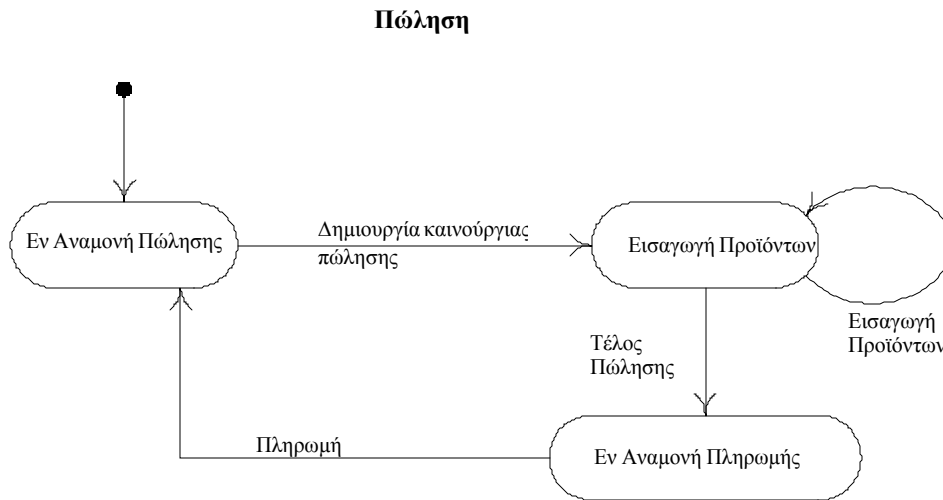
### 2.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ

Όσον αφορά την καταγραφή απαιτήσεων δεν έχουμε να προσθέσουμε τίποτα σε αυτή την επανάληψη. Από πλευράς ανάλυσης θα κατασκευάσουμε το διάγραμμα δραστηριοτήτων και το διάγραμμα καταστάσεων για την βασική περίπτωση χρήσης που παραδείγματός μας «πώληση». Τα διαγράμματα αυτά θα συμβάλουν στην περαιτέρω κατανόηση του συστήματος που έχουμε να αναπτύξουμε. Το διάγραμμα δραστηριοτήτων θα αποτυπώσει τις δραστηριότητες που συμβαίνουν για την περίπτωση χρήσης «πώληση» (εικ. 41).



Εικ. 41 Διάγραμμα δραστηριοτήτων για την περίπτωση χρήσης «πώληση»

Το διάγραμμα καταστάσεων θα περιγράψει τις καταστάσεις του συστήματος και το πώς αυτό αντιδρά σε συμβάντα (εικ. 42)



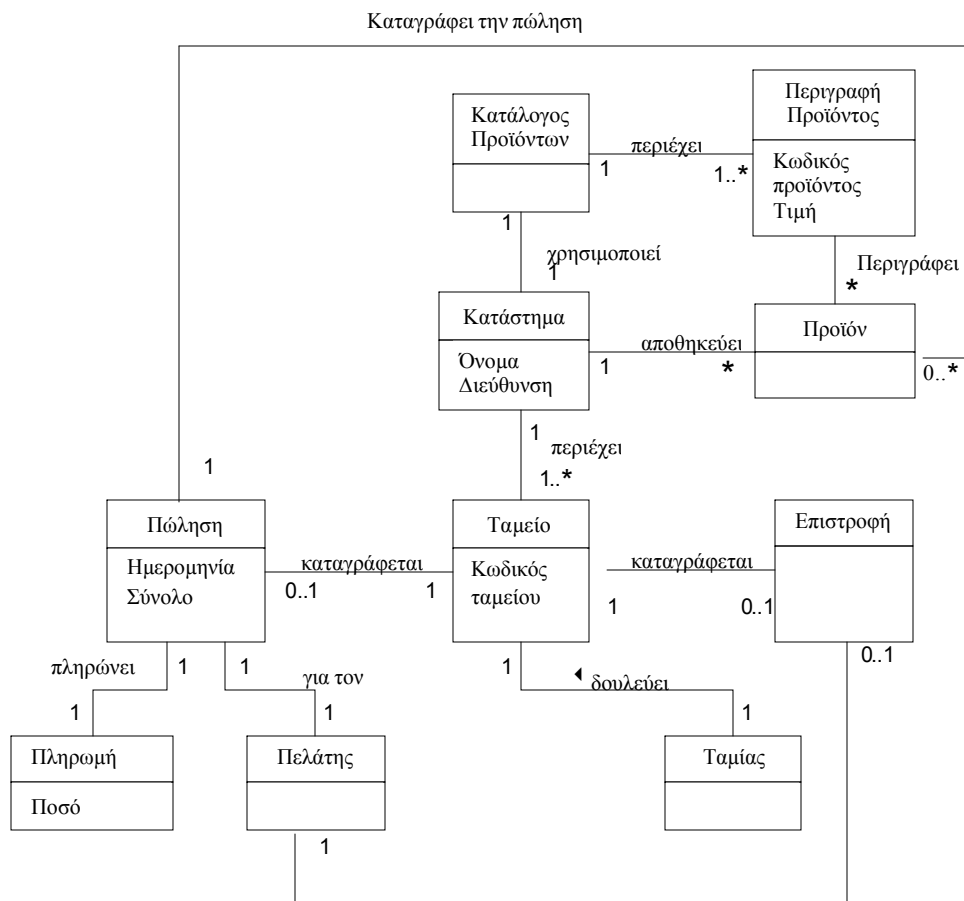
Εικ. 42 Διάγραμμα καταστάσεων για την περίπτωση χρήσης «πώληση»

Από πλευράς εργασιών σχεδίασης, υλοποίησης και ελέγχου δεν έχουμε να πραγματοποιήσουμε ακόμα τίποτα.

## 2.2 ΔΕΥΤΕΡΗ ΕΠΑΝΑΛΗΨΗ

Μετά τις σημαντικές πληροφορίες που αποκτήσαμε κατά την πρώτη επανάληψη με τη δημιουργία των διαγραμμάτων θα συνεχίσουμε την ανάλυση τη δημιουργία του μοντέλου του πεδίου προβλήματος. Σε αυτό θα προσθέσουμε τα χαρακτηριστικά για τις νοητές κλάσεις που έχουμε σχηματίσει (εικ. 43).

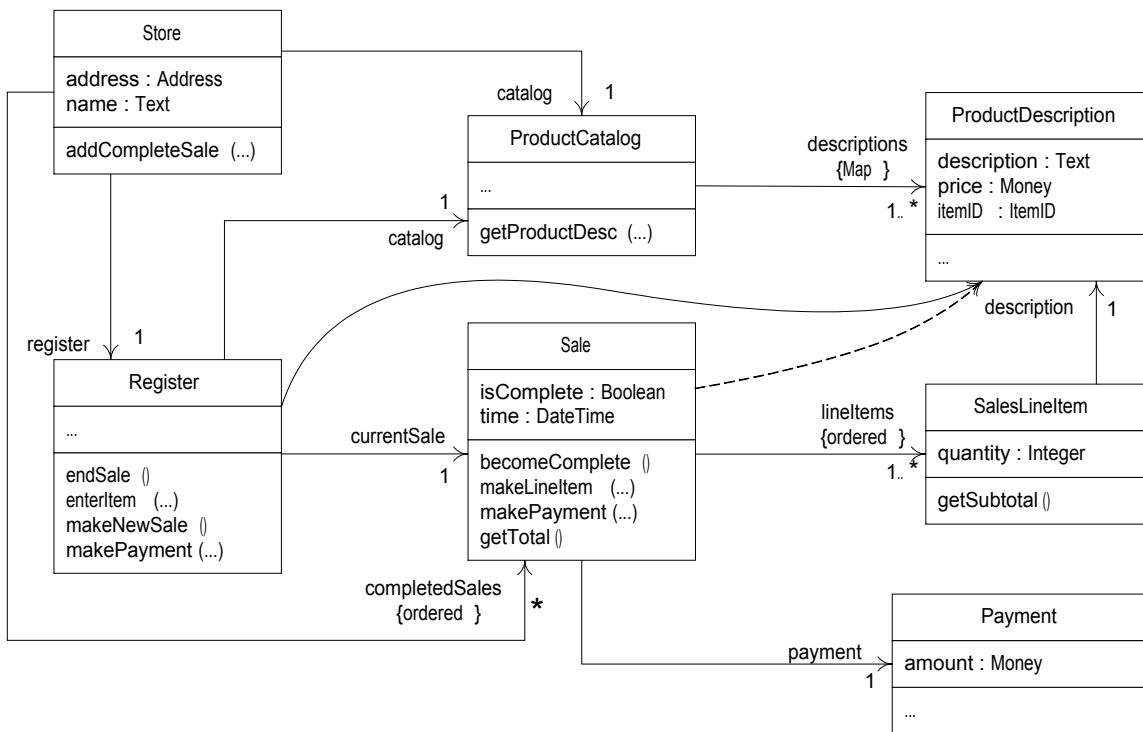




Εικ. 43 Το μοντέλο του πεδίου προβλήματος

### 2.3 ΤΡΙΤΗ ΕΠΑΝΑΛΗΨΗ

Στην τρίτη επανάληψη οι περιπτώσεις χρήσης έχουν σταθεροποιηθεί και καμιά μεταβολή τους δεν θα προκύψει. Θα σχεδιάσουμε το διάγραμμα κλάσεων, που θα περιέχει τις κλάσεις του συστήματος μας. Στο διάγραμμα κλάσεων τα ονόματα που θα χρησιμοποιήσουμε θα πηγάζουν από το μοντέλο του πεδίου του προβλήματος. Το διάγραμμα κλάσεων είναι μέρος του λογισμικού, ενώ το μοντέλο του πεδίου του προβλήματος είναι μέρος μιας νοητικής ανάλυσης της πραγματικότητας. Είναι φανερό ότι δεν είναι τα ίδια, αλλά δημιουργώντας ονόματα για τις κλάσεις που πηγάζουν από το μοντέλο του πεδίου του προβλήματος επιτυγχάνουμε το μικρότερο κενό αναπαράστασης μεταξύ του πραγματικού κόσμου και του συστήματος λογισμικού. Για παράδειγμα, το «κατάστημα» στο μοντέλο του πεδίου του προβλήματος, μας επηρεάζει στο να δημιουργήσουμε μια κλάση λογισμικού «κατάστημα» και στο διάγραμμα κλάσεων (εικ.44).



Εικ. 44 Διάγραμμα κλάσεων

### 3 ΚΑΤΑΣΚΕΥΗ

Μία επανάληψη θα χρειαστεί κατά τη φάση της κατασκευής. Τα παραγόμενα που δημιουργήθηκαν κατά την εργασία της σχεδίασης στις προηγούμενες επαναλήψεις, δηλαδή τα διαγράμματα αλληλεπίδρασης και διαγράμματα κλάσεων που αποτελούν και το μοντέλο σχεδίασης, θα χρησιμοποιηθούν για τη δημιουργία του κώδικα στις επόμενες επαναλήψεις. Πρέπει να επισημάνουμε ότι αν χρειαζόταν θα μπορούσαμε να προχωρήσουμε στη δημιουργία κώδικα και στις προηγούμενες επαναλήψεις, όπως για παράδειγμα αν έπρεπε να φτιάξουμε ένα μέρος του συστήματος για να το παρουσιάσουμε στον πελάτη ή για να λύσουμε δικές μας απορίες. Η δημιουργία κώδικα μπορεί να γίνει σε μια οποιαδήποτε αντικειμενοστραφή γλώσσα προγραμματισμού.

### **3.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ**

Στην πρώτη επανάληψη όσον αφορά την καταγραφή απαιτήσεων δεν έχουμε να κάνουμε τίποτα περισσότερο. Οι περιπτώσεις χρήσης έχουν καταγραφεί πλήρως κατά τις προηγούμενες επαναλήψεις. Από πλευράς ανάλυσης έχει κατασκευαστεί πλήρως το μοντέλο του πεδίου προβλήματος και έχουν κατασκευαστεί τα διαγράμματα καταστάσεων και δραστηριοτήτων. Το σύστημα έχει κατανοηθεί πλήρως οπότε και από την πλευρά της ανάλυσης δεν έχουμε τίποτα περαιτέρω να πραγματοποιήσουμε. Στην προηγούμενη επανάληψη κατασκευάσαμε το διάγραμμα κλάσεων του συστήματος. Οπότε σε αυτήν επανάληψη προχωράμε στην υλοποίηση των κλάσεων. Η επανάληψη θα περιλαμβάνει και εργασία ελέγχου, αφού κάθε κλάση που θα υλοποιούμε θα ενσωματώνεται στο σύστημα και δοκιμάζεται η σωστή λειτουργία του.

## **4 ΜΕΤΑΒΑΣΗ**

Μία επανάληψη είναι αρκετή για να ολοκληρωθεί η φάση της μετάβασης και να παραδοθεί το έργο.

### **4.1 ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ**

Σε αυτή την επανάληψη το σύστημα λογισμικού που έχουμε κατασκευάσει είναι μία δοκιμαστική έκδοση (beta release). Αυτήν την έκδοση θα την εγκαταστήσουμε σε ένα ή σε δύο ταμεία από τα πολλά που μπορεί να έχει το κατάστημα λιανικής πώλησης του παραδείγματός μας. Ταυτόχρονα με την εγκατάσταση του συστήματος λογισμικού θα γίνει και η εκπαίδευση των χρηστών/δοκιμαστών (beta users) στα συγκεκριμένα ταμεία. Θα δώσουμε τη δυνατότητα στους ταμίες να δοκιμάσουν το σύστημα για κάποιο χρονικό διάστημα ώστε να διαπιστώσουν ότι ικανοποιεί τις απαιτήσεις τους. Κατά την περίοδο αυτή θα καταγράφουμε όποιες τυχόν παρατηρήσεις (feedback) γίνονται από τους ταμίες και οι οποίες θα αποτελέσουν αντικείμενο βελτίωσης του λογισμικού σε αυτήν την επανάληψη. Οι όποιες αλλαγές προκύψουν πρέπει να διασφαλιστεί ότι δεν θα επηρεάσουν το υπόλοιπο σύστημα. Αφού το σύστημα λογισμικού που κατασκευάσαμε ανταποκρίνεται στις απαιτήσεις των χρηστών/δοκιμαστών, τότε θα εγκαταστήσουμε το σύστημα σε όλα τα υπόλοιπα ταμεία και θα παράσχουμε τα σχετικά εγχειρίδια χρήσης και εκπαιδευτικό υλικό για όλους τους ταμίες που θα κάνουν χρήση του συστήματος λογισμικού. Τέλος τα παραγόμενα που δημιουργήθηκαν κατά τη διαδικασία ανάπτυξης του συστήματος αποθηκεύονται και για χρήση σε κάποια πιθανή μελλοντική εξέλιξη του συστήματος.

## Δ. ΕΠΙΛΟΓΟΣ

Είναι βέβαιο λοιπόν ότι η UML αποτελεί απαραίτητο εφόδιο για τον επαγγελματία πληροφορικό – αναλυτή, σχεδιαστή, προγραμματιστή – ή οποιονδήποτε άλλον επιθυμεί να αναπτύξει λογισμικό. Επιπλέον, η χρησιμότητα της γλώσσας UML δεν περιορίζεται στο χώρο της πληροφορικής μια και η ίδια γλώσσα είναι σχεδιασμένη έτσι ώστε να είναι ανεξάρτητη από διαδικασίες ανάπτυξης και μπορεί να χρησιμοποιηθεί για τη μοντελοποίηση επιχειρηματικών διαδικασιών και συνεπώς η χρησιμότητά της επεκτείνεται και στον χώρο της διοίκησης επιχειρήσεων. Αυτή τη στιγμή η UML βρίσκεται στην έκδοση 2.1.1. Ένα CASE (Computer Aided Software Engineering) εργαλείο που επιτρέπει τη χρήση της UML με αυτοματοποιημένο τρόπο είναι το Visual Paradigm.

Μια διαδικασία ανάπτυξης λογισμικού περιγράφει την προσέγγιση με την οποία κατασκευάζεται, αναπτύσσεται και πιθανώς συντηρείται το λογισμικό. Η ενοποιημένη διεργασία έχει αναδυθεί ως μια δημοφιλής διαδικασία ανάπτυξης λογισμικού. Η ενοποιημένη διεργασία προωθεί πολλές πρακτικές, αλλά μία είναι η βασικότερη και αυτή είναι η επαναληπτική ανάπτυξη. Σε αυτήν την προσέγγιση η ανάπτυξη οργανώνεται σε ένα σύνολο σύντομων και συγκεκριμένων μίνι έργων που ονομάζονται επαναλήψεις. Το αποτέλεσμα της κάθε μίας επανάληψης είναι ένα δοκιμασμένο, ολοκληρωμένο και εκτελέσιμο σύστημα. Κάθε επανάληψη περιλαμβάνει τις δικές της δραστηριότητες της καταγραφής απαιτήσεων, ανάλυσης απαιτήσεων, σχεδίασης, υλοποίησης και δοκιμής. Ο επαναληπτικός κύκλος ζωής βασίζεται στη διαδοχική αύξηση και βελτίωση ενός συστήματος λογισμικού μέσα από μια σειρά επαναλήψεων, με κύριους οδηγούς την καταγραφή των απόψεων των χρηστών μέσα από δοκιμή κάθε επανάληψης και την απαραίτητη προσαρμογή που θα οδηγήσουν στο κατάλληλο σύστημα. Το σύστημα μεγαλώνει σταδιακά, επανάληψη με επανάληψη, και για αυτό αυτή η προσέγγιση είναι γνωστή και ως επαναληπτική και σταδιακή ανάπτυξη.

Κάθε επανάληψη περιλαμβάνει την επιλογή ενός μικρού υποσυνόλου απαιτήσεων και την ανάλυση, σχεδίαση, υλοποίηση και δοκιμή του. Στις πρώτες επαναλήψεις η επιλογή των απαιτήσεων και της σχεδίασης μπορεί να μην είναι η τελικώς επιθυμητή, αλλά η ενέργεια του να προχωρήσουμε σε περαιτέρω δραστηριότητες πριν ολοκληρωθεί η καταγραφή των απαιτήσεων ή σχεδιαστεί λεπτομερώς το σύστημα, οδηγεί στη γρήγορη ανάδραση (feedback) από τους χρήστες, τους κατασκευαστές και τους δοκιμαστές. Οι τελικοί χρήστες δηλαδή έχουν την ευκαιρία να δουν ένα μέρος του συστήματος και να εκφράσουν την άποψη τους. Αυτή η γρήγορη ανάδραση είναι μεγάλης αξίας. Έτσι αντί να κάνουμε υπολογισμούς πάνω σε υποθετικές σωστές απαιτήσεις ή σωστή σχεδίαση, με τη μέθοδο της ενοποιημένης διεργασίας έχουμε την ανάδραση από πραγματική κατασκευή και τη δυνατότητα να προσαρμόσουμε τις απαιτήσεις και τη σχεδίαση. Αυτή η αλλαγή είναι μοιραία και αναπόφευκτη, αλλά ταυτόχρονα, μέσα στο σύγχρονο ανταγωνιστικό περιβάλλον, θα αποτελέσει το συγκριτικό πλεονέκτημα για την επιχείρηση που θα κάνει χρήση του συγκεκριμένου συστήματος λογισμικού.

## ΓΛΩΣΣΑΡΙ

Activity Diagram	Διάγραμμα Δραστηριοτήτων
Actor	Χρήστης
Aggregation	Συναρμολόγηση
Architecture-Centric	Αρχιτεκτονικοκεντρική
Artifact	Παραγόμενο
Behavior Diagram	Διάγραμμα Συμπεριφοράς
Class Diagram	Διάγραμμα Κλάσεων
Class	Κλάση
Collaboration Diagram	Διάγραμμα Συνεργασίας
Component Diagram	Διάγραμμα Συστατικών
Component	Συστατικό
Composition	Σύνθεση
Construction	Κατασκευή
Deployment Diagram	Διάγραμμα Διάταξης
Domain Model	Μοντέλο του Πεδίου Προβλήματος
Elaboration	Λεπτομερής Επεξεργασία
Extend	Επέκταση
Extension Point	Σημείο Επέκτασης
Green Field	Καινούργιο Περιβάλλον
Implementation Diagram	Διάγραμμα Υλοποίησης
Inception	Σύλληψη
Include	Συμπερίληψη
Incremental	Σταδιακή
Interaction Diagram	Διάγραμμα Αλληλεπίδρασης
Interface	Διασύνδεση
Iterative	Επαναληπτική
Pattern	Κανόνας
Polymorphism	Πολυμορφισμός
Sequence Diagram	Διάγραμμα Ακολουθίας
State Diagram	Διάγραμμα Καταστάσεων
Transition	Μετάβαση
Unified Modeling Language	Ενοποιημένη Γλώσσα Μοντελοποίησης
Unified Process	Ενοποιημένη διεργασία
Use Case Diagram	Διάγραμμα Περιπτώσεων Χρήσης
Use Case	Περίπτωση Χρήσης
Use-Case Driven	Καθοδηγούμενη από τις Περιπτώσεις Χρήσης

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- Arlow, J. Neustadt, I. (2006). UML 2 And The Unified Process: Practical Object-Oriented Analysis And Design. 2nd Edition. Addison-Wesley
- Bergstrom, S. Raberg, L. (2003). Adopting The Rational Unified Process: Success With RUP. Addison-Wesley
- Booch, G. Jacobson, I. Rumbaugh, J. (1999). The Unified Software Development Process. Addison-Wesley
- Fowler, M. (1998). UML Distilled Applying The Standard Object Modeling Language. Addison Wesley Longman
- Fowler, M. (2003). UML Distilled A Brief Guide To the Standard Object Modeling Language. 3rd Edition. Addison-Wesley
- Kental, S. (2001). The Unified Process Explained. Addison-Wesley
- Kroll, P. Kruchten, P. (2003). The Rational Unified Process Made Easy: A Practitioner's Guide To The RUP. Addison-Wesley
- Larman, C (2005). Applying UML And Patterns: An Introduction To Object-Oriented Analysis And Design And The Unified Process. 3rd edition. NJ: Prentice Hall PTR
- Larman, C. (2002). Applying UML And Patterns, An Introduction To Object-Oriented Analysis And Design And The Unified Process. 2nd edition. NJ: Prentice Hall PTR
- Lethbridge, T. Laganier, R. (2005). Object Oriented Software Engineering. 2nd Edition. Mc Graw Hill
- Internet: < <http://www.omg.org> >
- Internet: < <http://www.swebok.org> >
- Internet: < <http://www.usecases.org> >