

---

ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΕΙΡΑΙΩΣ



ΤΜΗΜΑ  
ΨΗΦΙΑΚΩΝ  
ΣΥΣΤΗΜΑΤΩΝ

---

Ανάπτυξη και σχεδίαση  
Πληροφοριακού  
Συστήματος διαχείρισης  
αποθήκης σε εταιρεία  
παραγωγής παγοκύβων

**Κολάκας Δημήτριος (ME1906)**

ΠΜΣ «Πληροφοριακά Συστήματα & Υπηρεσίες»  
Ειδίκευση: «Προηγμένα Πληροφοριακά Συστήματα»

---

Πειραιάς, Φεβρουάριος 2022





*Η παρούσα εργασία διπλωματική εκτελείται στο πλαίσιο του ΠΜΣ «Πληροφοριακά Συστήματα & Υπηρεσίες» κατά το ακαδημαϊκό έτος 2022.*

*Copyright © 2022*

*Κολάκας Δημήτριος*



## **Περίληψη**

Ο σκοπός της παρούσας εργασίας είναι η υλοποίηση ενός Πληροφοριακού Συστήματος WMS για την διαχείριση της αποθήκης μιας εταιρίας παραγωγής παγοκύβων. Η όλη διεργασία που ακολουθείται για την επίτευξη του συγκεκριμένου στόχου θα περιγραφεί εκτενώς στα κεφάλαια που ακολουθούν.

Η ομάδα ανάπτυξης θα κάνει χρήση τεχνικών συστημικής και δομημένης ανάλυσης με σκοπό την ολοκληρωμένη προσέγγιση της προβληματικής κατάστασης, ενώ για την ανάπτυξη του Πληροφοριακού Συστήματος θα εφαρμοστεί η μεθοδολογία DevOps.

## **Abstract**

The purpose of the current paper is to implement a WMS Information System to manage the warehouse of ice cubes manufacturing company. The whole process followed to achieve this goal will be described in detail in the following chapters.

The development team will use systemic and structured analysis techniques in order to fully integrate the problematic situation, while agile system methodologies will be implemented. Finally, we have to note the fact the development of the Information System will be based on DevOps.

## Ευχαριστίες

Πρώτα από όλους θα ήθελα να ευχαριστήσω τους γονείς μου για την στήριξη που μου παρείχαν, την πίστη που μου έδειξαν και τις θυσίες που έκαναν για μένα όλα αυτά τα χρόνια. Στην συνέχεια θέλω να ευχαριστήσω τον καθηγητή μου κ. Βασιλακόπουλο Γεώργιο για την εμπιστοσύνη που έδειξε στο πρόσωπό μου με την ανάθεση της παρούσας πτυχιακής εργασίας και την βοήθειά του μέχρι την υλοποίησή της. Τέλος θέλω να ευχαριστήσω τους φίλους μου για όλη την στήριξη και την βοήθεια που μου πρόσφεραν μέχρι την ολοκλήρωση της πτυχιακή μου εργασίας.

*Κολάκας Δημήτριος*

# Περιεχόμενα

<b>Κεφάλαιο 1: Εισαγωγή.....</b>	<b>1</b>
Η επίδραση των Πληροφοριακών Συστημάτων στους οργανισμούς.....	3
Υπόθεση εργασίας.....	4
Ανάλυση Διεργασίας.....	5
Μέθοδοι ανάπτυξης του ΠΣ .....	6
<b>Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση .....</b>	<b>8</b>
Μεθοδολογία Μαλακών Συστημάτων.....	8
Διαγράμματα UML .....	9
Ανάλυση Λειτουργικών Σημείων .....	10
Τεχνικές χρονοπρογραμματισμού.....	11
Μεθοδολογία ανάπτυξης.....	12
<b>Κεφάλαιο 3: Συστημική Ανάλυση .....</b>	<b>14</b>
Αναγνώριση της Προβληματικής Κατάστασης.....	15
Επιλογή Θεμελιακού Ορισμού.....	16
Κατασκευή Εννοιολογικών Μοντέλων.....	17
Σύγκριση Μοντέλων με την Πραγματικότητα.....	20
Εντοπισμός και Υλοποίηση Αλλαγών.....	21
<b>Κεφάλαιο 4: Διαδικασία ανάπτυξης Πληροφοριακού Συστήματος .....</b>	<b>22</b>
Μεθοδολογία DevOps .....	23
Κύκλος Ζωής DevOps .....	25
Χρονοδιάγραμμα υλοποίησης.....	27
Διάγραμμα PERT .....	27
Διάγραμμα Gantt .....	29
Κοστολόγηση ΠΣ.....	30
Συστατικά συστήματος .....	30

Συντελεστής Μη Προσαρμοσμένων Λειτουργικών Σημείων – Unadjusted Functional



Points (UFP) .....	31
Συντελεστής Προσαρμογής – Value Adjustment Factor (VAF).....	34
Μέτρηση Λειτουργικών Σημείων.....	36
Υπολογισμός κόστους .....	38
<b>Κεφάλαιο 5: Σχεδιασμός Πληροφοριακού Συστήματος.....</b>	<b>38</b>
Προσδιορισμός απαιτήσεων .....	39
Είσοδος υπαλλήλου στο σύστημα .....	39
Επεξεργασία δεδομένων του συστήματος .....	40
Έξοδος υπαλλήλου από το σύστημα .....	41
Σχεδίαση ΠΣ με χρήση ενοποιημένης γλώσσας σχεδιασμού (Unified Modeling Language) (UML) .....	42
Διάγραμμα Δραστηριοτήτων (UML Activity Diagram).....	43
Διάγραμμα Ακολουθίας (Sequence Diagram) .....	44
Σχεδίαση Βάσης Δεδομένων.....	45
Μοντέλο οντοτήτων-συσχετίσεων .....	48
Σχεσιακό μοντέλο.....	50
<b>Κεφάλαιο 6: Εγχειρίδιο χρήστη (User manual).....</b>	<b>51</b>
Είσοδος χρήστη στο σύστημα.....	51
Κοινές Λειτουργίες .....	52
Υπάλληλος Παραγωγής.....	53
Υπάλληλος Αποθήκης.....	53
Καρτέλα «Products» .....	54
Υπάλληλος Πωλήσεων.....	55
Καρτέλα «Sels» .....	55
Καρτέλα «Clients».....	56
Υπεύθυνος Παραγωγής .....	58
Καρτέλα «Prediction».....	58
Καρτέλα «Store» .....	59

## Βιβλιογραφικές παραπομπές

### Κατάλογος σχημάτων πινάκων

Σχήμα 3.1 Η προσέγγιση των 7 σταδίων της SSM.....	14
Σχήμα 3.2 Γενικό Εννοιολογικό Μοντέλο .....	17
Σχήμα 3.3 Γνωστικό Μοντέλο.....	18
Σχήμα 3.4 Μοντέλο Απόκτησης και Διαχείρισης Πόρων.....	18
Σχήμα 3.5 Επιχειρησιακό Μοντέλο .....	19
Σχήμα 3.6 Μοντέλο Παρακολούθησης και Ελέγχου .....	19
Σχήμα 4.1 Διάγραμμα PERT .....	29
Σχήμα 4.2 Διάγραμμα Gantt .....	30
Σχήμα 5.1 Διάγραμμα Δραστηριοτήτων .....	44
Σχήμα 5.2 Διάγραμμα Ακολουθίας.....	45
Σχήμα 5.3 Μοντέλο οντοτήτων-συσχετίσεων .....	48
Σχήμα 5.4 Σχεσιακό μοντέλο .....	51

### Κατάλογος πινάκων

Πίνακας 4.1 Δεδομένα προγραμματισμού έργου για το διάγραμμα PERT .....	28
Πίνακας 4.2 Δεδομένα προγραμματισμού έργου για το διάγραμμα PERT .....	29

Πίνακας 4.3 Μέτρηση Λειτουργικών Σημείων Εισόδων.....	32
Πίνακας 4.4 Μέτρηση Λειτουργικών Σημείων Εξόδων .....	33
Πίνακας 4.5 Μέτρηση Λειτουργικών Σημείων Επερωτήσεων .....	33
Πίνακας 4.6 Μέτρηση Λειτουργικών Σημείων Εσωτερικών Λογικών Αρχείων .....	34
Πίνακας 4.7 Μέτρηση Λειτουργικών Σημείων Εξωτερικών Αρχείων Διεπαφής.....	34
Πίνακας 4.8 Κλίμακα επιρροής .....	35
Πίνακας 4.9 Γενικά χαρακτηριστικά του συστήματος .....	35
Πίνακας 4.10 Μέτρηση Μη Προσαρμοσμένων Λειτουργικών σημείων.....	37
Πίνακας 4.11 Υπολογισμός Συντελεστή προσαρμογής.....	37
Πίνακας 6.1 Κοινές Λειτουργίες ΠΣ.....	53

### **Κατάλογος Εικόνων**

Εικόνα 6.1 Είσοδος Χρήστη στο σύστημα.....	52
Εικόνα 6.2 Products .....	55
Εικόνα 6.3 Sales.....	57
Εικόνα 6.4 Clients.....	59
Εικόνα 6.5 Prediction.....	59
Εικόνα 6.6 Store.....	60





## Κεφάλαιο 1: Εισαγωγή

Τα τελευταία χρόνια αναπόσπαστο κομμάτι της ανθρώπινης καθημερινότητας σε όλο το φάσμα των δραστηριοτήτων της, αποτελούν τα πληροφορικά συστήματα. Από τη μόρφωση και την επικοινωνία, ως τη διασκέδαση και τις μεταφορές. Ο τομέας ο οποίος έχει αλλάξει πιο ριζοσπαστικά είναι ο τομέας της επιχειρηματικότητας. Δεν μπορεί να υπάρξει επιχείρηση στην σημερινή εποχή η οποία να είναι αποκομμένη από τον τεχνολογικό κόσμο. Η τεχνολογία βοηθάει ώστε σε όλα τα τμήματα μιας επιχείρησης, οι διεργασίες της να ολοκληρώνονται ταχύτερα, με ακρίβεια και αποτελεσματικότητα. Οι επιχειρήσεις που δεν χρησιμοποιούν τεχνολογικά μέσα ή που δεν τα εφαρμόζουν σωστά, δεν αντέχουν τον ανταγωνισμό της σύγχρονης αγοράς.

Σημασία δεν έχει απλά η χρήση τεχνολογικών μέσων. Σημασία έχει και η σωστή εφαρμογή της. Πέρα από την ανάπτυξη της ίδιας της τεχνολογίας τα τελευταία χρόνια έχει αναπτυχθεί και η θεωρία πίσω από την ανάπτυξη των πληροφοριακών συστημάτων. Τα πληροφοριακά συστήματα αναπτύσσονται με νέες τεχνοτροπίες και επικρατέστερες θεωρίες. Με αυτές τις τεχνοτροπίες έχουμε συστημική ανάπτυξη των πληροφοριακών συστημάτων με τέτοιο τρόπο ώστε να διευκολύνεται η ομάδα ανάπτυξής τους και να έχουμε ικανοποιητικότερο αποτέλεσμα για το χρήστη του πληροφοριακού συστήματος.

Είναι πολύ σημαντικό ένα πληροφοριακό σύστημα μεγάλης κλίμακας το οποίο θα υποστηρίζει τις λειτουργίες μιας επιχείρησης, να αναπτύσσεται από ομάδα ειδικών και έμπειρων τεχνικών οι οποίοι αφού είναι γνώστες των σωστών τεχνικών ανάπτυξης και αφού αφουγκραστούν και εκτιμήσουν τις ανάγκες αλλά και τις θεωρήσεις της ίδιας της επιχείρησης που θα χρησιμοποιήσει το υπό ανάπτυξη πληροφοριακό σύστημα, θα υλοποιήσει με τον βέλτιστο τρόπο την εφαρμογή.

Εκτός λοιπόν από τις απαραίτητες γνώσεις, ένα πληροφοριακό σύστημα για να κριθεί κατάλληλο είναι αναγκαίο να λειτουργεί με νοοτροπία η οποία συμερίζεται την ίδια νοοτροπία της επιχείρησης από την οποία θα χρησιμοποιηθεί. Αν η θεώρηση της επιχείρησης δεν είναι όμοια με αυτή του συστήματος, αυτό δεν θα υποστηρίζει αποτελεσματικά το στόχο του και κατ' επέκταση το στόχο της επιχείρησης.

Επίσης, ένα πληροφοριακό σύστημα, θα πρέπει να αφουγκράζεται τις ανάγκες και τις προσδοκίες της κοινωνίας στην οποία βρίσκεται η επιχείρηση που θα το χρησιμοποιήσει. Τα πληροφοριακά συστήματα δεν είναι αυτούσια και ανεξάρτητα εργαλεία. Διαμορφώνονται από περιβαλλοντικούς παράγοντες, όπως η παγκοσμιοποιημένη οικονομία, η ψηφιοποίηση



της αγοράς και της επιχειρηματικότητας, οι ραγδαίες εξελίξεις στον χώρο της τεχνολογίας, οι καινοτόμες τεχνοτροπίες αλλά και ο κοινωνικός και ανθρώπινος παράγοντας.

Η ομάδα ανάπτυξης του πληροφοριακού συστήματος θα πρέπει να λάβει υπ' όψιν όλες τις παραπάνω μεταβλητές και μετά από συνδιαλέξεις με την επιχείρηση για την οποία αναπτύσσεται το πληροφοριακό σύστημα να ξεκινήσει τη σχεδίαση του. Για παράδειγμα, αν η επιχείρηση έχει πελατοκεντρική θεώρηση, δηλαδή αν ο στόχος της είναι η ικανοποίηση των πελατών της μέσω των προϊόντων ή των υπηρεσιών της ή αν έχει λειτουργιοκεντρική θεώρηση, δηλαδή προσανατολίζει τους διαθέσιμους πόρους της στο σχεδιασμό, την ανάπτυξη, την επίβλεψη και τη βελτιστοποίηση των λειτουργιών της, αντίστοιχα θα πρέπει να σχεδιαστεί και το πληροφοριακό σύστημα.



## Η επίδραση των Πληροφοριακών Συστημάτων στις επιχειρήσεις

Ο βαθμός άρτιας λειτουργίας μιας επιχείρησης εξαρτάται από διάφορες παραμέτρους οι οποίες χαρακτηρίζουν την ίδια την επιχείρηση. Μερικές από αυτές τις παραμέτρους είναι η αποτελεσματικότητά της, δηλαδή το κατά πόσο η επιχείρηση πετυχαίνει τους στόχους τους οποίους η ίδια έχει θέσει, η αποδοτικότητά της, δηλαδή το μέτρο της ορθής χρήσης των πόρων της όπως ανθρώπινο δυναμικό και κεφάλαιο, η αξιοπιστία της και η ευελιξία της, δηλαδή το κατά πόσο η επιχείρηση ανταποκρίνεται στις αλλαγές του περιβάλλοντος της και στις ραγδαίες εξελίξεις της τεχνολογίας, της κοινωνίας και της οικονομίας.

Για όλες τις παραπάνω μεταβλητές είναι εμφανές πως η χρήση ενός Πληροφοριακού Συστήματος μπορεί να έχει ιδιαίτερα θετική επίδραση στην συνολική εικόνα της επιχείρησης. Αντίθετα η έλλειψη υποστήριξης της επιχείρησης από κάποιο πληροφοριακό σύστημα μπορεί να οδηγήσει σε διατάραξη της λειτουργίας της, μείωση του ποιοτικού της δείκτη και έλλειψη αποτελεσματικού ελέγχου των λειτουργιών της.

Επίσης στο σύγχρονο επιχειρησιακό περιβάλλον, είναι βέβαιο πως ο όγκος των πληροφοριών που έχει να διαχειριστεί μία επιχείρηση, είτε πρόκειται για μικρή τοπική επιχείρηση είτε για πολυεθνική, είναι τεράστιος. Γίνεται εύκολα αντιληπτό πως δεν γίνεται να διαχειριστεί σωστά και έγκαιρα οποιαδήποτε επιχείρηση αυτόν τον όγκο πληροφοριών τόσο σε διοικητικό επίπεδο όσο και σε βασικές επιχειρησιακές διεργασίες. Επομένως είναι επιτακτική πλέον η ανάγκη χρήσης κατάλληλου πληροφοριακού συστήματος για να ανταπεξέλθει η επιχείρηση στο σύγχρονο ανταγωνιστικό περιβάλλον.



## Υπόθεση εργασίας

Στο πλαίσιο της εργασίας αυτής ζητείται από την ομάδα ανάπτυξης του Πληροφοριακού Συστήματος να σχεδιάσει μία εφαρμογή για εταιρεία παραγωγής παγοκύβων στην οποία οι αρμόδιοι να μπορούν να βλέπουν τα αγαθά της επιχείρησης ώστε να γνωρίζουν πόσα έχουν αποθηκευμένα. Να μπορούν να αναζητήσουν και να μπορούν να τροποποιήσουν οποιαδήποτε στιγμή τα αγαθά της επιχείρησης. Επίσης να μπορούν να εισάγουν τις πωλήσεις των αγαθών. Τέλος να υπάρχει μία πρόβλεψη πωλήσεων.



## Ανάλυση Διεργασίας

Σε αυτή την ενότητα θα αναλυθούν περαιτέρω τα επιμέρους βήματα του Συστήματος Διαχείρισης Αποθήκης (WMS). Μετά από τη συνεργασία με το διευθυντή παραγωγής της εταιρίας, η εφαρμογή αναλύεται στα εξής βήματα:

1. Ο υπάλληλος παραγωγής όταν έχει έτοιμα τα αγαθά για αποθήκευση στην αποθήκη, τα προσθέτει στο σύστημα.
2. Ο υπάλληλος αποθήκης μπορεί να αλλάξει τα δεδομένα στο σύστημα αν αντιληφθεί ότι έχει αποθηκεύσει στο σύστημα λάθος είδος ή ποσότητα. Επίσης μπορεί να τα διαγράψει από το σύστημα τα σκάρτα αγαθά.
3. Ο υπάλληλος των πωλήσεων έχει τη δυνατότητα να προσθέτει νέους πελάτες στην εφαρμογή και βλέπει την ποσότητα των αγαθών που υπάρχουν στην αποθήκη και να ενημερώνει την εφαρμογή όταν πουλάει αγαθά.
4. Ο υπεύθυνος παραγωγής θα είναι σε θέση να προγραμματίζει την παραγωγή της επιχείρησης μέσα από την εφαρμογή, επειδή θα υπάρχει μια πρόβλεψη με βάση τις πωλήσεις του προηγούμενου χρονικού διαστήματος.

Βάση των βημάτων αυτών της διεργασίας θα σχεδιαστεί το Πληροφοριακό Σύστημα το οποίο θα στηρίζει την ψηφιακή λειτουργία της διαδικασίας. Ο τρόπος όμως, με τον οποίο σχεδιάστηκε και αναπτύχθηκε η εφαρμογή, διαδραματίζει κεντρικό ρόλο για την ποιότητα του τελικού αποτελέσματος. Όλες οι μεθοδολογίες που χρησιμοποιήθηκαν σε στα στάδια ανάλυσης σχεδίασης και ανάπτυξης, περιγράφονται αναλυτικά σε επόμενες ενότητες.

## Μέθοδοι ανάπτυξης του ΠΣ

Για την παρούσα εργασία, κατόπιν επικοινωνίας με τους εκπροσώπους της επιχείρησης, και μετά από κατανόηση του προβλήματος που καλείται να αντιμετωπίσει η εφαρμογή, αποφασίστηκαν από την ομάδα ανάπτυξης του Πληροφοριακού Συστήματος οι μέθοδοι με τις οποίες θα λειτουργήσει η ομάδα καθ' όλη τη διάρκεια ανάπτυξης της εφαρμογής.

Αποφασίστηκε λοιπόν πως αρχικά θα χρησιμοποιηθεί η Μεθοδολογία Μαλακών Συστημάτων (ΜΜΣ), έτσι ώστε να καταγραφούν τα πραγματικά προβλήματα που αντιμετωπίζει η επιχείρηση και να εδραιωθούν οι τρόποι με τους οποίους αυτά θα λυθούν. Με τη Μεθοδολογία Μαλακών Συστημάτων θα γίνει κατανοητό από την ομάδα ανάπτυξης του συστήματος ποιες ακριβώς είναι οι προσδοκίες της επιχείρησης και ποιος είναι ο κατάλληλος τρόπος ανάπτυξης της εφαρμογής έτσι ώστε ο στόχος της εφαρμογής να καλύπτει της ανάγκες της, καθώς θα λαμβάνει υπ' όψιν και την κοσμοθεώρηση του περιβάλλοντος στον οποίο δραστηριοποιείται.

Για την αποτελεσματική επικοινωνία της ομάδας ανάπτυξης με τους εκπροσώπους της επιχείρησης, χρησιμοποιείται η ενοποιημένη γλώσσα σχεδιασμού (Unified Modeling Language) (UML). Είναι καίριας σημασίας η κατανόηση των αναγκών της επιχείρησης, αλλά και η κατανόηση των λειτουργιών της εφαρμογής από τους εκπροσώπους. Πολλές φορές είναι δύσκολη η επικοινωνία μεταξύ των τεχνικών ανάπτυξης πληροφοριακών συστημάτων και των εκπροσώπων των επιχειρήσεων καθώς υπάρχει μεγάλο κενό ανάμεσα στην γλώσσα επικοινωνίας. Οι ομάδες ανάπτυξης χρησιμοποιούν ορολογία την οποία οι ίδιοι κατανοούν, όμως για κάποιο μέλος της επιχείρησης το οποίο δεν έχει ασχοληθεί με την ανάπτυξη πληροφοριακών συστημάτων, δεν γίνεται εύκολα αντιληπτή αυτή η ορολογία. Επομένως υπάρχει πιθανότητα να μην γίνονται εύκολα κατανοητοί οι τεχνικοί που έχουν αναλάβει την ανάπτυξη της εφαρμογής. Η γλώσσα UML επιτρέπει την αποτελεσματική επικοινωνία μεταξύ των συνεργαζόμενων μελών διότι παρουσιάζει συνοπτικά και ουσιαστικά τα στοιχεία που θέλει να παρουσιάσει η ομάδα ανάπτυξης χωρίς να εμπεριέχονται επιστημονικές ορολογίες και τεχνικοί όροι που χρησιμοποιούνται από την ομάδα.

Όταν θεμελιωθεί η μορφή του Πληροφοριακού Συστήματος που θα χρησιμοποιηθεί και αφού καθοριστεί ο βασικός του στόχος πρέπει να ορισθεί και το μέγεθός του. Η επιχείρηση που θα το χρησιμοποιήσει και η ομάδα ανάπτυξης που θα το υποστηρίξει, χρειάζεται να γνωρίζει τόσο τον τρόπο της ανάπτυξης του, όσο και τα βασικά χαρακτηριστικά του δηλαδή το χρόνο και το κόστος.



Προκειμένου να προσεγγίσουμε αυτές τις παραμέτρους χρησιμοποιούμε τη μέθοδο FPA (Function Points Analysis), την ανάλυση δηλαδή των λειτουργικών σημείων του πληροφοριακού συστήματος. Η FPA επειδή εστιάζει στη λειτουργικότητα της εφαρμογής και όχι στον απόλυτο όγκο της, έχει τη δυνατότητα να προσεγγίσει το μέγεθος της μέσω των λειτουργικών σημείων της. Μπορούμε να προσδιορίσουμε τις ανθρωπόωρες που απαιτούνται για την ανάπτυξη της εφαρμογής και κατ' επέκταση έχουμε τη δυνατότητα να ορίσουμε το χρόνο που απαιτείται προκειμένου να ολοκληρωθεί όπως επίσης και το κόστος πριν να ξεκινήσουν οι εργασίες για την ανάπτυξη. Όταν η επιχείρηση που θα χρησιμοποιήσει την εφαρμογή συμφωνήσει με το κόστος και το χρόνο, τότε μπορεί να ξεκινήσει η δημιουργία της, όπως έχει συμφωνηθεί.

Όμως και η ίδια η διαδικασία της ανάπτυξης του πληροφοριακού συστήματος δεν γίνεται άτυπα και τυχαία. Πρέπει να οριστεί μεθοδολογία και κύκλος ζωής και ανάπτυξης. Για τις ανάγκες της ανάπτυξης του παρόντος Πληροφοριακού Συστήματος αποφασίστηκε πως η αποτελεσματικότερη μεθοδολογία είναι αυτή του DevOps. Η βασική αρχή της μεθόδου του DevOps είναι πως εκπρόσωποι του οργανισμού είναι συνεχώς μαζί με την ομάδα ανάπτυξης του πληροφοριακού συστήματος έτσι ώστε να επιβλέπεται συνεχώς η διαδικασία, να συζητείται επιτόπου κάθε πτυχή της εφαρμογής που αναπτύσσεται, και να συμφωνούνται όλες οι λεπτομέρειες για το τελικό αποτέλεσμα.

Τέλος για την αποτύπωση του χρόνου που θα χρειαστεί σε κάθε βήμα της διαδικασίας ανάπτυξης του Πληροφοριακού Συστήματος, χρησιμοποιούνται διαγράμματα Gantt και PERT. Τα διαγράμματα αυτά δείχνουν αναλυτικά τον χρόνο που χρειάζεται κάθε βήμα ξεχωριστά για την περάτωση του και μπορούν να γίνουν εύκολα αντιληπτά από τους εκπροσώπους του οργανισμού για τον οποίο αναπτύσσεται η εφαρμογή.

## Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση

Για να γίνουν καλύτερα κατανοητές οι μέθοδοι οι οποίες χρησιμοποιούνται κατά τη διάρκεια του σχεδιασμού και της ανάπτυξης του Πληροφοριακού Συστήματος, κρίνεται αναγκαία η επιμέρους ανάλυσή τους. Για αυτόν τον λόγο στο παρόν κεφάλαιο θα αναλυθεί η βιβλιογραφία και η νοοτροπία των μεθόδων που αναφέρθηκαν στην προηγούμενη ενότητα.

### Μεθοδολογία Μαλακών Συστημάτων

Η Μεθοδολογία Μαλακών Συστημάτων (ΜΜΣ) (Soft Systems Methodology - SSM), αναπτύχθηκε τη δεκαετία του 1970 από τον Peter Checkland και τους συναδέλφους του στο πανεπιστήμιο του Lancaster στο τμήμα συστημάτων. Η μεθοδολογία αναπτύχθηκε μέσα από μία σχετική έρευνα πάνω σε πραγματικά συστήματα. Ακόμα και σήμερα η εφαρμογή της μεθόδου προσθέτει γνώσεις στην ήδη υπάρχουσα έρευνα και αναπτύσσεται όλο και περισσότερο. Η μεθοδολογία λαμβάνει υπ' όψιν τις κοσμοθεωρήσεις του οργανισμού που μελετά και τις αντιλήψεις των ανθρώπων της τοπικής κοινωνίας.

Η λέξη “μαλακά” συστήματα προέρχεται μεταφορικά από τις προβληματικές καταστάσεις τις οποίες καλείται να λύσει το πληροφοριακό σύστημα. Όταν μια επιχείρηση αντιλαμβάνεται την ύπαρξη προβλήματος σε κάποια διεργασία της, παρατηρώντας πως η διεργασία δεν είναι αποδοτική ή δεν επιφέρει τα αναμενόμενα αποτελέσματα, και δεν μπορεί να προσδιοριστεί ακριβώς ποιο είναι το κέντρο του προβλήματος, καθιστώντας αδύνατο τον ορισμό του προβλήματος και κατ' επέκταση δύσκολη την εύρεση της λύσης του, τότε αυτό το πρόβλημα ορίζεται ως μαλακό. Η μέθοδος μαλακών συστημάτων εξερευνά όλα τα πλαίσια της διεργασίας και βοηθάει στην σχεδίαση συστήματος το οποίο βελτιώνει συνολικά την απόδοση της διεργασίας χωρίς να εστιάζει σε ένα συγκεκριμένο, πιθανώς δυσλειτουργικό, κομμάτι της.

Η ΜΜΣ παρεμβαίνει σε περιπτώσεις που ακόμα και τα μέλη της επιχείρησης η οποία αντιμετωπίζει την προβληματική διεργασία, δεν συμφωνούν μεταξύ τους ως προς την θεώρηση του προβλήματος. Χρησιμοποιεί την έννοια του συστήματος ως διερευνητικό μέσο το οποίο είναι θεμελιωτής και καταλύτης ως προς την διευκόλυνση του διαλόγου μεταξύ των συνεργαζόμενων μελών.

Στην παρούσα εργασία, η ομάδα ανάπτυξης του πληροφοριακού συστήματος επέλεξε να χρησιμοποιήσει τη Μεθοδολογία Μαλακών Συστημάτων καθώς κρίθηκε ότι μέσω αυτής



μπορεί να «θεραπευτεί» αποτελεσματικότερα το πρόβλημα του της αποθήκης της επιχείρησης, το οποίο καλείται να λύσει η χρήση του Πληροφοριακού Συστήματος το οποίο θα κατασκευαστεί. Σε συνεδρίες που έγιναν με τους εκπροσώπους της επιχείρησης, καταγράφηκε το πρόβλημα που αντιμετωπίζεται και η επιθυμητή λύση που συμφωνήθηκε. Λαμβάνοντας υπ' όψη τις βλέψεις της επιχείρησης προχώρησε η διαδικασία του σχεδιασμού και ανάπτυξης της εφαρμογής. Η αναλυτική εφαρμογή της μεθόδου μαλακών συστημάτων όπως αυτή χρησιμοποιήθηκε και εφαρμόστηκε για την εύρεση των λύσεων, καταγράφεται και αναλύεται σε επόμενη ενότητα.

## Διαγράμματα UML

Τα διαγράμματα UML εξελίχθηκαν από τους Grady Booch, James Rumbaugh, Ivar Jacobson και τον οργανισμό Rational Software Corporation, και χρησιμοποιούνται για την αποτελεσματική απεικόνιση ενός Πληροφοριακού Συστήματος.

Τα διαγράμματα UML είναι υψηλής σημασίας για την ανάπτυξη Πληροφοριακών Συστημάτων καθώς επιτρέπουν με τρόπο ιδιαίτερα απλό και εύχρηστο να γίνεται κατανοητή μία λειτουργία η οποία έχει σχεδιαστεί με επιστημονικό τρόπο από άτομα τα οποία δεν κατανοούν την ορολογία και τις τεχνικές που χρησιμοποιούνται, θεμελιώνοντας έτσι την αποτελεσματική επικοινωνία μεταξύ των μελών της συνεργασίας. Πέρα όμως από την κατανόηση των λειτουργιών του πληροφοριακού συστήματος από μη εξοικειωμένα με την μεθοδολογία άτομα, η χρήση γλώσσας και διαγραμμάτων UML διευκολύνει και την επικοινωνία μεταξύ των ίδιων των συντελεστών της ομάδας που αναπτύσσει το πληροφοριακό σύστημα καθώς με ένα πολύ απλό και κατανοητό διάγραμμα μεταδίδει πληροφορίες οι οποίες με λέξεις χρειάζονται πολύ περισσότερο κόπο να μεταδοθούν.

Τα διαγράμματα UML χρησιμοποιούνται σε διάφορα βήματα της διαδικασίας ανάπτυξης μίας εφαρμογής σε διάφορες μορφές τους. Για παράδειγμα στην διαδικασία ανάλυσης των απαιτήσεων του πληροφοριακού συστήματος χρησιμοποιούνται διαγράμματα περιπτώσεων και διαγράμματα κλάσεων, κατά τον σχεδιασμό του συστήματος χρησιμοποιούνται διαγράμματα δραστηριοτήτων και διαγράμματα ακολουθίας και κατά την υλοποίηση της εφαρμογής χρησιμοποιούνται διαγράμματα αντικειμένων και διαγράμματα καταστάσεων.



## Ανάλυση Λειτουργικών Σημείων

Μετά την εφαρμογή της Μεθοδολογίας Μαλακών Συστημάτων, έχει σχεδιαστεί μια πρώτη προσέγγιση του πληροφοριακού συστήματος το οποίο θα κατασκευαστεί. Βάση αυτής της προσέγγισης πρέπει να υπολογιστούν ο χρόνος υλοποίησης, το μέγεθος και το κόστος του Πληροφοριακού Συστήματος το οποίο θα κατασκευαστεί, με σκοπό να επέλθει η οριστική συμφωνία με την επιχείρηση ως προς το έργο. Για να γίνει αυτή μέτρηση χρησιμοποιήθηκε η μέθοδος της Ανάλυσης Λειτουργικών Σημείων (Function Point Analysis -FPA).

Η ανάλυση λειτουργικών σημείων αναπτύχθηκε την δεκαετία του 1970 από τον Allan Albrecht στην IBM, στα πλαίσια της προσπάθειας να μετρηθεί με αποτελεσματικό τρόπο το μέγεθος και η λειτουργικότητα ενός προγράμματος. Η νοοτροπία πίσω από την μέθοδο είναι το σπάσιμο του πληροφοριακού συστήματος σε μικρότερα κομμάτια με σκοπό την εύκολη ανάλυση και κατανόηση του.

Η Function Point Analysis είναι μια πρωτοποριακή μέθοδος για την εποχή που αναπτύχθηκε, και είναι αξιοσημείωτο ότι σε έναν χώρο σαν αυτόν της τεχνολογίας, που τα πράγματα εξελίσσονται τόσο γρήγορα και μεταβάλλονται κάθε μέρα, η μέθοδος αυτή έχει ακόμα εφαρμογή από τις σύγχρονες επιχειρήσεις. Τον καιρό που πρωτοεμφανίστηκε η Function Point Analysis, ήταν μια πραγματικά πρωτοποριακή μέθοδος η οποία συνέβαλλε τα μέγιστα στον χώρο της μέτρησης των πληροφοριακών συστημάτων καθώς εισήγαγε την ιδέα της μέτρησης της λειτουργικότητας ενός συστήματος και όχι απλώς του χρόνου ανάπτυξης του ή του μεγέθους του κώδικα.

Για την ανάπτυξη του παρόντος πληροφοριακού συστήματος αποφασίστηκε από την ομάδα ανάπτυξης ότι η καταλληλότερη μέθοδος για τον υπολογισμό του προσεγγιστικού μεγέθους της εφαρμογής είναι η μέθοδος της ανάλυσης λειτουργικών σημείων. Αυτό αποφασίστηκε με γνώμονα το γεγονός ότι η εφαρμογή που θα αναπτυχθεί θα είναι μεσαίου μεγέθους και επομένως η FPA θα επιφέρει ακριβή αποτελέσματα για τον υπολογισμό των λειτουργικών σημείων του πληροφοριακού συστήματος που θα αναπτυχθεί. Με τα αποτελέσματα τα οποία προέκυψαν από την ανάλυση λειτουργικών σημείων του σχεδιασμένου πληροφοριακού συστήματος, συζητήθηκε ο χρόνος περάτωσης του έργου και το κόστος του και συμφωνήθηκε το έργο με την επιχείρηση.



## Τεχνικές χρονοπρογραμματισμού

Η διαδικασία του χρονοπρογραμματισμού είναι μια διαδικασία ιδιαίτερος σημαντική για τη διοίκηση του έργου. Δυο από τις πλέον συνηθισμένες τεχνικές είναι τα διαγράμματα PERT και Gantt.

- Τα διαγράμματα PERT, στοχεύουν στην απεικόνιση των χρονικών πλαισίων ανάπτυξης των επιμέρους εργασιών ενός έργου κατά την διαδικασία του προγραμματισμού του. Σύμφωνα με αυτό το διάγραμμα, τα έργα μπορούν να οργανωθούν κατά γεγονότα και επιμέρους διεργασίες.
- Τα διαγράμματα Gantt συνήθως χρησιμοποιούνται για την κατασκευή του χρονοδιαγράμματος ενός έργου και για την αξιολόγηση της προόδου του. Τα διαγράμματα αυτά είναι απλούστερα στη χρήση και την κατασκευή τους καθώς αποτελούνται από απλά ευθύγραμμα τμήματα.

Συνήθως για την παρουσίαση του χρονοδιαγράμματος ενός έργου χρησιμοποιούνται και τα δύο διαγράμματα από τη διοίκηση του έργου. Ωστόσο τα διαγράμματα Pert χρησιμοποιούνται περισσότερο για την παρουσίαση χρονοδιαγραμμάτων έργων μεγάλης κλίμακας τα οποία παρουσιάζουν μεγάλη αλληλεξάρτηση στις επιμέρους εργασίες τους. Τα διαγράμματα Gantt λόγω της μειωμένης περιπλοκότητας τους, χρησιμοποιούνται περισσότερο σε έργα μικρότερης κλίμακας. Παρόλα αυτά τα διαγράμματα Gantt και Pert χρησιμοποιούνται μόνο συμπληρωματικά και μόνο για απεικόνιση των χρονοδιαγραμμάτων, όχι για την σχεδίαση τους. Στην παρούσα εργασία για την απεικόνιση του υπολογισμένου χρονοδιαγράμματος για το έργο της ανάπτυξης του Πληροφοριακού Συστήματος για την επιχείρηση, χρησιμοποιήθηκαν και τα δύο διαγράμματα.

## Μεθοδολογία DevOps

Πολλές επιχειρήσεις τα τελευταία χρόνια έχουν βιώσει τα οφέλη από τη χρήση μεθόδων Agile με την οποία έχουμε γρήγορη παράδοση του λογισμικού, βελτίωση της ποιότητας και μείωση του κόστους. Όμως η ανάπτυξη της Agile έρχεται σε σύγκρουση με την παραδοσιακή διαχείριση υπηρεσιών που έχουν σχέση με τη διαχείριση πληροφοριών, εφαρμογών και υποδομής. Το χάσμα ανάμεσα στην ανάπτυξη και διαχείριση των υπηρεσιών είναι πολύ μεγάλο. Η παραδοσιακή οργάνωση διαχείρισης υπηρεσιών που βασίζεται στον έλεγχο δεν συμβαδίζει με την ευέλικτη κουλτούρα ταχύτητας, επικοινωνίας και συνεργασίας. Αυτό σημαίνει ότι οι ομάδες ανάπτυξης μπορούν να παραδώσουν λογισμικό γρήγορα, αλλά αυτό το λογισμικό δεν μπορεί να ενταχθεί γρήγορα στην παραγωγή. Αυτό είναι ένα πρόβλημα που μπορεί να λυθεί μόνο με μια θεμελιώδη αλλαγή στον τρόπο συνεργασίας.

Η ένωση ανάπτυξης (Development) και λειτουργιών (Operations) θα είχε τη δυνατότητα να λύσει το πρόβλημα. Το DevOps είναι η ένωση ανθρώπων, διεργασιών και τεχνολογίας. Οι γνώσεις και οι δεξιότητες συνδυάζονται με τις μεθόδους εργασίας. Το DevOps δίνει τη δυνατότητα συντονισμού και συνεργασίας σε ρόλους, όπως ανάπτυξη, λειτουργίες πληροφορικής, μηχανική ποιότητας και ασφάλεια, για την παραγωγή καλύτερων και πιο αξιόπιστων προϊόντων. Υιοθετώντας μια κουλτούρα DevOps μαζί με πρακτικές και εργαλεία DevOps οι ομάδες μπορούν να ανταποκρίνονται καλύτερα στις ανάγκες των πελατών, να αυξάνουν την εμπιστοσύνη στις εφαρμογές που δημιουργούν με αποτέλεσμα να επιτυγχάνουν γρηγορότερα τους επιχειρηματικούς στόχους.

Στη φάση του σχεδίου οι ομάδες DevOps σχεδιάζουν με ευελιξία και ορατότητα. Ορίζουν και περιγράφουν δυνατότητες και χαρακτηριστικά των εφαρμογών και των συστημάτων που δημιουργούν. Παρακολουθούν την πρόοδο σε χαμηλά και υψηλά επίπεδα ευαισθησίας, από εργασίες ενός προϊόντος έως πολλών προϊόντων. Διαπιστώνουν τα σφάλματα, διαχειρίζονται ευέλικτα την ανάπτυξη του λογισμικού με το Scrum ή τη χρήση πινάκων Kanban, οπτικοποιούν την πρόοδο με τους πίνακες εργαλείων.

Στη φάση της ανάπτυξης περιλαμβάνεται η κωδικοποίηση (γραφή, δοκιμή αναθεώρηση και ενσωμάτωση κώδικα από τα μέλη της ομάδας) και η δημιουργία αυτού του κώδικα σε κατασκευάσματα που μπορούν να αναπτυχθούν σε διάφορα περιβάλλοντα. Οι ομάδες DevOps χρησιμοποιώντας πολύ καλά παραγωγικά εργαλεία, αυτοματοποιούν τα χειροκίνητα βήματα κάνουν συνεχείς επαναλήψεις και δοκιμές και επιδιώκουν να καινοτομούν χωρίς να θυσιάζουν την ποιότητα, τη σταθερότητα και την παραγωγικότητα.





Στη φάση της παράδοσης περιλαμβάνεται η διαδικασία ανάπτυξης των εφαρμογών με συνεπή και αξιόπιστο τρόπο, στο χώρο της παραγωγής. Από τις ομάδες ορίζεται μια διαδικασία διαχείρισης έκδοσης με ξεκάθαρα στάδια μη αυτόματης έγκρισης. Ορίζονται επίσης αυτοματοποιημένες πύλες όπου μετακινούνται οι εφαρμογές ανάμεσα στα στάδια μέχρι να γίνουν διαθέσιμες στους πελάτες. Οι διαδικασίες αυτές μέσω της αυτοματοποίησης γίνονται επεκτάσιμες, επαναλαμβανόμενες και ελεγχόμενες.

Στη φάση της λειτουργίας περιλαμβάνεται η συντήρηση, η παρακολούθηση και η αντιμετώπιση προβλημάτων των εφαρμογών σε περιβάλλον παραγωγής. Με τις πρακτικές DevOps, οι ομάδες εργάζονται για τη διασφάλιση της αξιοπιστίας του συστήματος, την υψηλή διαθεσιμότητα με στόχο το μηδενικό χρόνο διακοπής της λειτουργίας, ενισχύοντας την ασφάλεια και τη διακυβέρνηση. Οι ομάδες εντοπίζουν οτιδήποτε μπορεί να προκαλέσει κάποιο ζήτημα και είναι έτοιμες να μετριάσουν τα προβλήματα όταν αυτά προκύψουν. Υπάρχει μια επαγρύπνηση η οποία απαιτεί πλούσια τηλεμετρία, ενεργή ειδοποίηση και πλήρη ορατότητα στις εφαρμογές και στο υποκείμενο σύστημα.

Με την υιοθέτηση των πρακτικών DevOps αυτοματοποιούνται και βελτιστοποιούνται οι διαδικασίες μέσω της τεχνολογίας. Όμως όλα ξεκινούν από την κουλτούρα και τα άτομα. Η καλλιέργεια μια κουλτούρας DevOps απαιτεί πολύ βαθιές αλλαγές στο τρόπο εργασίας και συνεργασίας των ανθρώπων.

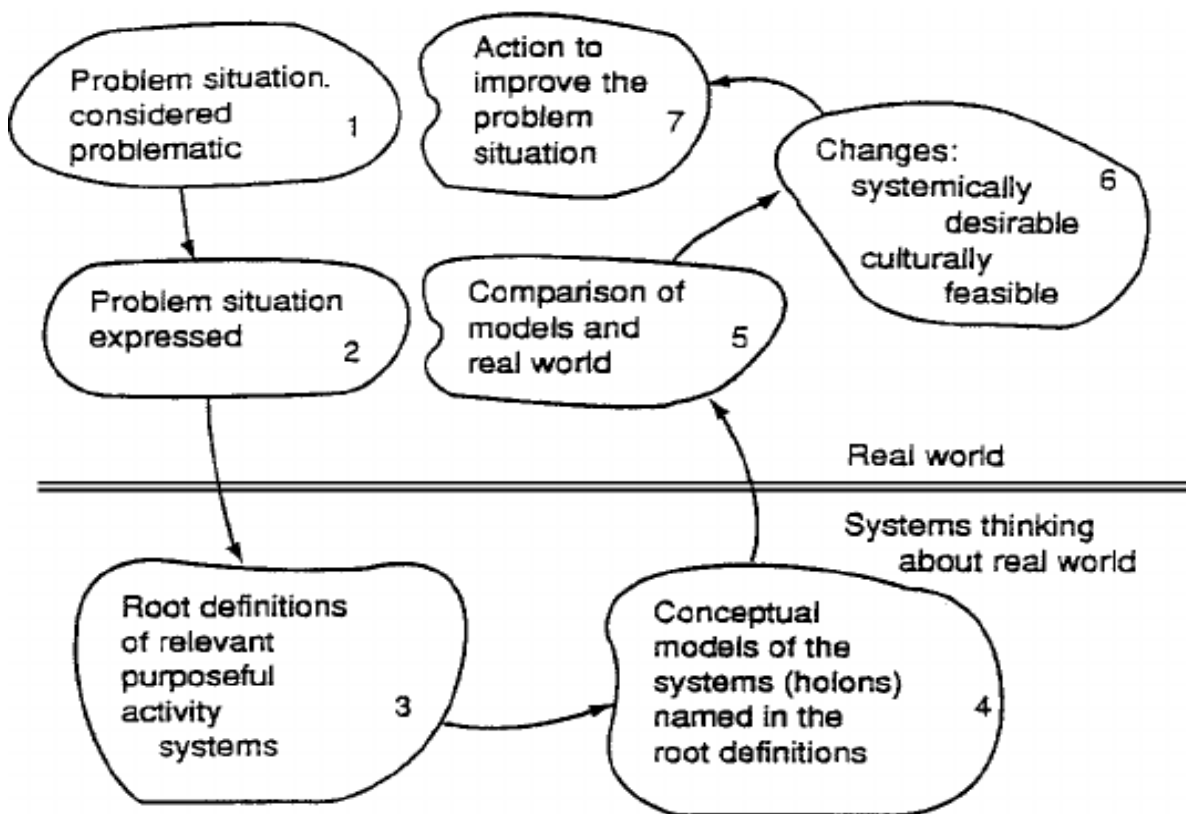
Η συνεργασία μεταξύ των ομάδων είναι χαρακτηριστικό γνώρισμα υγιούς κουλτούρας DevOps. Διαφορετικές ομάδες, όπως οι λειτουργίες ανάπτυξης και πληροφορικής, θα πρέπει να μοιράζονται μεταξύ τους τις διαδικασίες DevOps, τις προτεραιότητες και τις ανησυχίες τους. Πρέπει να σχεδιάζουν τη συνεργασία τους και να ευθυγραμμίζονται με τους στόχους της επιχείρησης. Καθώς οι ομάδες ευθυγραμμίζονται, αναλαμβάνουν ευθύνες και εμπλέκονται σε πρόσθετες φάσεις και όχι μόνο σε αυτές που είναι κεντρικές για τους ρόλους τους. Οι προγραμματιστές, υπεύθυνοι για την ποιότητα και καινοτομία στη φάση ανάπτυξης, γίνονται υπεύθυνοι και για την απόδοση και σταθερότητα που επιφέρουν οι αλλαγές στη φάση της λειτουργίας. Συγχρόνως, οι φορείς εκμετάλλευσης πληροφορικής θα συμπεριλάβουν στη φάση του σχεδίου και της ανάπτυξης, τη διακυβέρνηση, την ασφάλεια και τη συμμόρφωση.

Οι ομάδες DevOps δημιουργούν μια νοοτροπία ανάπτυξης ενσωματώνοντας την εκμάθηση στις διαδικασίες τους και επιτυγχάνουν τη συνεχή βελτίωση αυξάνοντας με αυτό τον τρόπο την ικανοποίηση των πελατών.

## Κεφάλαιο 3: Συστημική Ανάλυση

Σε όλες τις επιχειρήσεις υπάρχει μία σειρά προβλημάτων που καλείται να επιλυθεί. Προβλήματα που τα στελέχη τους χρειάζονται να επικεντρωθούν μόνο στις λύσεις (πως θα επιτευχθεί η αλλαγή;) και άλλα, αρκετά σύνθετα, που υπάρχει αδυναμία ακόμη και στον ορισμό του προβλήματος (τι πρέπει να αλλάξει;). Τα πρώτα είθισται να τα αναφέρουμε ως σκληρά (ή δομημένα) προβλήματα και τα πιο σύνθετα ως μαλακά (ή αδόμητα).

Στόχος σε αυτό το κεφάλαιο είναι να περιγράψουμε μία από τις πιο αποτελεσματικές μεθοδολογίες για την αντιμετώπιση των μαλακών προβλημάτων, γνωστή ως Μεθοδολογία Μαλακών Συστημάτων ή Soft Systems Methodology (SSM). Κύρια χαρακτηριστικά της είναι η ρεαλιστική προσέγγιση και ο σφαιρικός τρόπος που αντιμετωπίζει τα προβλήματα, αποτελούμενη συνολικά από 7 στάδια εφαρμογής, όπως αυτά απεικονίζονται στο **Σχήμα 3.1**.



Σχήμα 3.1: Η προσέγγιση των 7 σταδίων της SSM

(Πηγή: [https://www.researchgate.net/figure/The-seven-stage-approach\\_fig2\\_225960401](https://www.researchgate.net/figure/The-seven-stage-approach_fig2_225960401))



## Αναγνώριση της Προβληματικής Κατάστασης

Σε αυτή τη φάση στόχος είναι η ανάλυση της κατάστασης που γίνεται αντιληπτό το πρόβλημα, καταγράφοντας όσο το δυνατό περισσότερα στοιχεία για τον τρόπο αποθήκευσης, επεξεργασίας και πώλησης των προϊόντων της εταιρείας.

Στη συγκεκριμένη εταιρεία ό,τι αγαθά αποθηκεύονται στην αποθήκη σημειώνονται χειρόγραφα σε συγκεκριμένα έγγραφα. Επίσης οποιαδήποτε αλλαγή πραγματοποιείται, καταγράφεται χειρόγραφα. Στο τέλος της βάρδιας καταχωρούνται όλα τα χειρόγραφα σε φύλλο excel.

Καλούμαστε να βελτιώσουμε την λειτουργία του συστήματος των αποθηκών της εταιρείας.

Ο τρόπος αντιμετώπισης ενός τέτοιου προβλήματος είναι η δημιουργία μίας εφαρμογής στην οποία θα έχουν πρόσβαση συγκεκριμένοι υπάλληλοι οι οποίοι θα έχουν τη δυνατότητα να επεξεργάζονται τις πληροφορίες.

Για να γίνει αυτό αποφασίσαμε να δημιουργήσουμε ένα σύστημα διαχείρισης αποθήκης(WMS).

## Επιλογή Θεμελιακού Ορισμού

Μετά την έκφραση της κατάστασης με μία «πλούσια εικόνα», ακολουθεί η διατύπωση ενός θεμελιακού ορισμού, μίας εξιδανικευμένης περιγραφής του συστήματος. Κριτήριο αξιολόγησης του δεν είναι μόνο η χρησιμότητα του για την πληρέστερη κατανόηση της προβληματικής κατάστασης και την επίλυση της, αλλά και το πόσο ριζοσπαστικός είναι. Σε αυτό το στάδιο, πρέπει να τονιστεί ότι δεν γίνεται καταγραφή πραγματικών δεδομένων, αλλά προσπαθούμε να ορίσουμε το ιδανικό σύστημα.

Ο ορισμός αυτός περιλαμβάνει συνήθως 6 έννοιες (CATWOE):

- Customers, δηλαδή τα άτομα που επηρεάζονται από τις δραστηριότητες του συστήματος. Στο θέμα που πραγματεύεται η εργασία αυτή, είναι ο υπάλληλος της εταιρείας που επεξεργάζεται τα δεδομένα.
- Actors, δηλαδή τα άτομα που εκτελούν τις κύριες δραστηριότητες της εφαρμογής. Στην περίπτωση μας δρώντες είναι ο υπάλληλος της αποθήκης, ο υπάλληλος πωλήσεων, ο υπάλληλος παραγωγής, ο υπεύθυνος παραγωγής.
- Transformation, δηλαδή ο μετασχηματισμός του συστήματος που μετατρέπει τις εισροές (δεδομένα των πωλήσεων) σε εκροές (πρόβλεψη πωλήσεων).
- Worldview, ή αλλιώς κοσμοθεώρηση, βάσει της οποίας η επιχείρηση παρέχει υψηλής ποιότητας υπηρεσίες στους πελάτες της, ανταποκρινόμενη στις ανάγκες τους για υψηλή και αξιόπιστη παραγωγή των προϊόντων της, αναζητώντας νέες τεχνολογικές τάσεις.
- Ownership, ή ιδιοκτησία, στην περίπτωση μας η επιχείρηση.
- Environment, δηλαδή οι περιορισμοί του περιβάλλοντος, περιλαμβάνοντας όλους τους σχετικούς Νόμους που επιβάλλονται από την Ευρωπαϊκή Ένωση και το κράτος για την λειτουργία της επιχείρησης.

Βάσει όλων των προαναφερθέντων στοιχείων, θα μπορούσε να δοθεί ο εξής ορισμός:

Η εφαρμογή διαχείρισης της αποθήκης της εταιρείας, έχει σκοπός της οποίας είναι η διαχείριση των αποθεμάτων της αποθήκης, χρησιμοποιώντας ψηφιακά μέσα, υιοθετώντας βέλτιστες πρακτικές και αναζητώντας νέες τεχνολογικές τάσεις έτσι ώστε να ανταποκριθεί

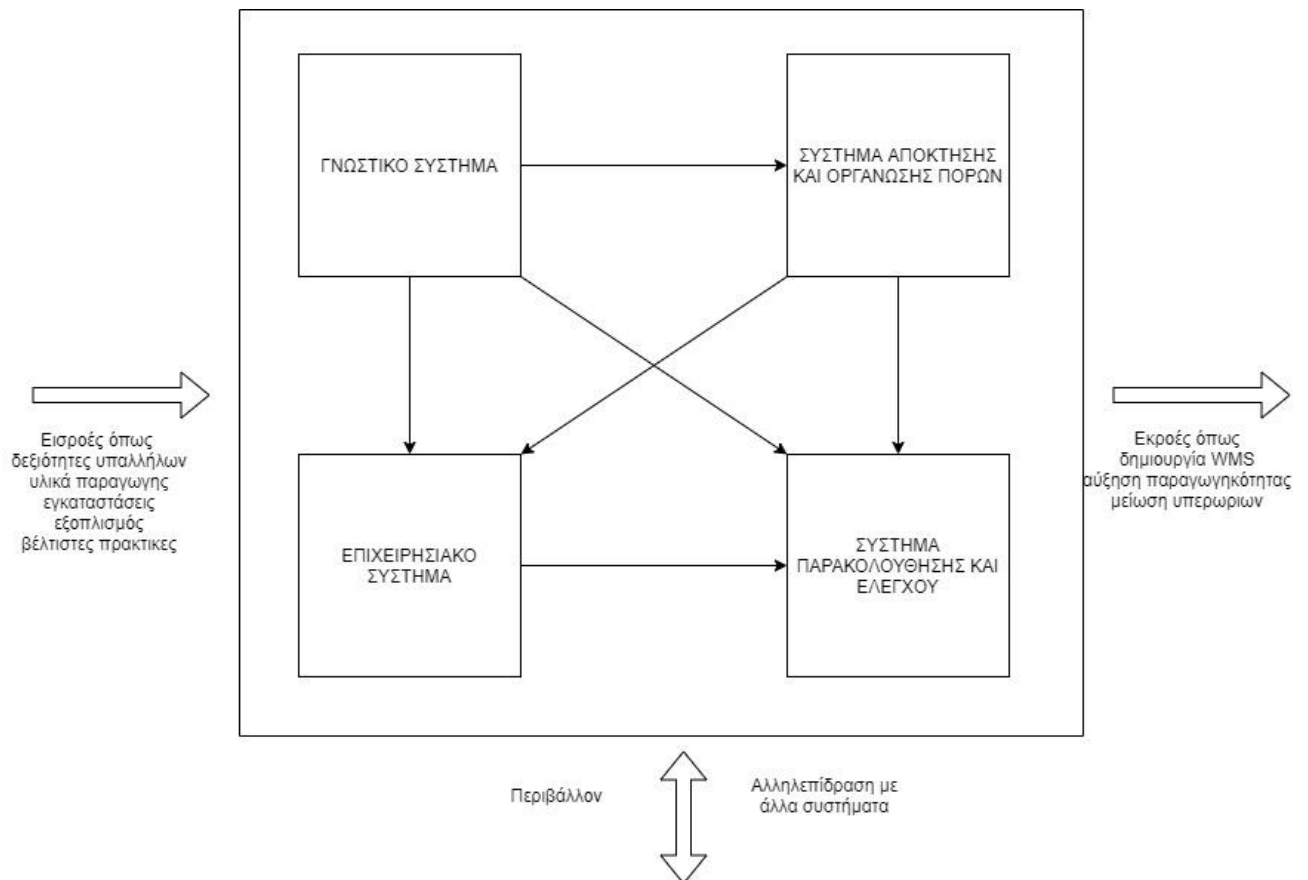
τόσο στις απαιτήσεις και στον ανταγωνισμό της ελληνικής και διεθνούς αγοράς, όσο και στις επιθυμίες των πελατών της.

## Κατασκευή Εννοιολογικών Μοντέλων

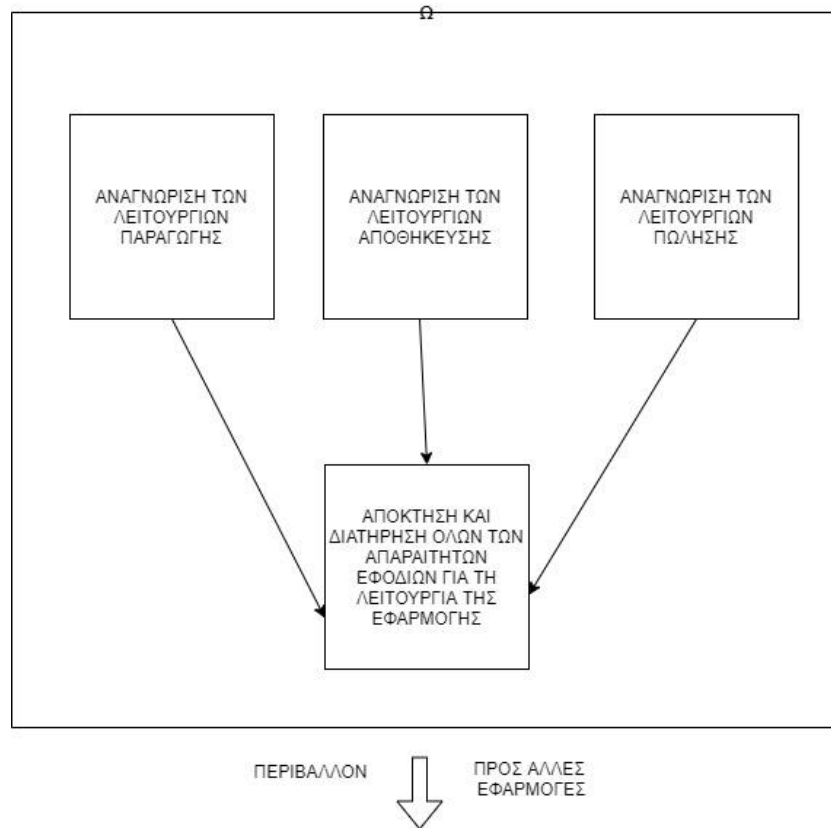
Στη συνέχεια, στόχος μας είναι να εκφράσουμε τον θεμελιακό ορισμό μέσω της ανάπτυξης εννοιολογικών μοντέλων που καταγράφουν τις λειτουργίες που πρέπει να εκτελεί το σύστημα.

Ο μεγαλύτερος κίνδυνος που εμφανίζεται σε αυτό το βήμα είναι να επηρεαστεί ο αναλυτής από την πραγματικότητα της κατάστασης. Για την επιτυχή εφαρμογή της μεθοδολογίας καλείται να μείνει ανεπηρέαστος κατασκευάζοντας εννοιολογικά μοντέλα που ανταποκρίνονται στην ιδανική εικόνα του συστήματος και όχι σε εκείνη που αναδύεται από την προβληματική κατάσταση.

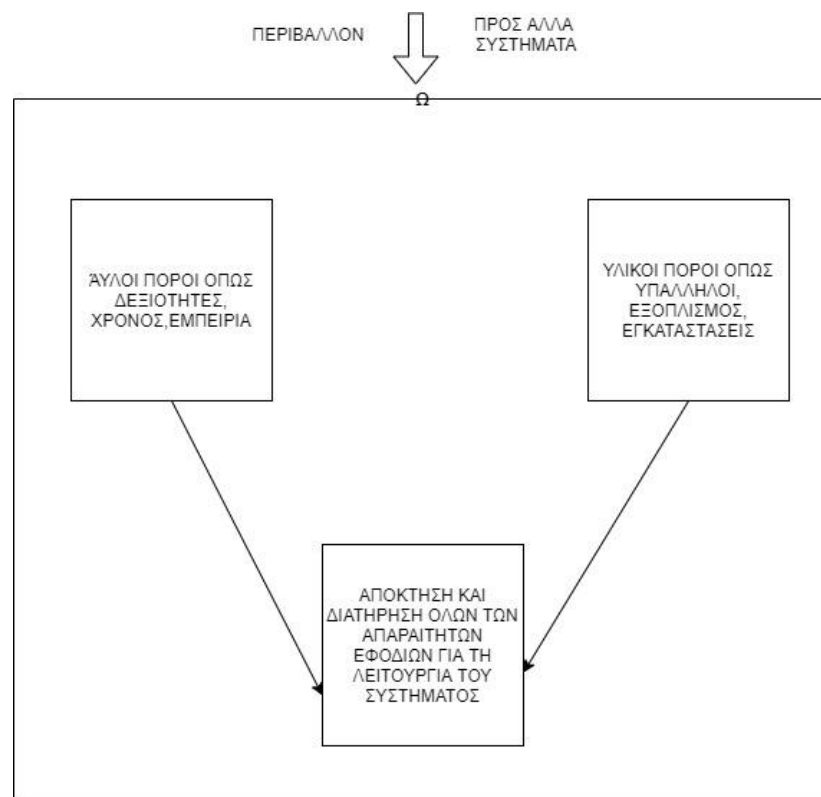
Ακολουθεί ένα γενικό εννοιολογικό μοντέλο της υπηρεσίας αναδιάρθρωσης ενός χαρτοφυλακίου (Σχήμα 3.2) καθώς και μοντέλα για τα επιμέρους συστήματα αυτού, όπως το γνωστικό (Σχήμα 3.3), το μοντέλο απόκτησης και διαχείρισης πόρων (Σχήμα 3.4), το επιχειρησιακό (Σχήμα 3.5) και τέλος, το μοντέλο παρακολούθησης και ελέγχου (Σχήμα 3.6).



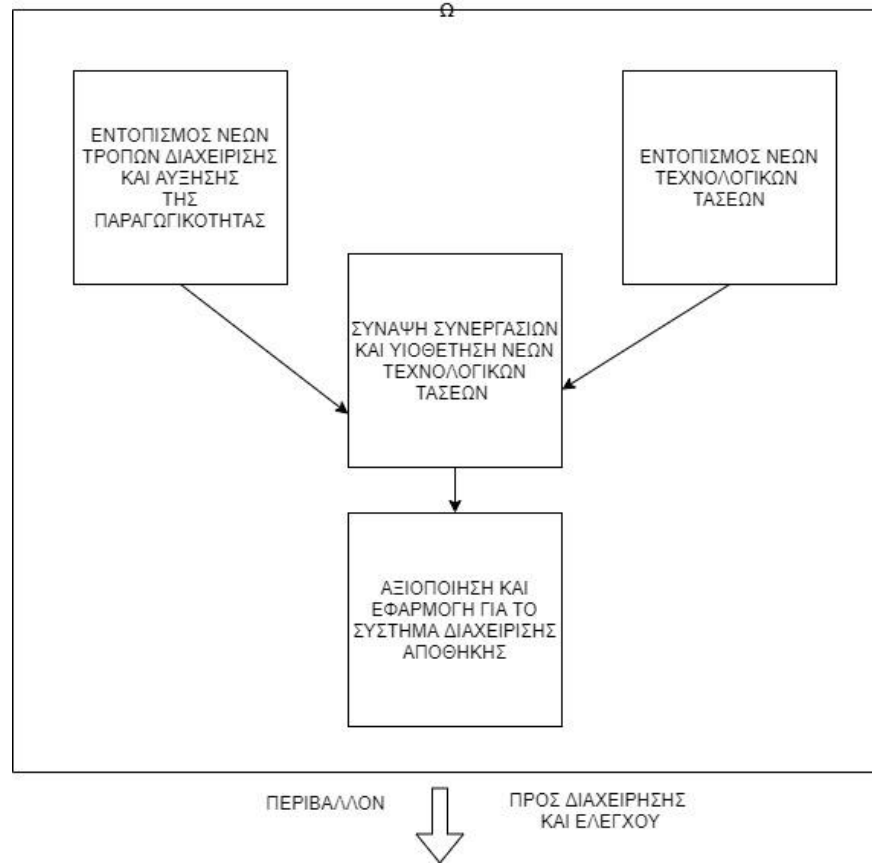
Σχήμα 3.2: Γενικό Εννοιολογικό Μοντέλο



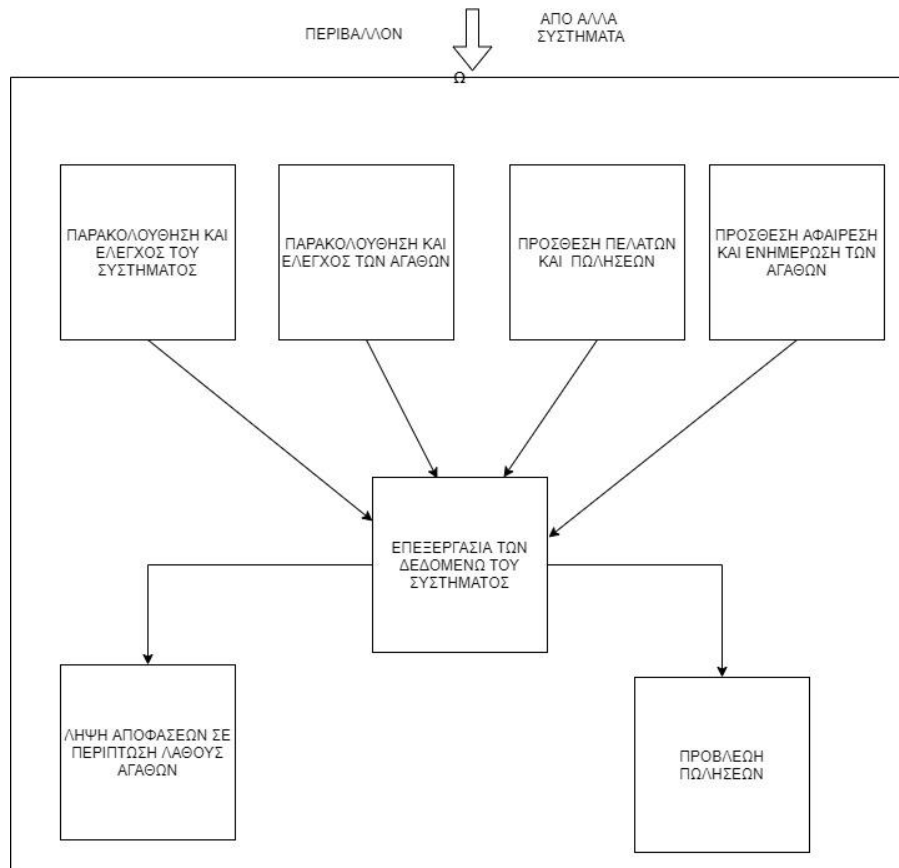
Σχήμα 3.3: Γνωστικό Μοντέλο



Σχήμα 3.4: Μοντέλο Απόκτησης και Διαχείρισης Πόρων



Σχήμα 3.5: Επιχειρησιακό Μοντέλο



Σχήμα 3.6: Μοντέλο Παρακολούθησης και Ελέγχου



## Σύγκριση Μοντέλων με την Πραγματικότητα

Εφόσον τα εννοιολογικά μοντέλα έχουν διαμορφωθεί, οι εμπλεκόμενοι στην προβληματική κατάσταση έρχονται αντιμέτωποι με δύο εικόνες: την «πλούσια εικόνα» και την ιδανική.

Αξίζει να σημειωθεί ότι στόχος δεν είναι η υιοθέτηση αυτών των μοντέλων ούτε η εφαρμογή τους εξασφαλίζει ένα επιτυχημένο αποτέλεσμα. Ουσιαστικά αποτελεί μία βάση, ένα γόνιμο έδαφος για συζήτηση των εμπλεκόμενων πλευρών, αναλυτών και εκπροσώπων του συστήματος, για πιθανές αλλαγές.

Γι' αυτό πρέπει να δοθεί ιδιαίτερη προσοχή στο συγκεκριμένο βήμα της διαδικασίας εφαρμογής της SSM καθώς μέσα από αυτό θα προκύψουν επιθυμητές και εφικτές λύσεις που θα ωφελήσουν πραγματικά την επιχείρηση.





## Εντοπισμός και Υλοποίηση Αλλαγών

Μετά τη σύγκριση των εννοιολογικών μοντέλων με την προβληματική κατάσταση, θα αναδυθούν οι πιθανές λύσεις. Κάποιες θα είναι επιθυμητές αλλά όχι απαραίτητα εφικτές. Έτσι θα ανοίξει νέος κύκλος συζητήσεων μέχρι να προσδιοριστούν λύσεις που να είναι και εφαρμόσιμες.

Να τονιστεί σε αυτό το σημείο ότι ο ανθρώπινος παράγοντας παίζει τον πλέον κρίσιμο ρόλο όχι μόνο για την υλοποίηση των αλλαγών αλλά και για την ουσιαστική εφαρμογή τους. Λαμβάνοντας αυτό υπόψη πρέπει οι αλλαγές να είναι επιθυμητές και αυτό μπορεί να επιτευχθεί μόνο με μια σειρά ενεργειών, όπως η ειλικρινής επικοινωνία, η ουσιαστική συμμετοχή στην λήψη αποφάσεων και η εκπαίδευση για απόκτηση νέων δεξιοτήτων που απαιτούν οι αλλαγές.

Τέλος, εφόσον προσδιοριστεί τι πρέπει να αλλάξει, θα πρέπει να γίνουν όλες οι απαραίτητες ενέργειες για την υλοποίηση των αλλαγών. Σε περίπτωση που προκύψουν νέα προβλήματα σε αυτό το στάδιο, μπορεί να γίνει πάλι αξιοποίηση της SSM για την καλύτερη δυνατή επίλυση τους.



## Κεφάλαιο 4: Διαδικασία ανάπτυξης Πληροφοριακού Συστήματος

Η ανάπτυξη ενός πληροφοριακού συστήματος αποτελεί μια σύνθετη και πολυεπίπεδη διεργασία. Η εφαρμογή μιας μεθοδολογίας ανάπτυξης πληροφοριακού συστήματος παρέχει σημαντικά εχέγγυα για την αποτελεσματική και αποδοτική τεχνική, λειτουργική, και οικονομική διοίκηση καθώς και για την επιτυχή κατάληξη του αντίστοιχου έργου.

Μια μεθοδολογία ανάπτυξης ΠΣ περιλαμβάνει στοιχεία του «τι» (φιλοσοφικά και διαδικαστικά) πρέπει να γίνει για την ανάπτυξη ενός ΠΣ κατά συστηματικό και συγκροτημένο τρόπο και του «πώς» (με ποιες τεχνικές και με ποια εργαλεία ή μέσα θα εφαρμοστεί κάθε τεχνική) πρέπει να γίνει αυτό.

Οι κύριες κατηγορίες μεθοδολογιών είναι οι ακόλουθες:

- Μεθοδολογία Dev (Development) Ops (Operations)
- Μεθοδολογία Κύκλου Ζωής DevOps

## Μεθοδολογία ανάπτυξης DevOps

Για τη δημιουργία του Πληροφοριακού Συστήματος η ομάδα ανάπτυξής μας, επέλεξε τη χρήση των ευέλικτων τεχνικών και πιο συγκεκριμένα τη μεθοδολογία DevOps.

Το DevOps ορίζεται σαν ένα πλαίσιο στο οποίο υπάρχει κάποια συνοχή των εννοιών που πρέπει να αναγνωρισθούν και έχουν σχέση μεταξύ τους

### ➤ Συνεχής Δοκιμή

Οι δοκιμαστικές περιπτώσεις γίνονται νωρίτερα από το λογισμικό και η διαχείριση δοκιμών είναι εξαιρετικά αυτοματοποιημένη. Το Continuous Testing βοηθά τη διαχείριση των δοκιμών που γίνονται σε όλη τη διαδικασία της ανάπτυξης και συμπεριλαμβάνουν δοκιμές μονάδας, δοκιμές ολοκλήρωσης, δοκιμές συστήματος, δοκιμές αποδοχής

### ➤ Ευέλικτη ανάπτυξη

Ο όρος Agile Development δείχνει ότι για την ανάπτυξη λογισμικού στο DevOps στηριζόμαστε σε ένα πολύ μεγάλο μέρος στην Agile. Μπορούν να χρησιμοποιηθούν διάφορες προσεγγίσεις της Agile, όπως Scrum, Kanban και XP.

### ➤ Συνεχής ολοκλήρωση

Με τη συνεχή ενσωμάτωση δίνεται η δυνατότητα στους προγραμματιστές κάθε φορά που εκτελούν μια εφαρμογή να συνδυάζουν τον πηγαίο κώδικα, τις επαληθευμένες περιπτώσεις δοκιμών και τις δυναμικές περιπτώσεις δοκιμών σε υψηλή συχνότητα.

### ➤ Συνεχής παράδοση

Είναι η διαδικασία παράδοσης του λογισμικού αφού γίνονται ενημερώσεις στην παραγωγή σε μικρότερα βήματα, διασφαλίζοντας ότι το λογισμικό μπορεί να κυκλοφορήσει ανά πάσα στιγμή. Η ομάδα έχοντας σαν βάση έναν υψηλό βαθμό αυτοματοποίησης είναι έτοιμη στο να μπορεί να παραδίδει οποτεδήποτε στην παραγωγή.



➤ Συνεχής παρακολούθηση

Η συνεχής παρακολούθηση είναι ουσιαστικό μέρος του DevOps το οποίο παρακολουθεί το λογισμικό (πόρους), τους προγραμματιστές (άτομα) και τις διαδικασίες ανάπτυξης (μεθόδους). Προκειμένου να εντοπίζονται τα σφάλματα πολύ νωρίς μετρώνται συνέχεια οι πόροι σε όλα τα περιβάλλοντα, μετρώνται οι άνθρωποι σε σχέση με την ανάπτυξη των ικανοτήτων τους (γνώση, δεξιότητες, στάση) και η μέθοδος η οποία περιλαμβάνει τη μέτρηση της ταχύτητας και την αποτελεσματικότητα.

➤ Ευέλικτες διαδικασίες

Περιλαμβάνει τη συνέπεια και την αυτοματοποίηση σε νέα επίπεδο ενώ η ομάδα κάνει οποιεσδήποτε αλλαγές στις απαιτήσεις ενός συστήματος ή διαχειρίζεται τη συνολική πολυπλοκότητα ενός έργου.

## Κύκλος Ζωής DevOps

Ο κύκλος ζωής στο DevOps αποτελείται από οκτώ στάδια: πλάνο, κώδικας, κατασκευή, δοκιμή, release, ανάπτυξη, λειτουργία και παρακολούθηση. Παράλληλα, υπάρχουν διάφορα δωρεάν open source εργαλεία (FOSS) που μπορούν να χρησιμοποιηθούν για τη βελτίωση και την αυτοματοποίηση αυτού του κύκλου ζωής.

Τα οκτώ στάδια στην DevOps μεθοδολογία είναι:

- Πλάνο: Αυτό είναι το πρώτο στάδιο όπου ο προγραμματισμός για την ανάπτυξη λογισμικού γίνεται από την ομάδα ανάπτυξης. Τα πάντα, από τις απαιτήσεις λογισμικού έως το χρονοδιάγραμμα ανάπτυξης, σχεδιάζονται και η ομάδα λειτουργεί ανάλογα. Σε αυτό το στάδιο χρησιμοποιούνται σύγχρονα εργαλεία όπως τα Redmine, Trac και Git.
- Κώδικας: Αφού προγραμματιστούν όλα, γίνεται το coding για την ανάπτυξη του λογισμικού. Το coding μπορεί να γίνει από το μηδέν ή μπορεί να επαναχρησιμοποιηθεί, ανάλογα με τις απαιτήσεις. Στην φάση αυτή, πολλά εργαλεία FOSS μπορούν να χρησιμοποιηθούν: Το Git είναι ένα τέτοιο εργαλείο που χρησιμοποιείται στην αυτοματοποίηση της διαδικασίας.
- Κατασκευή: Αφού ολοκληρωθεί το coding, παραδίδεται στους μηχανικούς λογισμικού της ομάδας ανάπτυξης. Ο κώδικας εγκρίνεται ή απορρίπτεται μετά τον έλεγχο. Εάν εγκριθεί, τότε συγχωνεύεται με το maincode του repository. Εργαλεία όπως το **Gradle** χρησιμοποιούνται για την αυτοματοποίηση αυτής της διαδικασίας.
- Δοκιμή: Μόλις ο νέος κωδικός συγχωνευθεί με το maincode, δοκιμάζεται σε εικονικό περιβάλλον, χρησιμοποιώντας κυρίως VM (Kubernetes κλπ). Στο στάδιο αυτό διεξάγονται μια σειρά από χειροκίνητες και αυτοματοποιημένες δοκιμές. Είναι το πιο κρίσιμο στάδιο του DevOps και πρέπει να περάσει χωρίς αποτυχία. Σε αυτό το στάδιο μπορεί να αυξήσει το τελικό χρονοδιάγραμμα. Για να ολοκληρωθεί επιτυχώς το στάδιο χωρίς καθυστέρηση, χρησιμοποιούνται διάφορα εργαλεία συνεχούς δοκιμής. Εργαλεία όπως το **CruiseControl** και το **Selenium** χρησιμοποιούνται για την αυτοματοποίηση της διαδικασίας.
- Release: Αφού ο κώδικας δοκιμαστεί επιτυχώς, προετοιμάζεται για το release. Στο στάδιο αυτό, η ομάδα ανάπτυξης αποφασίζει ποιες δυνατότητες του προϊόντος λογισμικού πρέπει να είναι ενεργοποιημένες ή απενεργοποιημένες by default και τότε πρέπει να κυκλοφορήσει. Αυτό είναι το τελευταίο στάδιο του κύκλου ανάπτυξης του DevOps και μετά, το λογισμικό παραδίδεται στην ομάδα operations. Αυτό το στάδιο μπορεί επίσης να αυτοματοποιηθεί χρησιμοποιώντας εργαλεία FOSS όπως



τα **Jenkins** και **Bamboo**.

- **Ανάπτυξη:** Το στάδιο αυτό ανήκει στην ομάδα operations και ξεκινά μετά το release του λογισμικού από την ομάδα ανάπτυξης. Επίσης το λογισμικό αναπτύσσεται από την ομάδα του operations χρησιμοποιώντας διάφορα εργαλεία. Το release διαμορφώνεται σύμφωνα με τις απαιτήσεις της ομάδας operations. Τα εργαλεία **Puppet**, **Ansible** και **Saltstack** χρησιμοποιούνται για την αυτοματοποίηση της διαδικασίας ανάπτυξης.
- **Λειτουργία:** Μετά την ενημέρωση και τη διαμόρφωση του λογισμικού, η ομάδα operations αρχίζει να λειτουργεί τα προϊόντα και τις υπηρεσίες με το ενημερωμένο λογισμικό.
- **Monitoring:** Αυτό είναι το τελευταίο στάδιο για την ομάδα operations και για τον κύκλο ζωής στο DevOps. Εδώ συγκεντρώνονται οι απαιτήσεις των πελατών και τα δεδομένα αποστέλλονται στην ομάδα ανάπτυξης για να ενημερώσουν το προϊόν / την υπηρεσία λογισμικού. Τα εργαλεία όπως το **Nagios** χρησιμοποιούνται από την ομάδα operations για την αυτοματοποίηση της διαδικασίας Monitoring.

## Χρονοδιάγραμμα υλοποίησης

Ένα από τα πλέον βασικά στοιχεία για τον προγραμματισμό ενός έργου αποτελεί η οριοθέτηση του χρονικού του ορίζοντα. Η διαδικασία αυτή του χρονοπρογραμματισμού είναι καίρια για την υλοποίηση του συστήματος. Για το σκοπό αυτό θα χρησιμοποιήσουμε δυο ευρέως διαδεδομένες τεχνικές. Τα διαγράμματα *PERT* και *Gantt*.

Τα διαγράμματα PERT αποτελούν μια γραφική κατασκευή ενός δικτύου επιμέρους εργασιών του έργου. Σύμφωνα με αυτή, τα έργα μπορούν να οργανωθούν κατά γεγονότα και επιμέρους εργασίες.

Τα διαγράμματα Gantt χρησιμοποιούνται συνήθως για την κατασκευή του χρονοδιαγράμματος ενός έργου και την αξιολόγηση της προόδου του.

### Διάγραμμα PERT

Όλες οι επιμέρους εργασίες και γεγονότα, καθώς και οι αναμενόμενοι χρόνοι περάτωσης του έργου περιγράφονται στους πίνακες 4.1 και 4.2.

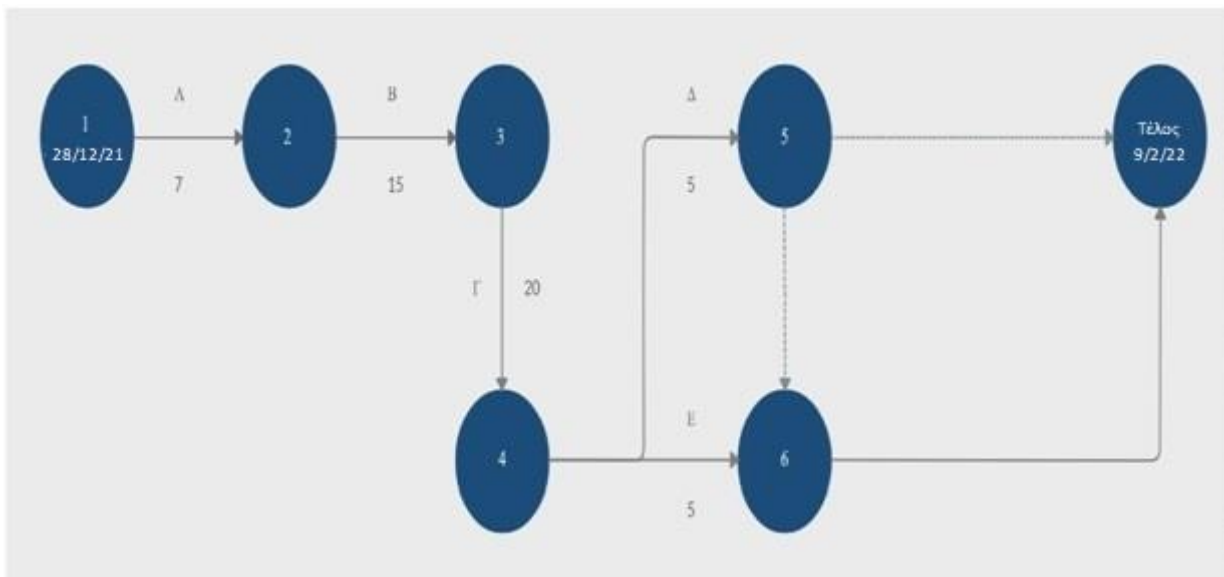
**Πίνακας 4.1: Δεδομένα προγραμματισμού έργου για το διάγραμμα PERT**

Εργασία	Περιγραφή εργασίας	Γεγονός	Προηγούμενο γεγονός	Επόμενο γεγονός	Αναμενόμενος χρόνος
A	Μελέτη απαιτήσεων	2	1	3	7
B	Σχεδίαση & Οργάνωση Λειτουργιών ΠΣ	3	2	4	15
Γ	Προγραμματισμός	4	3	5	20
Δ	Έλεγχος	5	4	6	5
E	Τεκμηρίωση	6	5	-	5

Πίνακας 4.2

Γεγονός	Περιγραφή	Εργασία	Περιγραφή
1	Ανάληψη εργασίας	-	-
2	Ανάλυση απαιτήσεων	A	Μελέτη απαιτήσεων
3	Σχεδίαση	B	Σχεδίαση & Οργάνωση Λειτουργιών ΠΣ
4	Συγγραφή κώδικα	Γ	Προγραμματισμός
5	Πραγματοποίηση ελέγχων	Δ	Έλεγχος
6	Συγγραφή τεκμηριωτικού και συνοδευτικού υλικού	E	Τεκμηρίωση

Με βάση τα δεδομένα των **πινάκων 4.1** και **4.2** κατασκευάζουμε το διάγραμμα PERTόπως αυτό εμφανίζεται στο **σχήμα 4.1**.



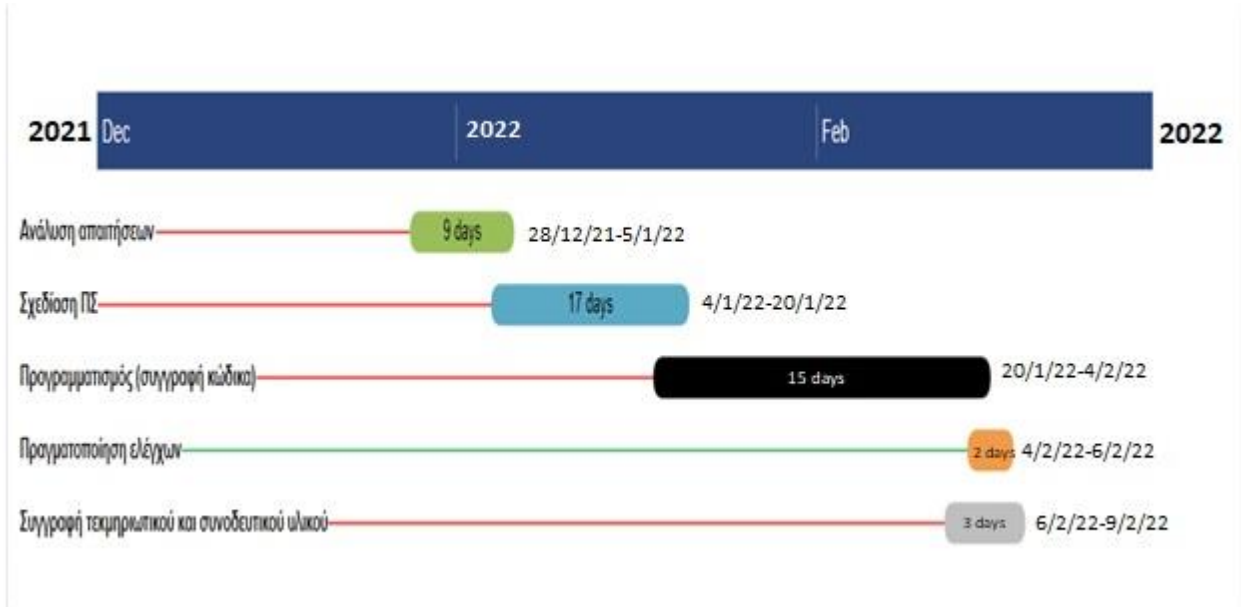
Σχήμα 4.1

Τέλος, να αναφέρουμε ότι η κρίσιμη διαδρομή του διαγράμματός μας αποτελείται από τις επιμέρους εργασίες A, B, Γ και E.



## Διάγραμμα Gantt

Στο διάγραμμα Gantt που φαίνεται στο **σχήμα 4.2** παρουσιάζονται οι χρόνοι που απαιτήθηκαν για την υλοποίηση της κάθε επιμέρους εργασίας.



Σχήμα 4.2

Αυτό που παρατηρούμε στο διάγραμμα Gantt είναι το γεγονός πως ο πραγματικός χρόνος υλοποίησης της κάθε εργασίας (εκτός από την πραγματοποίηση των ελέγχων οι οποίοι παρέμειναν εντός χρονοδιαγράμματος) ήταν ελαφρώς μεγαλύτερος από τον αναμενόμενο, ο οποίος είχε προβλεφθεί στον αρχικό χρονοπρογραμματισμό του έργου.

## Κοστολόγηση ΠΣ

Για την κοστολόγηση και τη μέτρηση του λειτουργικού μεγέθους του λογισμικού μας θα εφαρμόσουμε τη μεθοδολογία Ανάλυσης Λειτουργικών Σημείων (Function Points Analysis – FPA).

Για την εφαρμογή της μεθόδου θα πρέπει πρώτα να κατηγοριοποιήσουμε τα συστατικά του συστήματος. Κατόπιν, μπορούμε να προχωρήσουμε στη μέτρηση των Λειτουργικών Σημείων (FP) τα οποία προκύπτουν ως το γινόμενο του Συντελεστή Μη Προσαρμοσμένων Λειτουργικών Σημείων (UFP) επί το Συντελεστή Προσαρμογής (VAF).

$$FP = UFP \times VAF$$

### Συστατικά συστήματος

Με βάση την FPA, τα συστήματα διαιρούνται σε πέντε κύριες κατηγορίες συστατικών:

- Είσοδοι (External Inputs – EI): Αποτελούν μία βασική διαδικασία κατά την οποία εισέρχονται δεδομένα στο σύστημα από εξωτερικές πηγές, όπως οθόνες εισαγωγής δεδομένων είτε ακόμη και κάποια άλλη εφαρμογή.
- Έξοδοι (External Outputs - EO): Κατά τη διαδικασία αυτή λαμβάνει χώρα εξαγωγή δεδομένων από το σύστημα και επιπλέον μπορεί να γίνει ενημέρωση ενός Εσωτερικού Λογικού Αρχείου. Τα δεδομένα αυτής της διαδικασίας μπορούν να δημιουργήσουν αναφορές ή αρχεία που αποστέλλονται σε άλλες εφαρμογές.
- Επερωτήσεις (External Inquiries – EQ): Αφορά διαδικασίες τόσο εισόδου όσο και εξόδου που μπορούν να οδηγήσουν στην ανάκτηση δεδομένων από ένα ή περισσότερα Εσωτερικά Λογικά Αρχεία και Εξωτερικά Αρχεία Διεπαφής. Η διαδικασία εισόδου δεν ενημερώνει κάποιο αρχείο, ενώ η πλευρά της εξόδου δεν περιέχει εξαγόμενα δεδομένα.
- Εσωτερικά Λογικά Αρχεία (Internal Logical Files - ILF): Αποτελούν μια ομάδα λογικά συσχετισμένων δεδομένων τα οποία βρίσκονται αποκλειστικά εντός των ορίων της εφαρμογής και συντηρούνται μέσω των Εισόδων.
- Εξωτερικά Αρχεία Διεπαφής (External Interface Files - EIF): Το σύνολο αυτών των δεδομένων μπορεί να χρησιμοποιηθεί μόνο για λόγους αναφοράς. Τα δεδομένα βρίσκονται

εκτός εφαρμογής και συντηρούνται από κάποια τρίτη εφαρμογή. Τα Εξωτερικά Αρχεία Διεπαφής αποτελούν Εσωτερικά Λογικά Αρχεία για κάποια άλλη εφαρμογή.

## Συντελεστής Μη Προσαρμοσμένων Λειτουργικών Σημείων – Unadjusted Functional Points (UFP)

Αφού κατηγοριοποιήσουμε τα συστατικά του συστήματος στις πέντε κύριες κατηγορίες τα ταξινομούμε με βάση τρεις βαθμολογίες: «ΧΑΜΗΛΗ», «ΜΕΣΗ», «ΥΨΗΛΗ».

Για τις Εισόδους, Εξόδους και Επερωτήσεις η βαθμολογία εξάγεται με βάση τον αριθμό των Αρχείων Αναφοράς (FTR) και τον αριθμό των Τύπων Δεδομένων (DET).

Όσον αφορά τα Εσωτερικά Λογικά Αρχεία και τα Εξωτερικά Αρχεία Διεπαφής, η βαθμολογία προκύπτει από την καταγραφή των Τύπων Εγγραφής (RET) και των Τύπων Δεδομένων (DET).

Ο τελικός αριθμός UFP προκύπτει από το άθροισμα των πινάκων των επιμέρους κατηγοριών:

$$UFP = \sum(EI + EO + EQ + ILF + EIF)$$

Ο τρόπος μέτρησης Λειτουργικών Σημείων ανά κατηγορία παρουσιάζεται στους **πίνακες 4.3, 4.4, 4.5, 4.6, 4.7**:

**Πίνακας 4.3: Μέτρηση Λειτουργικών Σημείων Εισόδων**

Είσοδοι (EI)			
Αρχεία Αναφοράς (FTR)	Τύπος Δεδομένων (DET)		
	1-4	5-15	>15
<2	ΧΑΜΗΛΗ (X3)	ΧΑΜΗΛΗ (X3)	ΜΕΣΗ (X4)
2	ΧΑΜΗΛΗ (X3)	ΜΕΣΗ (X4)	ΥΨΗΛΗ (X6)
>2	ΜΕΣΗ (X4)	ΥΨΗΛΗ (X6)	ΥΨΗΛΗ (X6)

**Πίνακας 4.4: Μέτρηση Λειτουργικών Σημείων Εξόδων**

Έξοδοι (EO)			
Αρχαία Αναφοράς (FTR)	Τύπος Δεδομένων (DET)		
	1-5	6-19	>19
<2	ΧΑΜΗΛΗ (X4)	ΧΑΜΗΛΗ (X4)	ΜΕΣΗ (X5)
2-3	ΧΑΜΗΛΗ (X4)	ΜΕΣΗ (X5)	ΥΨΗΛΗ (X7)
>3	ΜΕΣΗ (X5)	ΥΨΗΛΗ (X7)	ΥΨΗΛΗ (X7)

**Πίνακας 4.5: Μέτρηση Λειτουργικών Σημείων Επερωτήσεων**

Επερωτήσεις (EQ)			
Αρχαία Αναφοράς (FTR)	Τύπος Δεδομένων (DET)		
	1-5	6-19	>19
<2	ΧΑΜΗΛΗ (X3)	ΧΑΜΗΛΗ (X3)	ΜΕΣΗ (X4)
2-3	ΧΑΜΗΛΗ (X3)	ΜΕΣΗ (X4)	ΥΨΗΛΗ (X6)
>3	ΜΕΣΗ (X4)	ΥΨΗΛΗ (X6)	ΥΨΗΛΗ (X6)

**Πίνακας 4.6: Μέτρηση Λειτουργικών Σημείων Εσωτερικών Λογικών Αρχείων**

Εσωτερικά Λογικά Αρχεία (ILF)			
Τύπος Εγγραφής (RET)	Τύπος Δεδομένων (DET)		
	1-19	20-50	>51
1	ΧΑΜΗΛΗ (X7)	ΧΑΜΗΛΗ (X7)	ΜΕΣΗ (X10)
2-5	ΧΑΜΗΛΗ (X7)	ΜΕΣΗ (X10)	ΥΨΗΛΗ (X15)
>5	ΜΕΣΗ (X10)	ΥΨΗΛΗ (X15)	ΥΨΗΛΗ (X15)

**Πίνακας 4.7: Μέτρηση Λειτουργικών Σημείων Εξωτερικών Αρχείων Διεπαφής**

Εξωτερικά Αρχεία Διεπαφής (EIF)			
Τύπος Εγγραφής (RET)	Τύπος Δεδομένων (DET)		
	1-19	20-50	>51
1	ΧΑΜΗΛΗ (X5)	ΧΑΜΗΛΗ (X5)	ΜΕΣΗ (X7)
2-5	ΧΑΜΗΛΗ (X5)	ΜΕΣΗ (X7)	ΥΨΗΛΗ (X10)
>5	ΜΕΣΗ (X7)	ΥΨΗΛΗ (X10)	ΥΨΗΛΗ (X10)

## Συντελεστής Προσαρμογής – Value Adjustment Factor (VAF)

Ο Συντελεστής Προσαρμογής βασίζεται σε 14 γενικά χαρακτηριστικά (πίνακας 4.9) του συστήματος τα οποία αξιολογούν τη γενικότερη λειτουργικότητα της υπό αξιολόγηση εφαρμογής. Καθένα από αυτά τα χαρακτηριστικά συνοδεύεται από μια περιγραφή η οποία καθορίζει το βαθμό επίδρασης του στο σύστημα (πίνακας 4.8). Η κλίμακα επιρροής κυμαίνεται από το 0 έως το 5:

**Πίνακας 4.8: Κλίμακα επιρροής**

Βαθμός Επίδρασης	
0	Καθόλου
1	Τυχαία
2	Ήπια
3	Μέση
4	Σημαντική
5	Ισχυρή

**Πίνακας 4.9: Γενικά χαρακτηριστικά του συστήματος**

Γενικά Χαρακτηριστικά	Σύντομη Περιγραφή	
1	Επικοινωνία Δεδομένων (Data Communication)	Πόσες συνδέσεις υπάρχουν, ώστε να συμβάλουν τη μεταφορά πληροφοριών του συστήματος;
2	Κατανομημένη Επεξεργασία Δεδομένων (Distributed Data Processing)	Πώς γίνεται ο χειρισμός των κατανομημένων δεδομένων και των διάφορων λειτουργιών;
3	Απόδοση (Performance)	Αντιστοιχεί ο χρόνος απόκρισης με τις απαιτήσεις του χρήστη;
4	Διαμόρφωση Υπερφόρτωσης (Heavily Used Configuration)	Πόσο καταπονείται το εν λόγω υλικό από την εφαρμογή του λογισμικού;
5	Ρυθμός Διεξαγωγής (Transaction Rate)	Πόσο συχνά διεξάγονται οι διάφορες ενέργειες;
6	Απευθείας Καταχώρηση Δεδομένων (On-Line Data Entry)	Τι ποσοστό των πληροφοριών εισάγεται με απευθείας καταχωρήσεις;
7	Αποδοτικότητα Τελικού Χρήστη (End-User Efficiency)	Η εφαρμογή σχεδιάστηκε με στόχο την αύξηση της αποδοτικότητας του τελικού χρήστη;
8	Απευθείας Ενημέρωση (On-Line Update)	Πόσα Εσωτερικά Λογικά Αρχεία ενημερώνονται με απευθείας καταχωρήσεις;
9	Σύνθετη Επεξεργασία (Complex Processing)	Η εφαρμογή έχει αυξημένη λογική και μαθηματική επεξεργασία;
10	Επαναχρησιμοποίηση (Reusability)	Η εφαρμογή σχεδιάστηκε για να εξυπηρετεί τις ανάγκες ενός ή περισσότερων χρηστών;
11	Ευκολία Εγκατάστασης (Installation Ease)	Πόσο δύσκολη είναι η μετατροπή και εγκατάσταση;
12	Ευκολία Λειτουργίας (Operational Ease)	Πόσο αποτελεσματικές και/ή αυτοματοποιημένες είναι οι διαδικασίες;



13	Πολλαπλά Σημεία Εγκατάστασης (Multiple Sites)	Η εφαρμογή εγκαταστάθηκε για να υποστηρίξει πολλαπλές εγκαταστάσεις σε διάφορους οργανισμούς;
14	Διευκόλυνση Αλλαγής (Facilitate Change)	Η εφαρμογή σχεδιάστηκε με τρόπο που διευκολύνει τις αλλαγές;

Ο Συντελεστής Προσαρμογής (VAF) υπολογίζεται από τον ακόλουθο τύπο:

$$VAF = 0,65 + [(\sum Di) / 100],$$

όπου  $Di$  ο βαθμός επιρροής του κάθε γενικού χαρακτηριστικού του συστήματος

## Μέτρηση Λειτουργικών Σημείων

Ο Συντελεστής Μη Προσαρμοσμένων Λειτουργικών Σημείων προκύπτει με βάση τα δεδομένα του πίνακα 4.10.

Πίνακας 4.10: Μέτρηση Μη Προσαρμοσμένων Λειτουργικών Σημείων

	ΧΑΜΗΛΗ	ΜΕΣΗ	ΥΨΗΛΗ	
Είσοδοι (EI)	9 (X3) = 27	3 (X4) = 12	2 (X6) = 12	51
Έξοδοι (EO)	3 (X4) = 12	5 (X5) = 25	1 (X7) = 7	44
Επερωτήσεις (EQ)	8 (X3) = 24	14 (X4) = 56	4 (X6) = 24	104
Εσωτερικά Λογικά Αρχεία (ILF)	3 (X7) = 21	3 (X10) = 30	4 (X15) = 60	111
Εξωτερικά Αρχεία Διεπαφής (EIF)	3 (X5) = 15	2 (X7) = 14	0	29
<b>UFP</b>				<b>339</b>

Ο Συντελεστής Προσαρμογής προκύπτει με βάση τα δεδομένα του πίνακα 4.11.

Πίνακας 4.11: Υπολογισμός Συντελεστή προσαρμογής

Γενικά Χαρακτηριστικά		Βαθμός Επίδρασης
1	Επικοινωνία Δεδομένων (Data Communication)	3
2	Κατανεμημένη Επεξεργασία Δεδομένων (Distributed Data Processing)	2
3	Απόδοση (Performance)	2
4	Διαμόρφωση Υπερφόρτωσης (Heavily Used Configuration)	1
5	Ρυθμός Διεξαγωγής (Transaction Rate)	1
6	Απευθείας Καταχώρηση Δεδομένων (On-Line Data Entry)	2
7	Αποδοτικότητα Τελικού Χρήστη (End-User Efficiency)	2
8	Απευθείας Ενημέρωση (On-Line Update)	3



9	Σύνθετη Επεξεργασία (Complex Processing)	3
10	Επαναχρησιμοποίηση (Reusability)	5
11	Ευκολία Εγκατάστασης (Installation Ease)	4
12	Ευκολία Λειτουργίας (Operational Ease)	4
13	Πολλαπλά Σημεία Εγκατάστασης (Multiple Sites)	5
14	Διευκόλυνση Αλλαγής (Facilitate Change)	4
<b>ΣDi</b>		<b>41</b>

$$VAF = 0,65 + (41/ 100) = 1,08$$

Πλέον, μπορούμε να προχωρήσουμε στη μέτρηση των Λειτουργικών Σημείων του συστήματός μας τα οποία προκύπτουν ως εξής:

$$FP = 339 \times 1,08 = 366,12$$

### Υπολογισμός κόστους

Σύμφωνα με στοιχεία που αντλήθηκαν από την αναφορά του Jones Capers (αντιπρόεδρος της Namcook Analytics LLC) το 2017 με τίτλο: “*Software Economics and Function Point Metrics: Thirty years of IFPUG Progress*”, τα λειτουργικά σημεία ανά μήνα στην Ελλάδα ανέρχονται σε 11,7.

Διαιρώντας, λοιπόν, τα συνολικά λειτουργικά σημεία που μετρήσαμε στην προηγούμενη ενότητα (366,12) με τα λειτουργικά σημεία ανά μήνα (11,7) προκύπτει ότι απαιτούνται 31,29 ανθρωπομήνες για την ανάπτυξη του ΠΣ.

Υπολογίζοντας το κόστος του ανθρωπομήνα στα 900 €, τότε το κόστος ανάπτυξης του ΠΣ ανέρχεται σε  $31,29 \times 900 = 28.161$  €.



## Κεφάλαιο 5: Σχεδιασμός Πληροφοριακού Συστήματος

Στο προηγούμενο κεφάλαιο πραγματοποιήθηκε η επιλογή μεθοδολογίας, η οποία θα ακολουθηθεί από την ομάδα ανάπτυξης για τη δημιουργία του Πληροφοριακού Συστήματος. Επομένως, μπορούμε να προχωρήσουμε στο στάδιο του σχεδιασμού του ΠΣ. Σε αυτή την ενότητα θα παρουσιαστούν λεπτομέρειες, οι οποίες σχετίζονται με το σχεδιασμό, τόσο σε φυσικό όσο και σε λογικό επίπεδο.

Το πρώτο τη χρήση καρτών ιστορίας (story cards). βήμα που πρέπει να γίνει προς αυτή την κατεύθυνση είναι ο προσδιορισμός των απαιτήσεων, ο οποίος στην περίπτωση μας (DevOps) γίνεται με

Αφότου συλλεγούν οι αρχικές απαιτήσεις του συστήματος από τον πελάτη, μπορεί να ξεκινήσει η διαδικασία σχεδιασμού του συστήματος λογισμικού. Για το σκοπό αυτό θα χρησιμοποιηθούν τεχνικές δομημένης ανάλυσης και σχεδιασμού συστημάτων.

Πιο συγκεκριμένα, θα γίνει χρήση διαγραμμάτων της ενοποιημένης γλώσσας σχεδιασμού (Unified Modeling Language) (UML), μέσω της οποίας θα γίνει ο αρχικός προσδιορισμός της δομής του συστήματος.

Επιπλέον, σε αυτό το κεφάλαιο λαμβάνει χώρα και η σχεδίαση της Βάσης Δεδομένων του συστήματος (περιλαμβάνονται το διάγραμμα οντοτήτων-συσχετίσεων, καθώς και το σχεσιακό μοντέλο).

## Προσδιορισμός Απαιτήσεων

Η DevOps ανήκει στην ευρύτερη οικογένεια των ευέλικτων μεθοδολογιών ανάπτυξης λογισμικού και ως εκ τούτου η συλλογή και ανάλυση των απαιτήσεων λαμβάνει χώρα μέσω καρτών ιστορίας (story cards) και όχι μέσω αυστηρά τυποποιημένων εγγράφων.

Μέσα από τις κάρτες ιστορίας οι χρήστες ορίζουν το «τι» περιγράφοντας ένα κομμάτι της χρησιμότητας της εφαρμογής από τη δική τους οπτική γωνία, καθώς και την αξία που περιμένουν να αποκομίσει η εταιρία από αυτό το χαρακτηριστικό του λογισμικού.

Σε αυτή την ενότητα, γίνεται παρουσίαση των αρχικών απαιτήσεων του συστήματος, όπως αυτές παρουσιάστηκαν από τους χρήστες υπό τη μορφή story cards.

### Είσοδος υπαλλήλου στο σύστημα

<b>Αριθμός Ιστορίας:</b> 1	<b>Τίτλος:</b> Είσοδος του υπάλληλου στο σύστημα
<b>Ενέργεια:</b> Η είσοδος του υπαλλήλου στο σύστημα για να δει να δεδομένα της εταιρείας στο σύστημα.	
<b>Σκοπός:</b> Η άμεση πρόσβαση σε όλες τις πληροφορίες όλων των δεδομένων.	

Για την περάτωση της συγκεκριμένης ενέργειας ο υπάλληλος θα πρέπει να έχει πρόσβαση στο σύστημα για να δει τα δεδομένα του.



## Επεξεργασία δεδομένων του συστήματος

**Αριθμός Ιστορίας:**  
2

**Τίτλος:** Επεξεργασία των δεδομένων του συστήματος

**Ενέργεια:** Η επεξεργασία των δεδομένων στο σύστημα (πρόσθεση, τροποποίηση, διαγραφή, αναζήτηση).

**Σκοπός:** Η άμεση επεξεργασία οποιουδήποτε δεδομένου στο σύστημα.

Η επεξεργασία των δεδομένων του συστήματος περιλαμβάνει τις ενέργειες:

- Πρόσθεση
- Τροποποίηση
- Διαγραφή
- Αναζήτηση



## Έξοδος υπαλλήλου από το σύστημα

**Αριθμός Ιστορίας:**  
3

**Τίτλος:** Έξοδος του υπάλληλου από το σύστημα

**Ενέργεια:** Η έξοδος του υπαλλήλου στο σύστημα αφού ολοκληρώσει την επεξεργασία.

**Σκοπός:** Η έξοδος του από το σύστημα για να μην έχουν πρόσβαση οι μην έχοντες εργασία.

## Σχεδίαση ΠΣ με χρήση ενοποιημένης γλώσσας σχεδιασμού (Unified Modeling Language) (UML)

Η ενοποιημένη γλώσσα σχεδιασμού (Unified Modeling Language) (UML) αποτελεί την πλέον βασική και ευρέως χρησιμοποιούμενη γλώσσα μοντελοποίησης συστημάτων. Μέσω των διαφόρων διαγραμμάτων της UML παρέχεται η δυνατότητα οπτικοποίησης των βασικών χαρακτηριστικών ενός πληροφοριακού συστήματος.

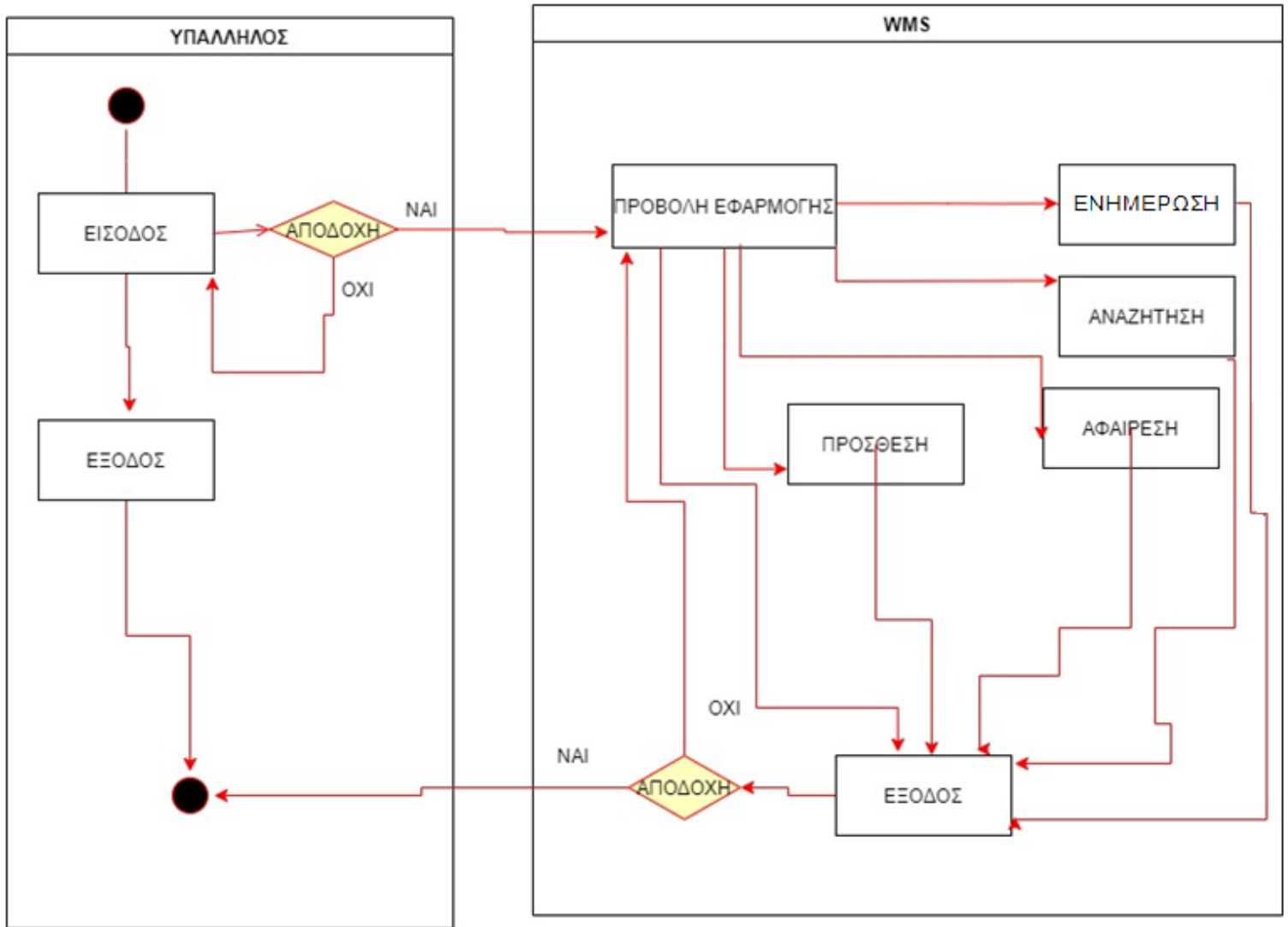
Επιπλέον, μέσω της UML μπορεί να επιτευχθεί ο προσδιορισμός της δομής και της συμπεριφοράς του συστήματος, καθώς επίσης και η δημιουργία προτύπων πάνω στα οποία μπορεί να βασιστεί κατασκευή του ΠΣ. Ενώ, σημαντικό της πλεονέκτημα αποτελεί το γεγονός ότι είναι ανεξάρτητη του πεδίου εφαρμογής και της μεθοδολογίας ανάπτυξης του λογισμικού.

Η UML περιλαμβάνει 14 διαγράμματα τα οποία διακρίνονται σε δυο κύριες κατηγορίες: τα *διαγράμματα συμπεριφοράς*, τα οποία παρουσιάζουν τις αλληλεπιδράσεις των συστατικών του συστήματος και τα *διαγράμματα δομής*, που παρουσιάζουν τις στατικές σχέσεις μεταξύ των συστατικών στοιχείων.

Μερικά από τα πιο συχνά χρησιμοποιούμενα διαγράμματα UML, τα οποία χρησιμοποιούμε και στην παρούσα εργασία είναι:

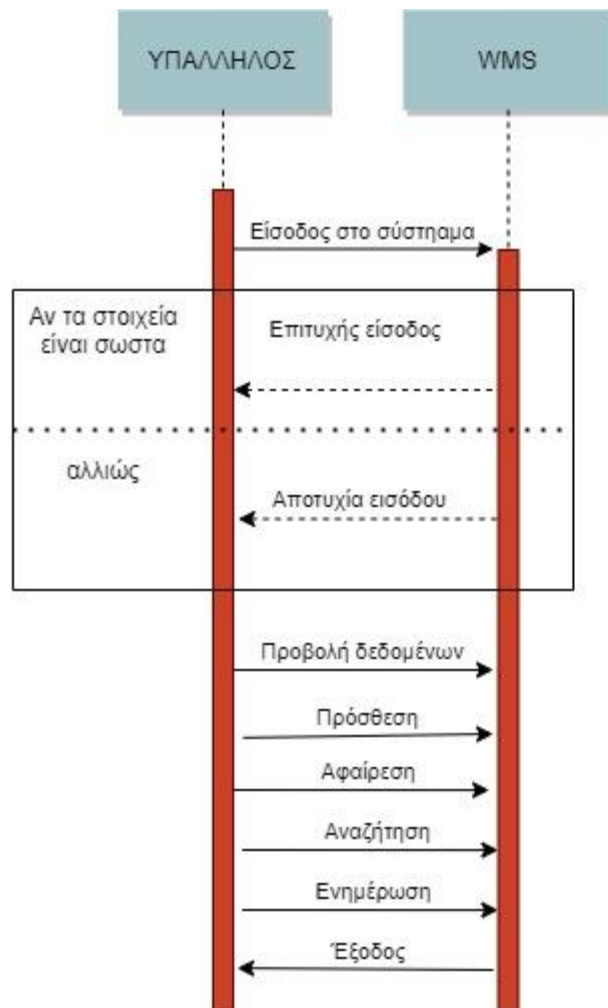
- Διαγράμματα δραστηριοτήτων (Activity diagrams)
- Διάγραμμα ακολουθίας (Sequence diagrams)

### Διάγραμμα Δραστηριοτήτων (UML Activity Diagram)



Σχήμα 5.1: Διάγραμμα Δραστηριοτήτων

## Διάγραμμα Ακολουθίας (Sequence Diagram)



Σχήμα 5.2: Διάγραμμα Ακολουθίας





## Σχεδίαση Βάσης Δεδομένων

Με βάση τα όσα έχουμε αναφέρει στα προηγούμενα κεφάλαια, η Σχεσιακή Βάση Δεδομένων που θα σχεδιάσουμε για την περάτωση του Πληροφοριακού Συστήματος της περίπτωσης μας θα αποτελείται από τους εξής τέσσερις πίνακες:

- Πίνακας Clients: Αποτελεί τον πίνακα ο οποίος περιέχει τα δεδομένα των πελατών της εταιρείας.
- Πίνακας Products: Αποτελεί τον πίνακα ο οποίος περιέχει τα δεδομένα των προϊόντων της εταιρείας.
- Πίνακας Warehouses: Αποτελεί τον πίνακα ο οποίος περιέχει τα δεδομένα των αποθηκών της εταιρείας.
- Πίνακας Sells: Αποτελεί τον πίνακα ο οποίος περιέχει τα δεδομένα των πωλήσεων εταιρείας.

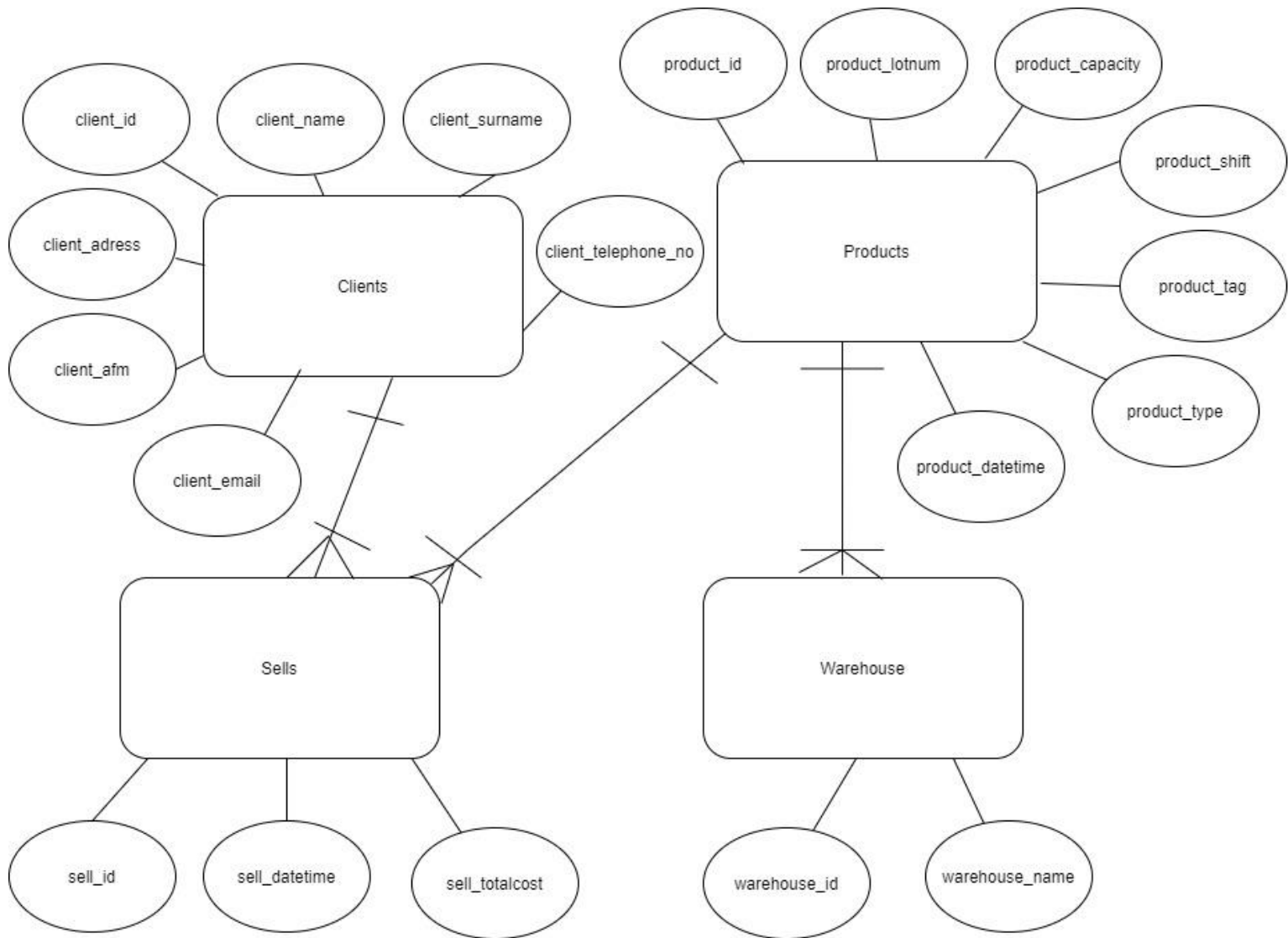


Οι σχέσεις μεταξύ των πινάκων της Βάσης Δεδομένων ορίζονται ως εξής:

- Ο πίνακας *Products* έχει σχέση ένα προς πολλά (1:N) με τον πίνακα *Warehouse*, αφού κάθε αποθήκη μπορεί να έχει πολλά προϊόντα, ενώ κάθε προϊόν δεν μπορεί να ανήκει σε περισσότερες από μια αποθήκες.
- Ο πίνακας *Clients* έχει σχέση ένα προς πολλά (1:N) με τον πίνακα *Sells*, αφού ένας πελάτης μπορεί να υποβάλει πολλές αγορές, αλλά η κάθε αγορά αντιστοιχεί σε ένα πελάτη.
- Ο πίνακας *Products* έχει σχέση ένα προς πολλά (N:1) με τον πίνακα *Sells*, αφού ένα προϊόν μπορεί να κάνει περισσότερες από μια πωλήσεις, ενώ η κάθε πώληση αντιστοιχεί ένα προϊόν.

Με βάση αυτές τις προϋποθέσεις προχωρήσαμε στη δημιουργία του μοντέλου οντοτήτων-συσχετίσεων, όπως αυτό εμφανίζεται στο **Σχήμα 5.3**

### Μοντέλο οντοτήτων-συσχετίσεων



Σχήμα 5.3: Μοντέλο οντοτήτων-συσχετίσεων



### Πίνακας Clients

- CLIENT\_ID: Αποτελεί το πρωτεύον κλειδί (primary key) του πίνακα, είναι αριθμητικού τύπου (int) και έχει τη δυνατότητα αυτόματης προσαύξησης (auto increment) για κάθε νέα εγγραφή του πίνακα.
- CLIENT\_NAME: Προσδιορίζει το όνομα ενός πελάτη, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- CLIENT\_SURNAME: Προσδιορίζει το επίθετο ενός πελάτη, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- CLIENT\_MAIL: Προσδιορίζει τη διεύθυνση ηλεκτρονικής αλληλογραφίας ενός πελάτη, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- CLIENT\_ADDRESS: Προσδιορίζει το email ενός πελάτη είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- CLIENT\_AFM: Προσδιορίζει το αφμ ενός πελάτη είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- CLIENT\_TELEPHONR\_NO: Προσδιορίζει το αριθμό τηλεφώνου ενός πελάτη είναι τύπου int και έχει μέγιστο μέγεθος 10 χαρακτήρες. Το πεδίο αυτό μπορεί να είναι κενό

### Πίνακας Products

- PRODUCT\_NP: Αποτελεί το πρωτεύον κλειδί (primary key) του πίνακα, είναι αριθμητικού τύπου (int) και έχει τη δυνατότητα αυτόματης προσαύξησης (auto increment) για κάθε νέα εγγραφή του πίνακα.
- PRODUCT\_LOTNUM: Προσδιορίζει το όνομα ενός εργαζόμενου, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.



- **PRODUCT\_CAPACITY:** Προσδιορίζει το lotnumber του προϊόντος, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- **PRODUCT\_SHIFT:** Προσδιορίζει την βάρδια παραγωγής του προϊόντος, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- **PRODUCT\_DATETIME:** Προσδιορίζει την ημέρα παραγωγής του προϊόντος, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- **PRODUCT\_TAG:** Προσδιορίζει την ετικέτα του προϊόντος, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.
- **PRODUCT\_TYPE:** Προσδιορίζει το επίθετο τύπο του προϊόντος, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.

#### Πίνακας Warehouses

- **WAREHOUSE\_ID:** Αποτελεί το πρωτεύον κλειδί (primary key) του πίνακα, είναι αριθμητικού τύπου (int) και έχει τη δυνατότητα αυτόματης προσαύξησης (auto increment) για κάθε νέα εγγραφή του πίνακα.
- **WAREHOUSE\_NAME:** Προσδιορίζει το όνομα της αποθήκης, είναι τύπου varchar και έχει μέγιστο μέγεθος 25 χαρακτήρες. Το πεδίο αυτό δεν μπορεί να είναι κενό.

#### Πίνακας Sells

- **SELL\_ID:** Αποτελεί το πρωτεύον κλειδί (primary key) του πίνακα, είναι αριθμητικού τύπου (int) και έχει τη δυνατότητα αυτόματης προσαύξησης (auto increment) για κάθε νέα εγγραφή του πίνακα.
- **SELL\_DATETIME:** Είναι η ημερομηνία υποβολής της αίτησης επένδυσης και είναι τύπου date. Το πεδίο αυτό δεν μπορεί να είναι κενό.



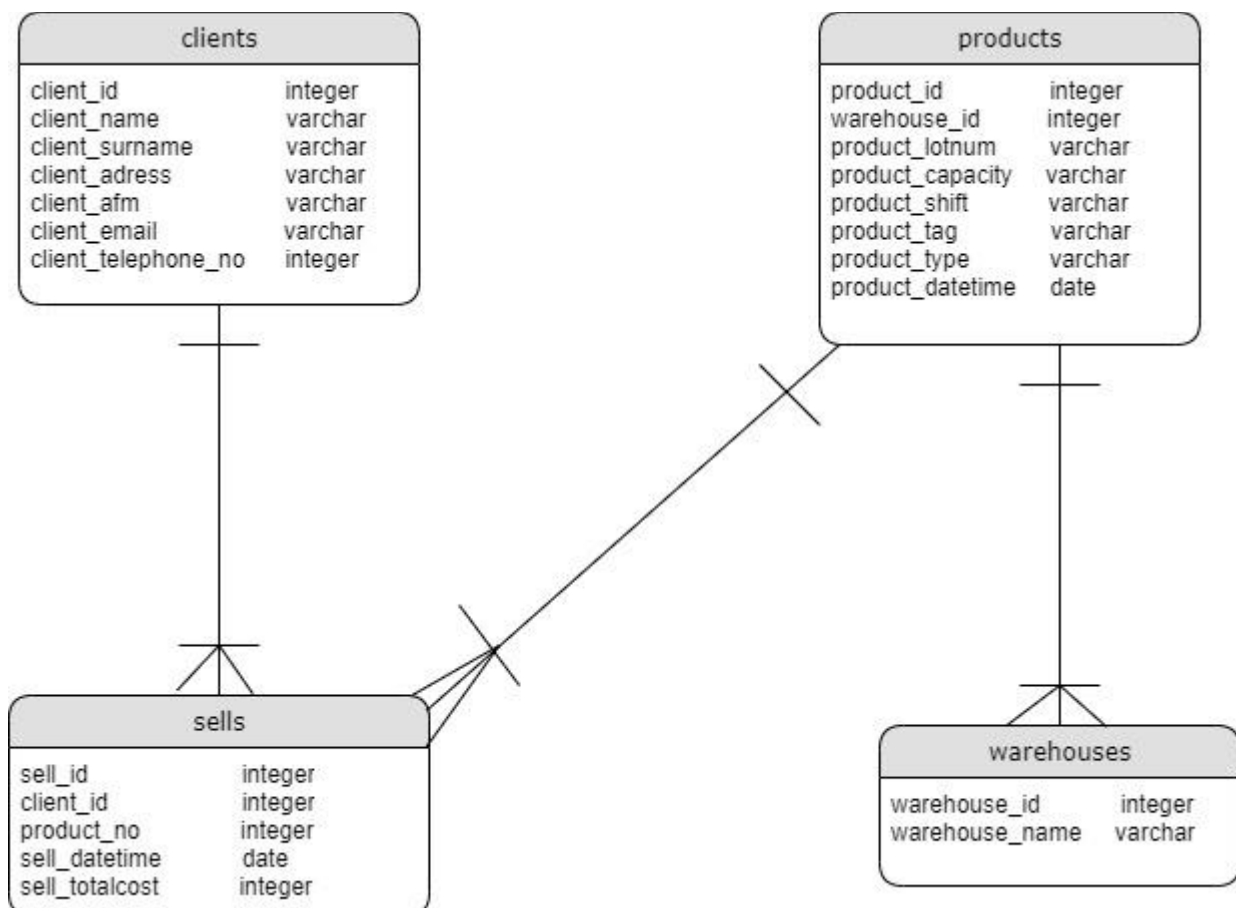
- SELL\_TOTAL\_COST: Είναι το συνολικό κόστος πώλησης και είναι τύπου (int). Το πεδίο αυτό δεν μπορεί να είναι κενό.

## Σχεσιακό μοντέλο

Λόγω των σχέσεων ένα προς πολλά μεταξύ των πινάκων της βάσης μας προστίθενται ως ξένα κλειδιά τα εξής:

- Η στήλη WAREHOUSE\_ID στον πίνακα warehouses ως ξένο κλειδί από τον πίνακα products.
- Η στήλη CLIENT\_ID στον πίνακα clients και η στήλη PRODUCT\_NO στον πίνακα products ως ξένο κλειδί από τον πίνακα sells.

Τα παραπάνω εμφανίζονται στο **Σχήμα 5.4**.



Σχήμα 5.4: Σχεσιακό μοντέλο

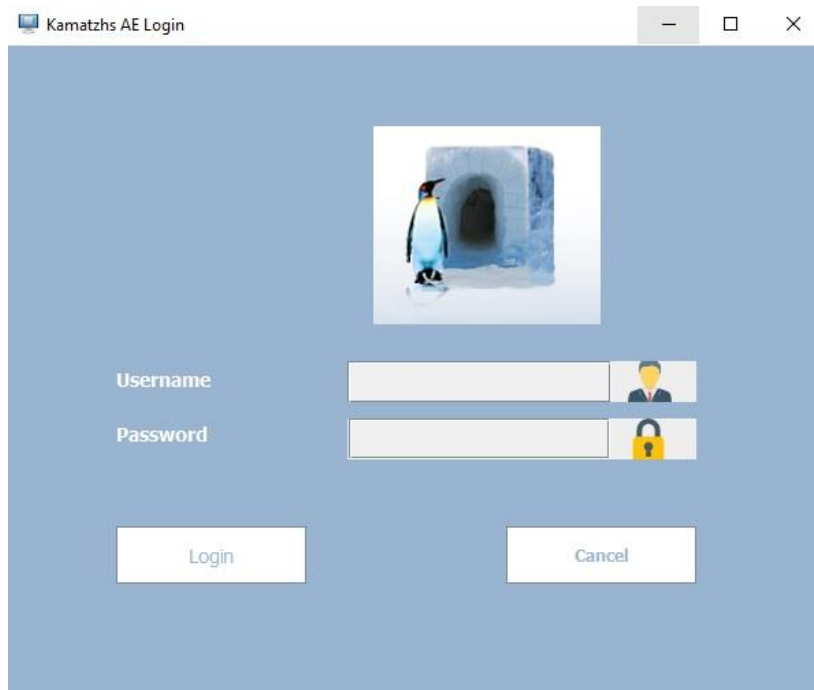
## Κεφάλαιο 6: Εγχειρίδιο χρήστη (User manual)

Το παρόν εγχειρίδιο περιγράφει το Πληροφοριακό Σύστημα διαχείρισης αποθήκης (WMS) της εταιρείας. Παρουσιάζονται αναλυτικά όλες οι πληροφορίες και τα βήματα που πρέπει να ακολουθήσει ο κάθε χρήστης, ώστε να υποβάλει ηλεκτρονικά μια επένδυση, καθώς επίσης και επιπλέον δυνατότητες που παρέχονται από το σύστημα.

Όσον αφορά στην δομή του εγχειριδίου, αρχικά περιγράφονται οι βασικές επιλογές (κοινές λειτουργίες) που υπάρχουν σε διάφορα σημεία της εφαρμογής, έτσι ώστε να αποκτήσει ο χρήστης τις απαραίτητες γνώσεις που απαιτούνται κατά την πλοήγησή του στις οθόνες. Στη συνέχεια, περιγράφεται η διαδικασία σύνδεσης του χρήστη στο σύστημα και τέλος η Αρχική Σελίδα της εφαρμογής με όλες τις επιλογές και τις λειτουργίες της.

### Είσοδος χρήστη στο σύστημα

Ο κάθε χρήστης χρησιμοποιεί το δικό του όνομα χρήστη (username) και κωδικό πρόσβασης (pass) για την είσοδό του στο σύστημα, όπως φαίνεται στην **εικόνα 6.1**



**Εικόνα 6.1: Είσοδος χρήστη στο σύστημα**

## Κοινές Λειτουργίες

Στον **πίνακα 6.1** γίνεται μία σύντομη περιγραφή των κοινών λειτουργιών που θα συναντήσει ο χρήστης του συστήματος κατά την περιήγησή του σε αυτό.

**Πίνακας 6.1: Κοινές Λειτουργίες ΠΣ**

Λειτουργία	Περιγραφή
<b>View</b>	Προβολή του πίνακα
<b>Add</b>	Πρόσθεση στον πίνακα
<b>Delete</b>	Διαγραφή από τον πίνακα
<b>Update</b>	Ενημέρωση στον πίνακα
<b>Refresh</b>	Ανανέωση του πίνακα
<b>Login</b>	Είσοδος στο Πληροφοριακό Σύστημα
<b>Cancel</b>	Έξοδος από το Πληροφοριακό Σύστημα
<b>Clean</b>	Καθαρισμός πίνακα
<b>Logout</b>	Αποσύνδεση από το Πληροφοριακό Σύστημα



## Υπάλληλος παραγωγής

Ο υπάλληλος παραγωγής αφότου εισέλθει στο σύστημα έχει πρόσβαση σε ένα μενού το οποίο περιλαμβάνει τις επιλογές:

- «*Product List*», όπου μπορεί να δει τα προϊόντα που είναι στην λίστα.
- «*Add Prouct*», για να πρόσθεση προϊόντα στη λίστα.
- «*Search*», για να αναζητήσει προϊόντα από τη λίστα.
- «*Refresh*», για να κάνει ανανέωση τη λίστα.
- «*Clean List*», για μπορεί να καθαρίσει τη λίστα.
- «*Logout*», για να αποσυνδεθεί από το σύστημα

## Υπάλληλος αποθήκης

Ο υπάλληλος αποθήκης αφότου εισέλθει στο σύστημα έχει πρόσβαση σε ένα μενού το οποίο περιλαμβάνει τις επιλογές:

- «*Product List*», όπου μπορεί να δει τα προϊόντα που είναι στην λίστα.
- «*Add Prouct*», για να πρόσθεση προϊόντα στη λίστα.
- «*Delete Product*», για να διαγράψει προϊόντα από τη λίστα.
- «*Update Product*», για να αλλάξει τα δεδομένα των προϊόντα από τη λίστα.
- «*Search*», για να αναζητήσει προϊόντα από τη λίστα.
- «*Refresh*», για να κάνει ανανέωση τη λίστα.
- «*Clean List*», για μπορεί να καθαρίσει τη λίστα.
- «*Logout*», για να αποσυνδεθεί από το σύστημα

## Καρτέλα «Products»

Μέσω της καρτέλας αυτής οι υπάλληλοι(παραγωγής, αποθήκης) έχουν πρόσβαση στα δεδομένα της αποθήκης και μπορούν να προσθέσουν, να αφαιρέσουν, να διαγράψουν και να αναζητήσουν προϊόντα. Τα παραπάνω φαίνονται στην εικόνα 6.2.

The screenshot shows a web application interface for managing products. At the top, there are navigation tabs: Clients, Products (selected), Stores, Sales, Prediction, and Logout. Below the tabs are four main action buttons: Product List, Add Product, Delete Prod..., and Update Prod....

On the left side, there are several input fields for filtering and searching:

- Product No: 9
- Store Id: 2
- Product Datetime: 2/2/2022
- Product Capacity: 125
- Product Type: NUGGET 4KG
- Production Shift: A
- Production Tag: CRAZY ICE
- Production Lotn...: B0322

At the bottom left, there are three buttons: Search, Refresh, and Clear List.

Product No	Store ID	Product Datetime	Product Capacity	Product Type	Production Shift	Product Tag	Product Lotnum
1	2	04/02/2022	150	SCOT 40 5KG	B	SUPER ICE	B0422
2	3	01/02/2022	100	SCOT 20 6KG	A	ΛΕΥΚΑ	B0122
3	2	01/02/2022	200	SCOT 20 4KG	A	ΒΟΛΟΣ	B0122
4	2	01/02/2022	200	NUGGET 6KG	B	ΘΕΣ ΠΑΓΟ	B0422
5	2	01/02/2022	200	ΑΥΔΙΚΟ 4KG	C	ΒΟΛΟΣ	B0622
6	2	04/02/2022	150	SCOT 40 5KG	B	SUPER ICE	B0422
7	2	01/02/2022	100	SCOT 20 7KG	B	ΝΕΟΣ ΚΟΣΜΟΣ	B0122
8	2	02/02/2022	200	SCOT 20 4KG	A	ΚΑΜΑΤΖΗΣ	B0222
9	2	02/02/2022	125	NUGGET 4KG	A	CRAZY ICE	B0322

Εικόνα 6.2: «Products»

## Υπάλληλος πωλήσεων

Ο υπάλληλος πωλήσεων αφότου εισέλθει στο σύστημα έχει πρόσβαση σε ένα μενού το οποίο περιλαμβάνει τις επιλογές:

### Στην Καρτέλα «Sales»

- «Sales List», όπου μπορεί να δει τις πωλήσεις που είναι στην λίστα.
- «Add Sales», για να πρόσθεση μια πώληση στη λίστα.
- «Delete Sales», για να διαγράψει μια πώληση από τη λίστα.
- «Update Sales», για να αλλάξει τα δεδομένα μιας πώληση από τη λίστα.
- «Search», για να αναζητήσει μια πώληση από τη λίστα.
- «Refresh», για να κάνει ανανέωση τη λίστα.
- «Clean List», για μπορεί να καθαρίσει τη λίστα.
- «Logout», για να αποσυνδεθεί από το σύστημα

Μέσω της καρτέλας αυτής οι υπάλληλοι πωλήσεων έχει πρόσβαση στις πωλήσεις και μπορούν να προσθέσουν, να αφαιρέσουν, να διαγράψουν και να αναζητήσουν και άλλες πωλήσεις. Τα παραπάνω φαίνονται στην εικόνα 6.3.

Sell Id	Product No	Client Id	Sell Datetime	Total Cost
1	1	1	28/10/2021	28.3
2	1	1	02/02/2022	28.3
3	2	2	02/02/2022	600.0
4	4	3	06/02/2022	500.0
5	5	4	07/02/2022	400.0
6	9	4	06/02/2022	800.0

Εικόνα 6.3: Sales

## Στην Καρτέλα «Clients»

- «*Clients List*», όπου μπορεί να δει τις πωλήσεις που είναι στην λίστα.
- «*Add Clients*», για να πρόσθεση μια πώληση στη λίστα.
- «*Delete Clients*», για να διαγράψει μια πώληση από τη λίστα.
- «*Update Clients*», για να αλλάξει τα δεδομένα μιας πώληση από τη λίστα.
- «*Search*», για να αναζητήσει μια πώληση από τη λίστα.
- «*Refresh*», για να κάνει ανανέωση τη λίστα.
- «*Clean List*», για μπορεί να καθαρίσει τη λίστα.
- «*Logout*», για να αποσυνδεθεί από το σύστημα

Μέσω της καρτέλας αυτής οι υπάλληλοι πωλήσεων έχει πρόσβαση στους πελάτες και μπορούν να προσθέσουν, να αφαιρέσουν, να διαγράψουν και να αναζητήσουν και άλλες πωλήσεις. Τα παραπάνω φαίνονται στην εικόνα 6.4.

Client ID	Client Name	Client Surname	Client AFM	Client Address	Client E-mail	Client Telephone Number
1	Δημήτρης	Κολάκας	1982016	Σύρου 482	dimitriskolakas@gmail.com	6983299910
2	Σταμάτης	Παπασταύρου	1798365	Μηνάτη 18-22	stamatisp@gmail.com	2106485540
3	Γεώργιος	Τερτίπης	1698365	Παπανδρέου Γεωργίου 150	tertipismarket@gmail.com	2221085540
4	Γεώργιος	Αλιμπινής	2198365	Φαρμακίδου 1	netfood@gmail.com	2221060436
5	Θανάσης	Βασιλείου	1657895	Ληλαντίων 105	basthanasis@yahoo.com	6985478624
6	Βάσιλης	Θεωδώρα	1655569	Ληλαντίων 26	leto@yahoo.com	69856588745

Εικόνα 6.4: Clients

## Υπεύθυνος παραγωγής

Ο υπάλληλος παραγωγής αφότου εισέλθει στο σύστημα έχει πρόσβαση σε όλες τις καρτέλες και συγκεκριμένα σε ένα μενού το οποίο περιλαμβάνει τις επιλογές:

### Στην Καρτέλα «Prediction»

- «Refresh», για να κάνει ανανέωση τη λίστα.
- «Logout», για να αποσυνδεθεί από το σύστημα

Τα παραπάνω φαίνονται στην εικόνα 6.5.

Sell id	Product No	Client id	Sell Datetime	Total Cost
2	1	1	02/02/2022	28.3
3	2	2	02/02/2022	600.0
4	4	3	08/02/2022	500.0
5	5	4	07/02/2022	400.0
6	9	4	06/02/2022	800.0
7	8	6	08/02/2022	650.0

Εικόνα 6.5: Prediction

## Στην Καρτέλα «Store»

- «Store List», όπου μπορεί να δει τις πωλήσεις που είναι στη λίστα.
- «Add Store», για να πρόσθεση μια πώληση στη λίστα.
- «Delete Store», για να διαγράψει μια πώληση από τη λίστα.
- «Update Store», για να αλλάξει τα δεδομένα μιας πώληση από τη λίστα.
- «Search», για να αναζητήσει μια πώληση από τη λίστα.
- «Refresh», για να κάνει ανανέωση τη λίστα.
- «Clean List», για μπορεί να καθαρίσει τη λίστα.
- «Logout», για να αποσυνδεθεί από το σύστημα.

Store ID	Store Name
3	Μπροστά Αποθήκη
2	Πίσω Αποθήκη
4	Νέα αποθήκη

Εικόνα 6.6: Store





## Βιβλιογραφικές παραπομπές

1. Βασιλακόπουλος, Γ. (2018). *Πληροφοριακά Συστήματα*. Αθήνα: Εκδόσεις Τσότρας
2. <https://netweek.gr/%CF%84%CE%B9-%CE%B5%CF%83%CF%84%CE%AF-devops/> [07 Ιανουαρίου 2022]
3. Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd edition)*. Boston: Addison-Wesley.
4. Albrecht, A. (1979). *Measuring Application Development Productivity*. New York: IBM Corporation.
5. Brennan, M. (1968). *PERT and CPM: a selected bibliography*. Monticello, Ill: Council of Planning Librarians.
6. Clark, W. (1922). *The Gantt Chart: A working tool of management*. New York: The Ronald Press Company.
7. <https://techacute.com/what-is-devops/> [05 Φεβρουαρίου 2022]
8. <https://www.opengroup.org/> [07 Φεβρουαρίου 2022]
9. <https://www.ifpug.org/wp-content/uploads/2017/04/IYSM.-Thirty-years-of-IFPUG.-Software-Economics-and-Function-Point-Metrics-Capers-Jones.pdf> [3 Φεβρουαρίου 2022]
10. stonith.gr/?p=2852 <https://www.stonith.gr/?p=2852> [4 Φεβρουαρίου 2022]
11. Βασιλακόπουλος, Γ. (2016). *Σχεδιασμός Βάσεων Δεδομένων*. Πειραιάς: Γεώργιος Βασιλακόπουλος