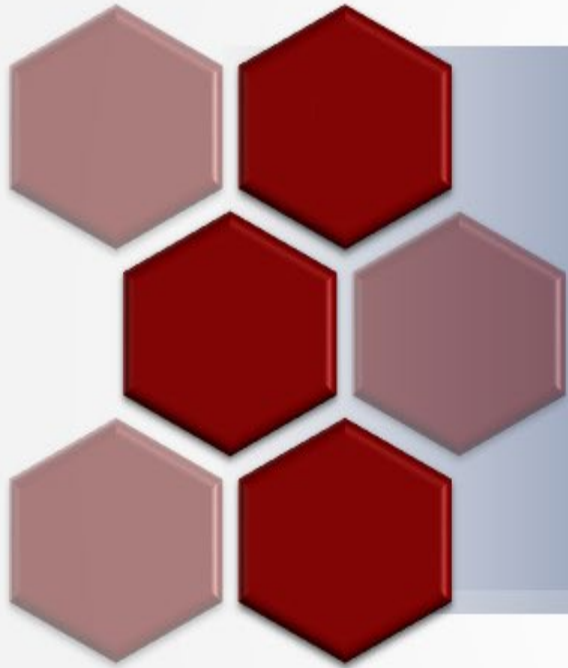




ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ



# Πληροφορικά Συστήματα

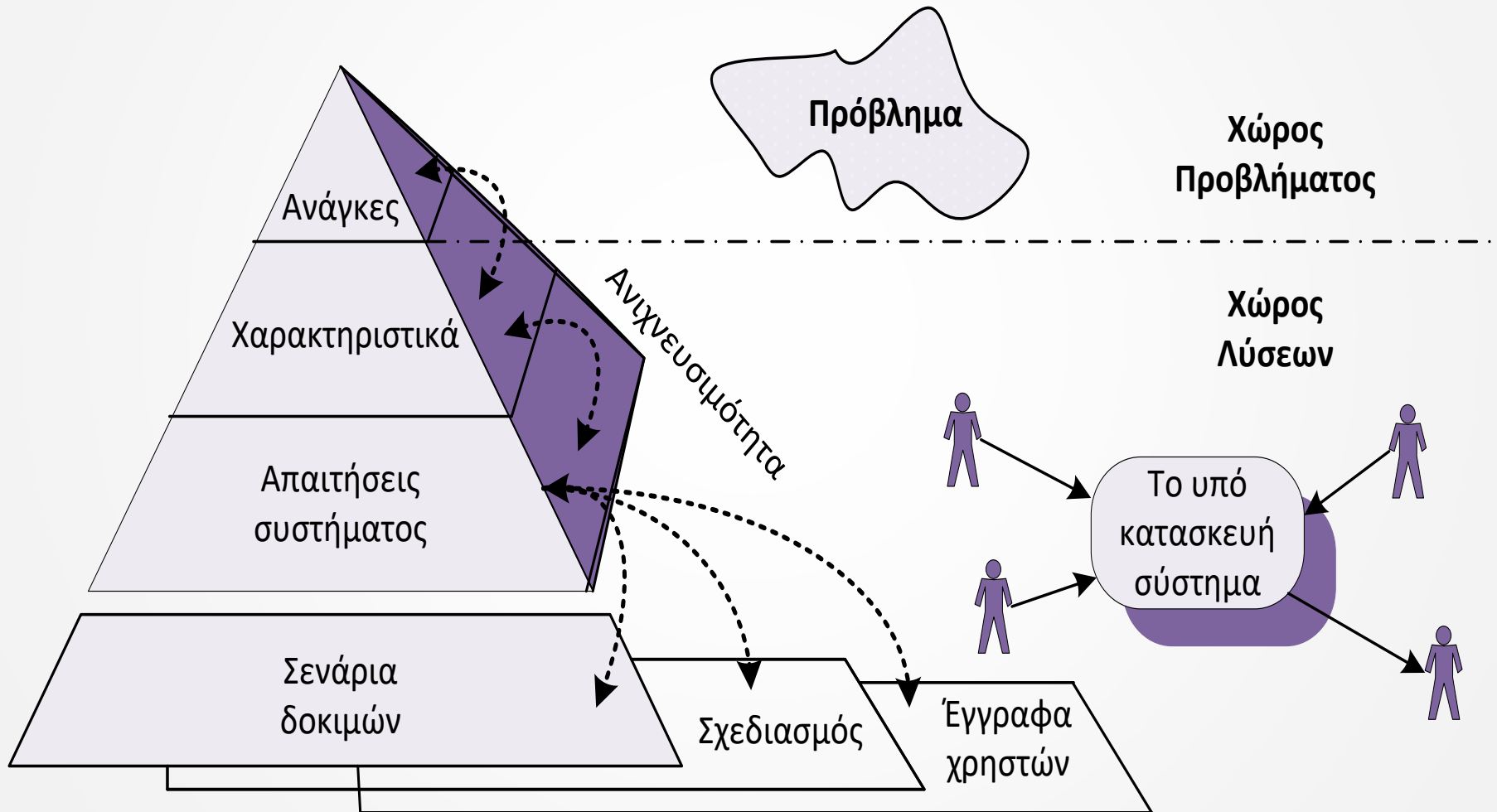
- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Η μεθοδολογία RUP αποτελεί ένα από τα βασικά εργαλεία ανάπτυξης συστημάτων της εταιρείας IBM (αρχικά της, εξαγορασθείσης από την IBM, εταιρείας *RATIONAL*) και συνιστά μια από τις πιο διαδεδομένες και ευρέως χρησιμοποιούμενες μεθοδολογίες παγκοσμίως.
- Η μεθοδολογία RUP διαθέτει τα ακόλουθα βασικά χαρακτηριστικά:
  - Ακολουθεί το μοντέλο των εξελικτικών (επαυξητικών) μεθοδολογιών ανάπτυξης συστημάτων βασιζόμενη στη σταδιακή και επαναληπτική ανάπτυξη του συστήματος.
  - **Χρησιμοποιεί τη UML**, η οποία παρέχει μια πλούσια συλλογή διαγραμμάτων για την μοντελοποίηση διαφόρων θεωρήσεων του συστήματος (λειτουργικών, στατικών και δυναμικών), γεγονός το οποίο της προσδίδει το πρόσθετο πλεονέκτημα της παραγωγής τεκμηριωτικού υλικού σε ενιαία γλώσσα.

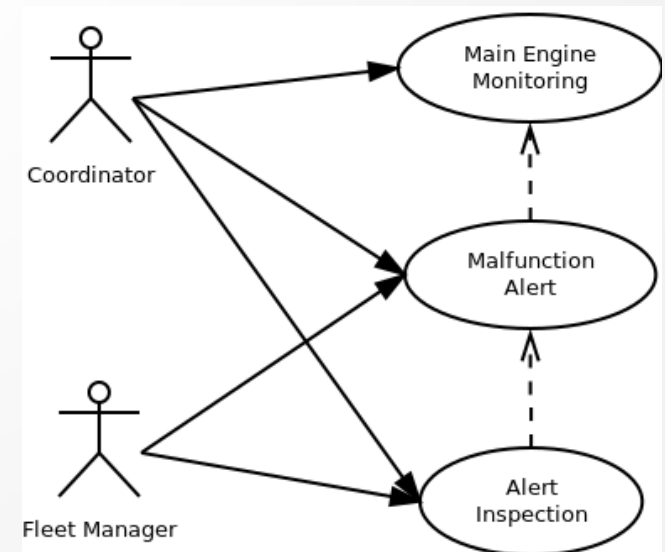
- Η RUP αποτελείται από τέσσερις κύριες φάσεις οι οποίες εκτελούνται διαδοχικά:
  - Σύλληψη (*Inception*),
  - Εντρύφηση (*Elaboration*),
  - Κατασκευή (*Construction*) και
  - Μετάβαση (*Transition*).
- Κάθε φάση έχει συγκεκριμένο σκοπό και περατώνεται σε ένα ορόσημο που σηματοδοτεί το γεγονός (*event*) και το χρονικό σημείο (*time point*) στο οποίο επιτυγχάνεται μια συγκεκριμένη πρόοδος του έργου.
- Γενικά, η μεθοδολογία RUP μπορεί να θεωρηθεί ως ενδιάμεση λύση μεταξύ της καταρρακτοειδούς μεθοδολογίας και των μεθοδολογιών ευέλικτης ανάπτυξης συστημάτων.

Οι διαδικαστικές ροές εργασίας είναι:

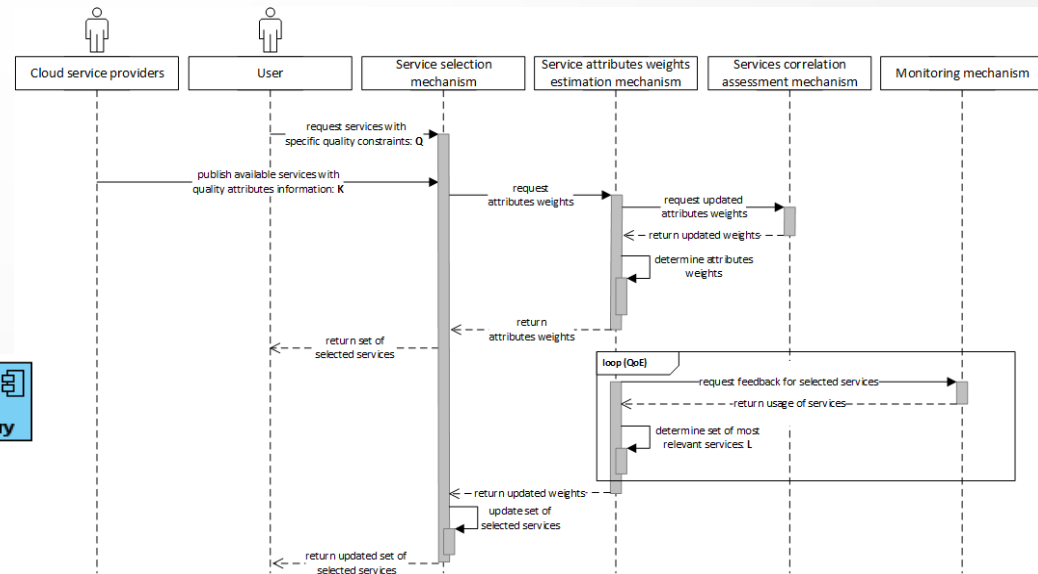
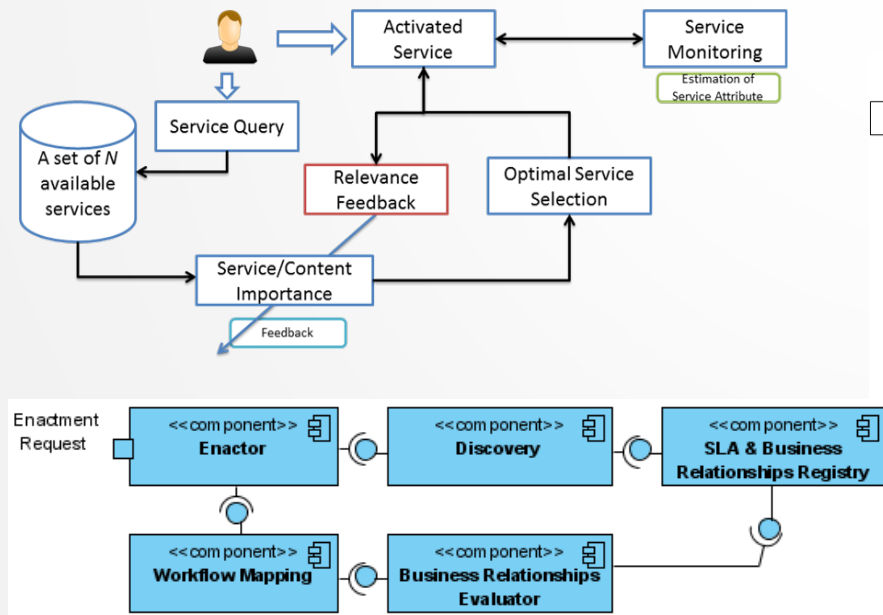
- Η μοντελοποίηση του επιχειρησιακού περιβάλλοντος (*Business Modeling*),
- Ο προσδιορισμός των απαιτήσεων (*Requirements Specification*),
- Ο αρχιτεκτονικός σχεδιασμός του συστήματος (*System Architecture and Design*),
- Η πραγμάτωση του συστήματος (*System Implementation*),
- Ο έλεγχος του συστήματος αναφορικά με την ικανοποίηση των απαιτήσεων (*System Testing*), και
- Η εγκατάσταση στο σύστημα στο παραγωγικό περιβάλλον (*System Deployment*).



- i. **Καθοδηγούμενη από περιπτώσεις χρήσης** – Κατασκευάζονται σενάρια από τα οποία αντλούνται και συνάγονται οι λειτουργικές και τεχνικές απαιτήσεις του συστήματος. Τα σενάρια αυτά κατασκευάζονται χρησιμοποιώντας την τεχνική των διαγραμμάτων περιπτώσεων χρήσης (*use-case diagrams*) της UML και μπορεί να περιγράφουν επιθυμητές ή/και ανεπιθύμητες ακολουθίες γεγονότων. Οι περιπτώσεις χρήσης χρησιμοποιούνται ως βασικά στοιχεία για τον καθορισμό της συμπεριφοράς, την επαλήθευση της αρχιτεκτονικής, τον έλεγχο και την επικοινωνία μεταξύ των εμπλεκομένων στην ανάπτυξη του συστήματος.



- ii. **Αρχιτεκτονικο-κεντρική** – Ακολουθείται μια διεργασία ανάπτυξης συστημάτων που βασίζεται στην αρχιτεκτονική του συστήματος. Η αρχιτεκτονική του συστήματος χρησιμοποιείται ως βασικό μέσο για την κατανόηση, την κατασκευή, τη διαχείριση και την εξέλιξη του υπό κατασκευή συστήματος οδηγώντας στην προδιαγραφή (*specification*), στην κατασκευή (*construction*) και στην τεκμηρίωση (*documentation*) αυτού. Για το σκοπούς αυτούς πρέπει να υποστηρίζονται τουλάχιστον τρεις διαφορετικές όψεις της αρχιτεκτονικής: η λειτουργική (*functional - conceptual*), η δομική ή στατική (*structural or static = component model*) και η δυναμική ή συμπεριφορική (*dynamic or behavioral*) με αντίστοιχες διαγραμματικές τεχνικές που περιλαμβάνονται στην UML.

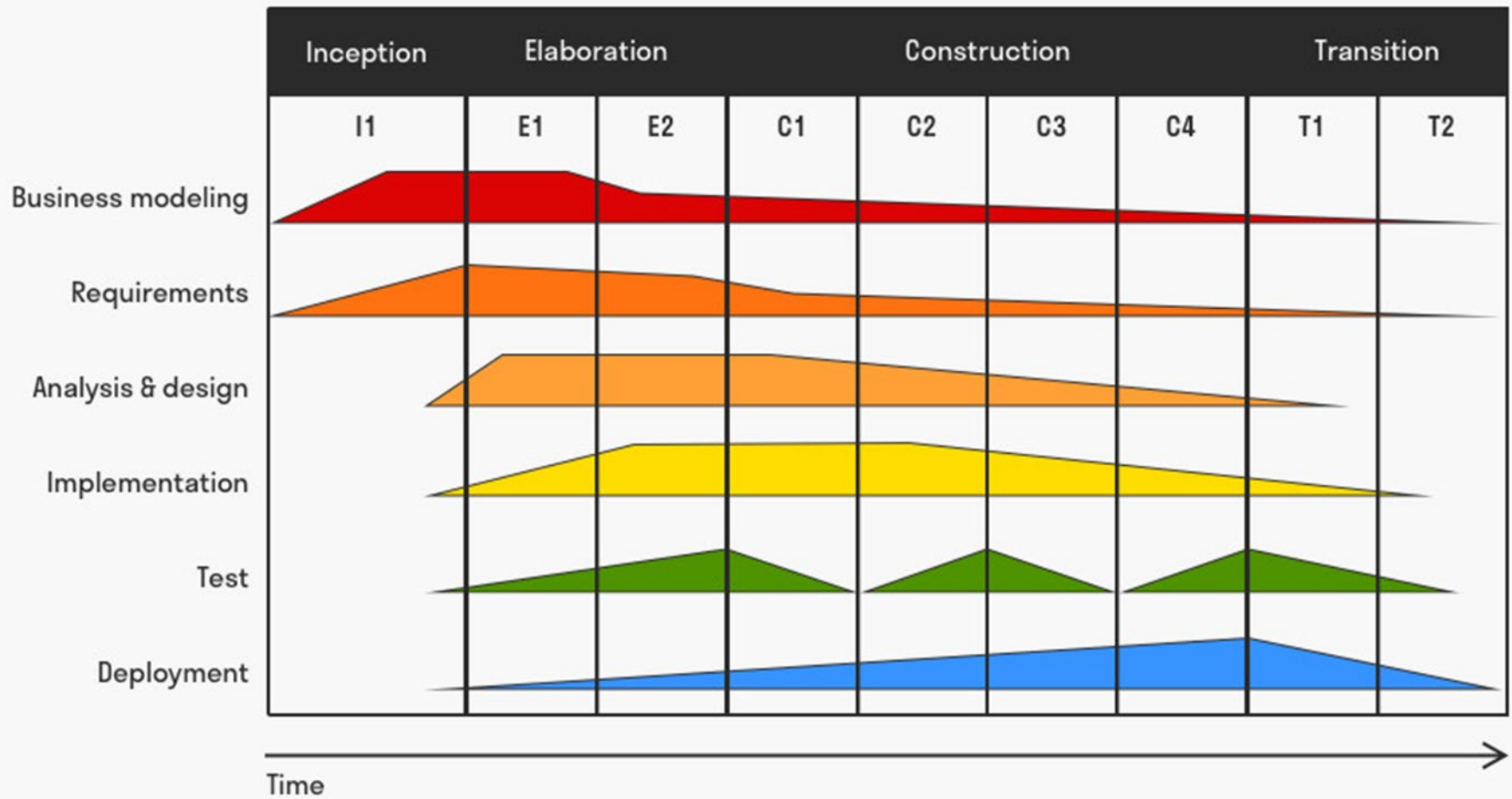




**iii. Επαναληπτική και επαυξητική:** Η ακολουθούμενη διεργασία ανάπτυξης συστημάτων είναι **επαναληπτική** και **επαυξητική**.

- Η επαναληπτική διεργασία αφορά στη διαχείριση των εκτελέσιμων εκδόσεων ενός συστήματος και
- η επαυξητική διεργασία αφορά στη συνεχή ολοκλήρωση της αρχιτεκτονικής για την παραγωγή των εκτελέσιμων εκδόσεων του συστήματος.

Το σύστημα κατασκευάζεται κατά διαδοχικές εκδόσεις, κάθε μία από τις οποίες διαθέτει βελτιωμένα χαρακτηριστικά, και δεν παρουσιάζεται ως τετελεσμένο στο χρονικό πέρας του έργου.



## SCRUM



## RUP

## ➤ Scope of the project

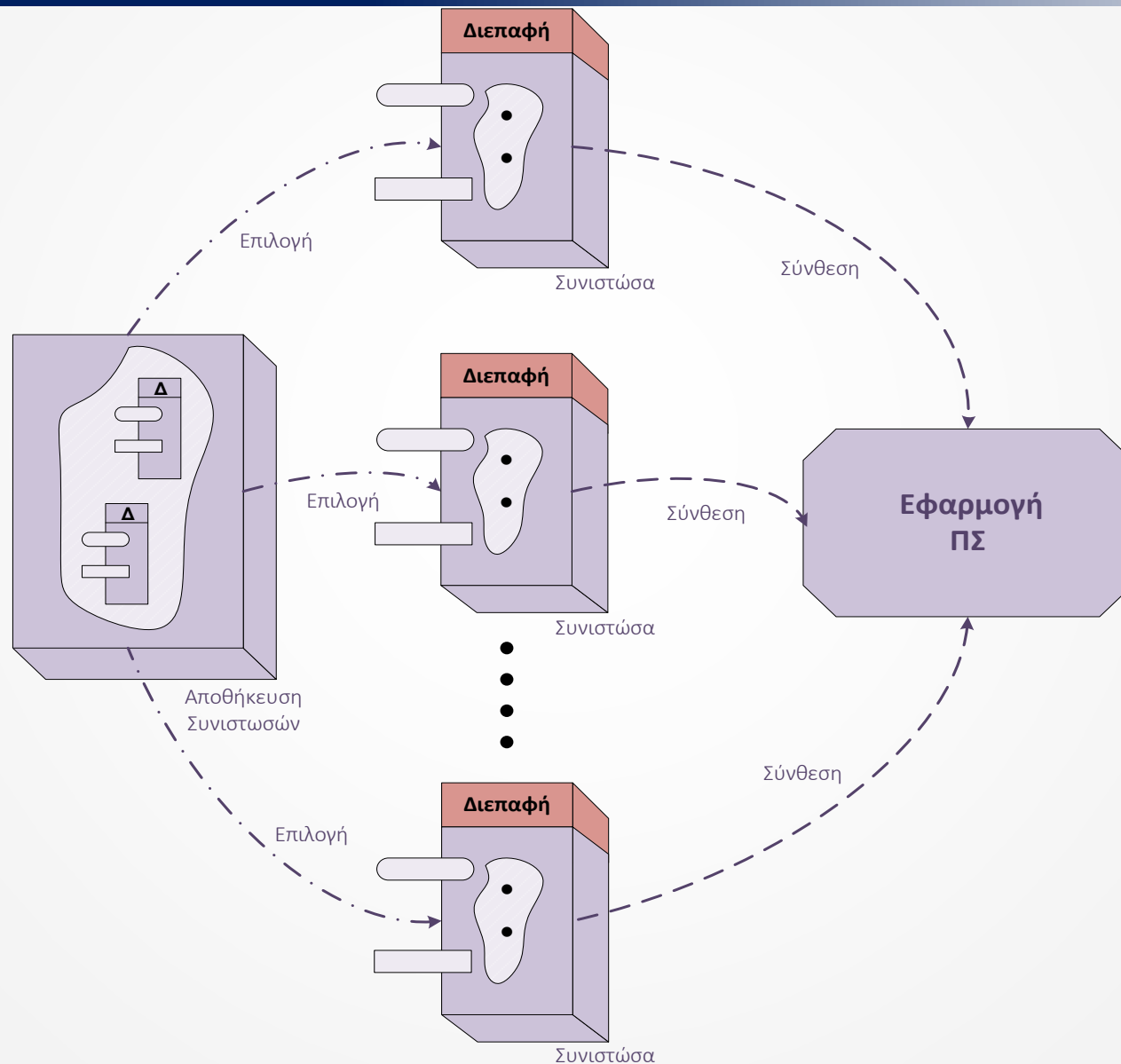
- ▶ RUP: Specific (inception phase) -> will not change
- ▶ SCRUM: Flexible -> might change

## ➤ Planning of the project

- ▶ RUP: Well-specified (milestones) with iterations
- ▶ SCRUM: Much more dynamic (product owner decide when something is completed) with iterations

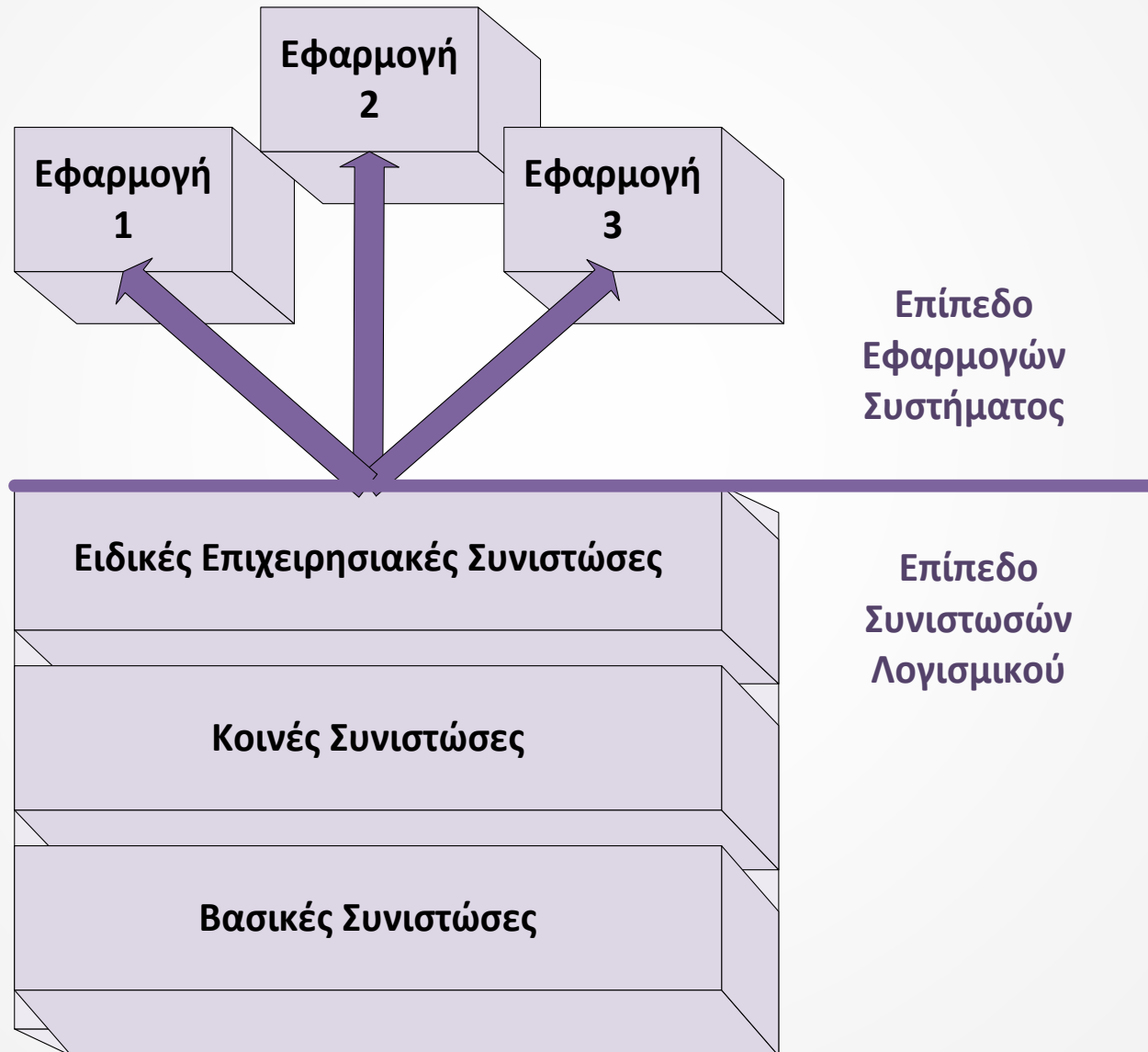
## ➤ Cycle

- ▶ RUP: 4 phases with various overlapping processes (e.g. requirements, analysis, implementation, etc) for potentially different components
- ▶ SCRUM: Each cycle is “atomic” / “complete” per component



Τα βασικά (και ευκταία) χαρακτηριστικά της ανάπτυξης ΠΣ κατά συνιστώσες είναι:

- Ότι οι συνιστώσες πρέπει να είναι ανεξάρτητες μεταξύ τους και να προσδιορίζονται μόνο μέσω των διεπαφών τους.
- Ότι οι συνιστώσες πρέπει να βασίζονται σε διεθνή πρότυπα προκειμένου να διευκολύνεται η ολοκλήρωση/διαλειτουργικότητα μεταξύ τους.
- Ότι η ανάπτυξη ΠΣ κατά συνιστώσες συνιστά μια διεργασία ανάπτυξης συστημάτων λογισμικού που αναφέρεται ευθέως στην ιδέα της επαναχρησιμοποίησης του υπάρχοντος λογισμικού.



## App1: eClass

Submit  
grades



Login /  
authentication

Business  
component:  
Submit grades

Business  
component:  
Upload PPT

## App2: Evdoxos

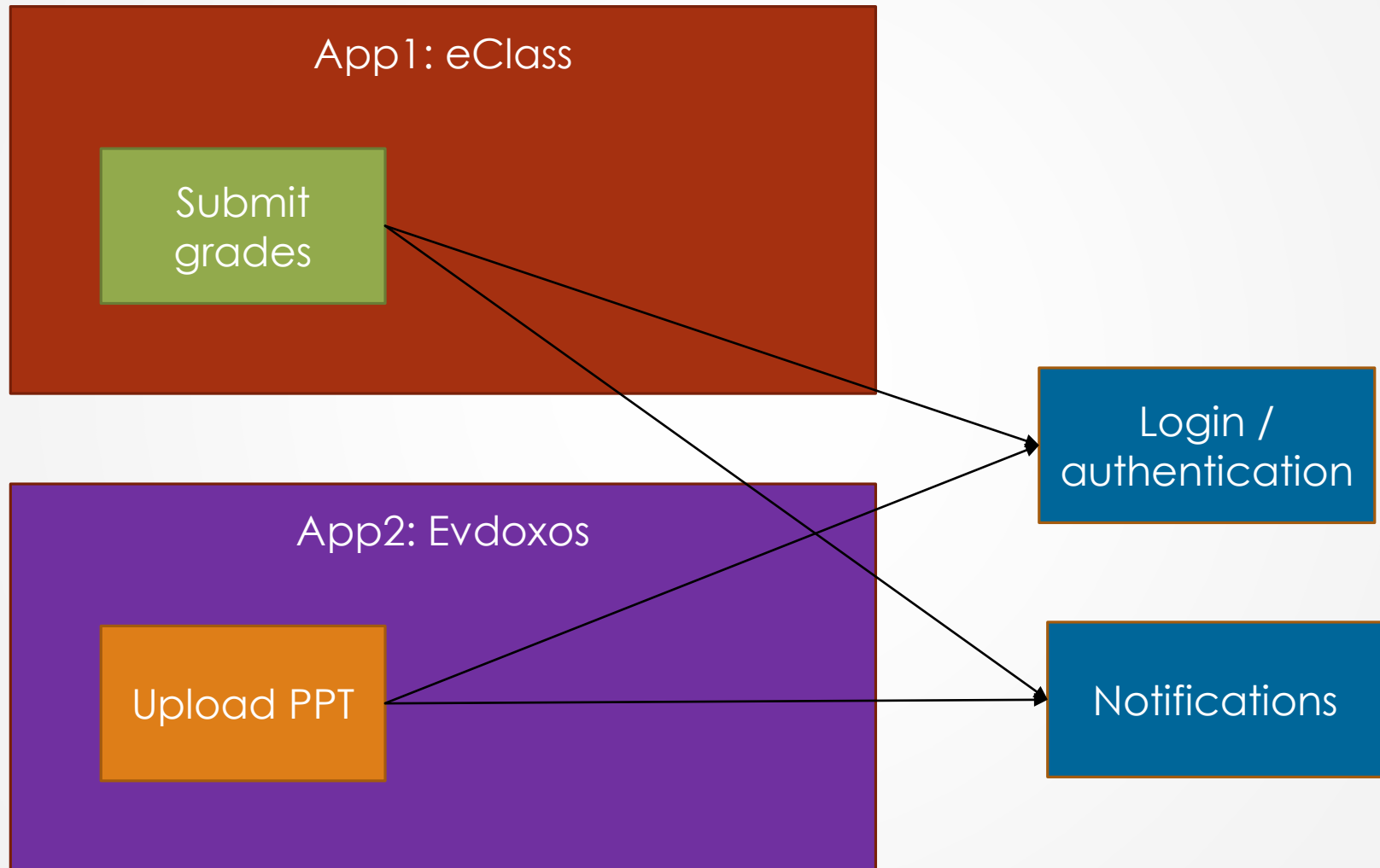
Upload PPT



Login /  
authentication

Common  
component:  
Login /  
authentication

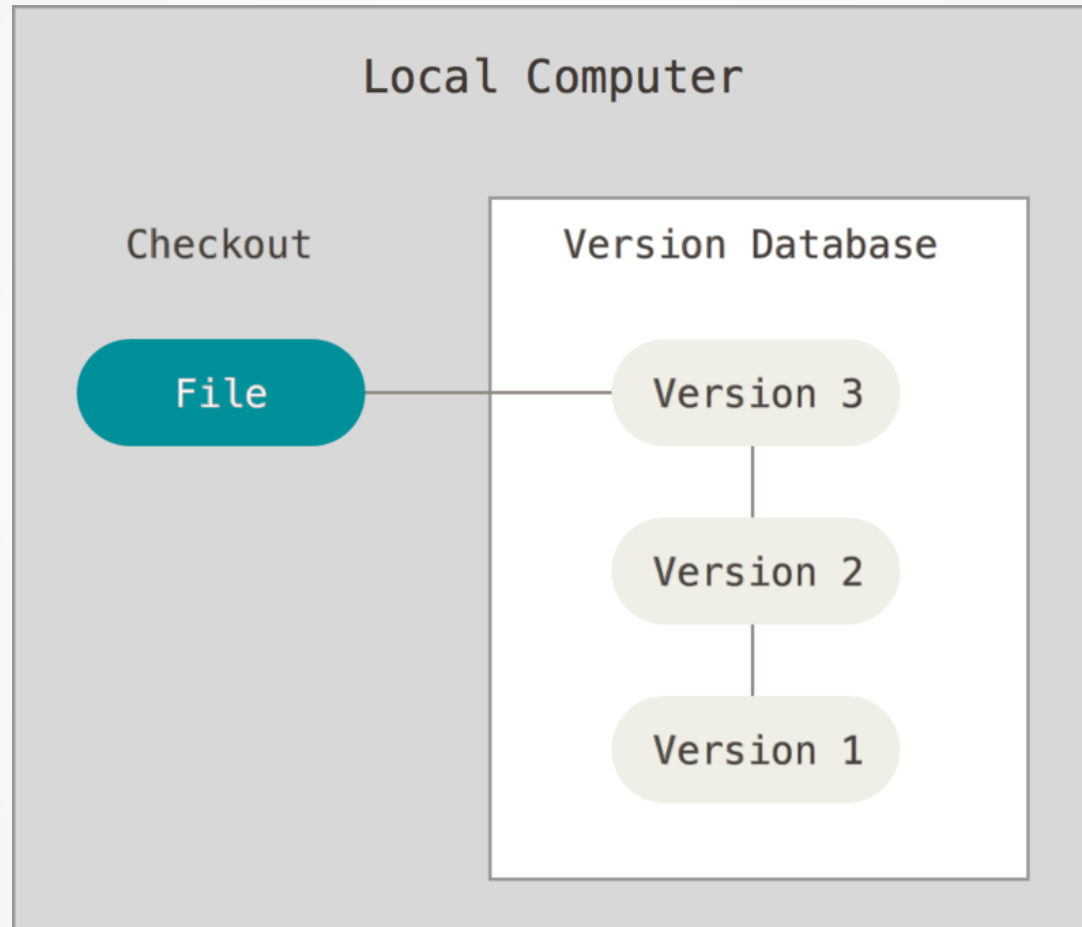




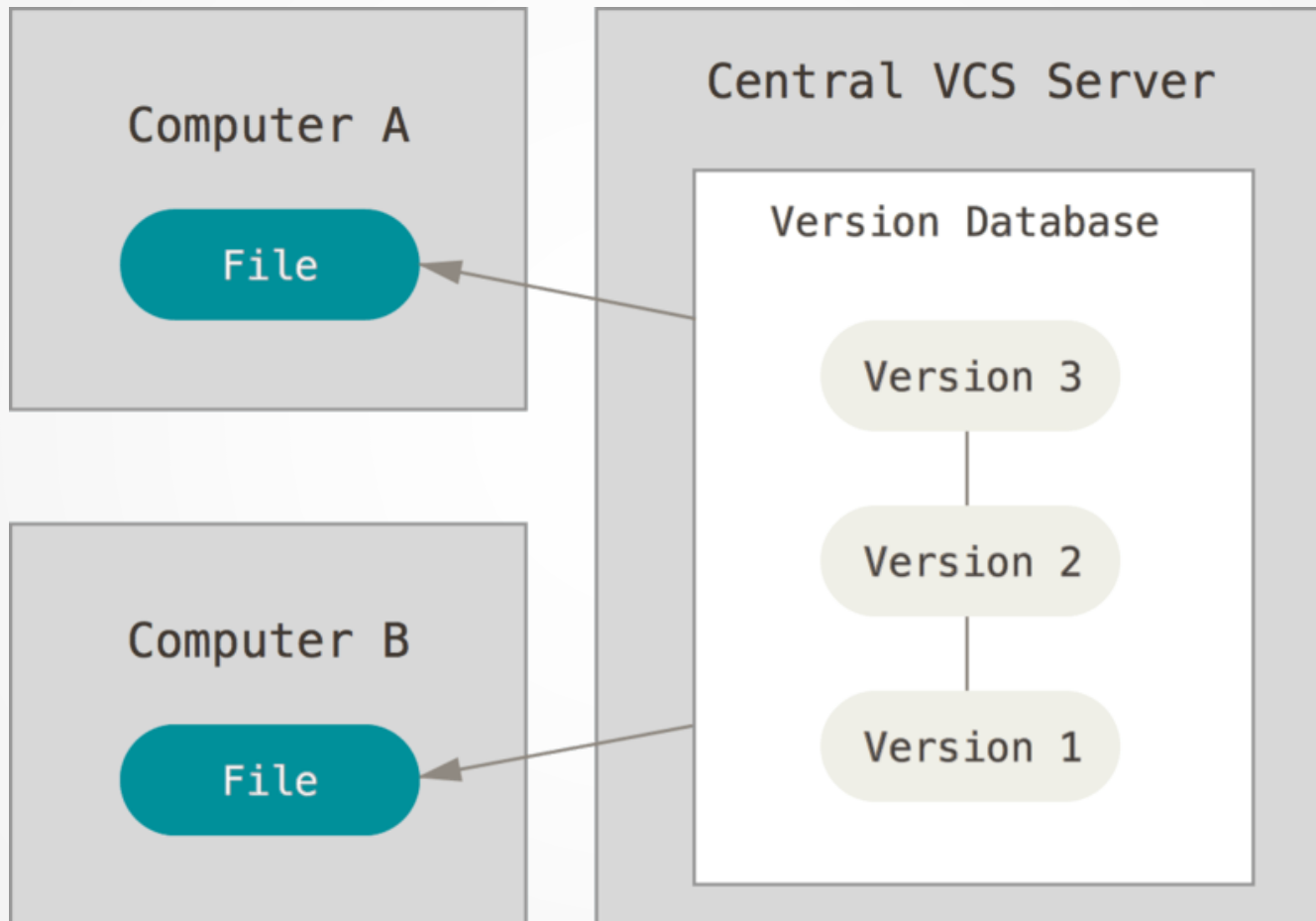
# Version Control

- Version control: Σύστημα καταγραφής αλλαγών σε 1 ή περισσότερα αρχεία ώστε να είναι εφικτή η μελλοντική ανάκτηση εκδόσεων
- Τύποι
  - ▶ Local
  - ▶ Centralized
  - ▶ Distributed
- Παραδείγματα
  - ▶ CVS
  - ▶ Subversion
  - ▶ Git

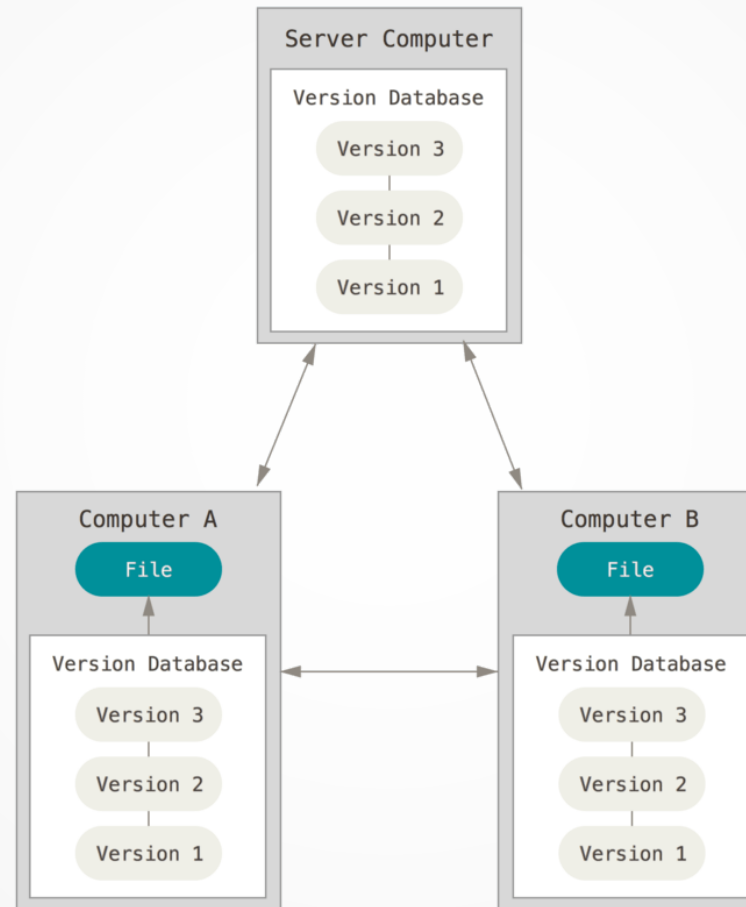
# Local Version Control Systems



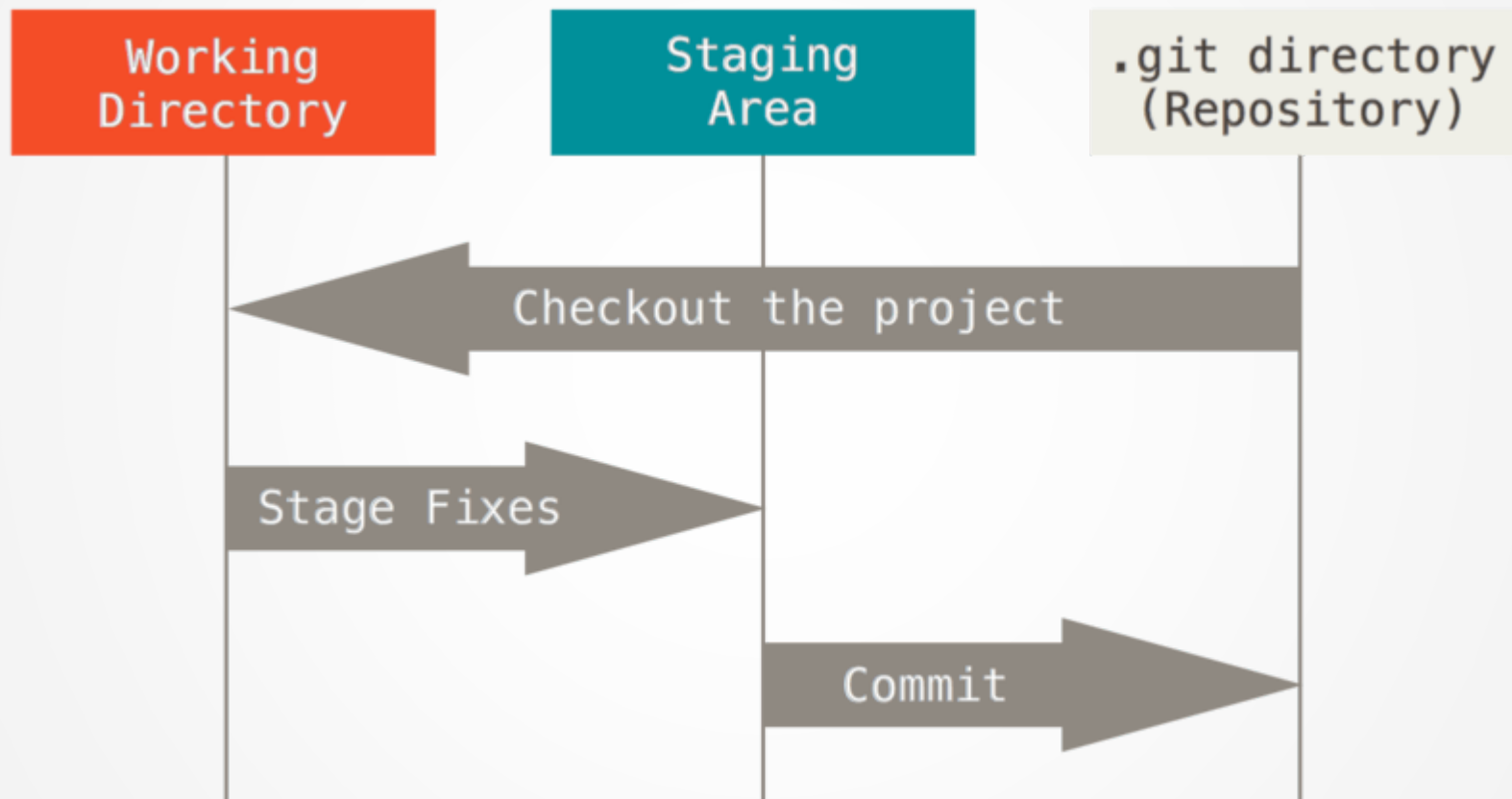
# Centralized Version Control Systems + LOCKING!



# Distributed Version Control Systems



## Git States & Git repositories for developers (i.e. development phase)



Docker (and containers and Docker Hubs / repositories) are for administrators (i.e. operations phase)

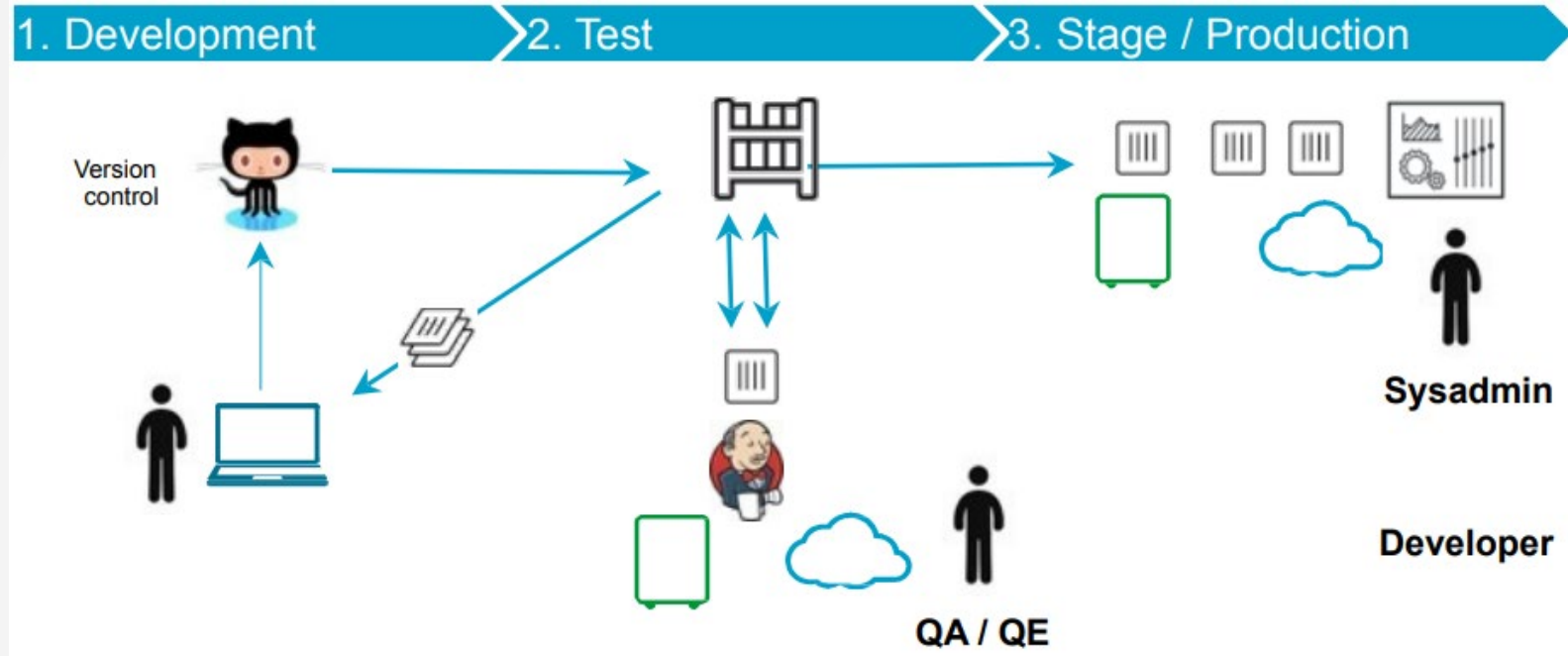
- Ως “**υπηρεσία**” ορίζεται μια λογική αναπαράσταση μιας επιχειρησιακής δραστηριότητας που επαναλαμβάνεται και παράγει συγκεκριμένα αποτελέσματα (π.χ. έλεγχος του πιστωτικού υπολοίπου ενός πελάτη τράπεζας, παραγωγή μηνιαίων αναφορών πωλήσεων).
  - Είναι αυτόνομη και αυτοδύναμη
  - Μπορεί να αποτελείται από άλλες υπηρεσίες
  - Είναι “μαύρο κουτί” για τους καταναλωτές της υπηρεσίας
- Αλλιώς, “**υπηρεσία**” είναι μια λειτουργία ή επιχειρησιακή διεργασία ενός οργανισμού η οποία είναι καλά ορισμένη, είναι αυτοδύναμη και δεν εξαρτάται από το περιβάλλον ή την κατάσταση άλλων υπηρεσιών.

# DevOps

- Σύμφωνα με ένα μοντέλο DevOps, οι ομάδες ανάπτυξης και λειτουργίας δεν είναι πλέον απομονωμένες
  - ▶ Συνήθως οι δύο αυτές ομάδες **συγχωνεύονται** σε μία ομάδα όπου οι μηχανικοί εργάζονται σε ολόκληρο τον κύκλο ζωής της εφαρμογής, από την ανάπτυξη και δοκιμή μέχρι την ανάπτυξη σε λειτουργίες, δεξιοτήτων που δεν περιορίζονται σε μία μόνο λειτουργία
- Αυτές οι ομάδες χρησιμοποιούν πρακτικές για την αυτοματοποίηση διαδικασιών
  - ▶ **Developer: code, push git, docker build**
    - Tester: integrate (Jenkins), testing (junit), bug reporting (gitlab)
  - ▶ **Operations: get git, docker build, docker push (production environment)**
- Χρησιμοποιούν μια στοίβα τεχνολογίας και εργαλεία που τους βοηθούν να λειτουργούν και να εξελίσσονται εφαρμογές γρήγορα και αξιόπιστα
- Όταν η ασφάλεια είναι η εστίαση όλων σε μια ομάδα DevOps, αυτό μερικές φορές αναφέρεται ως **DevSecOps**



# Continuous Integration and Delivery



App1: eClass

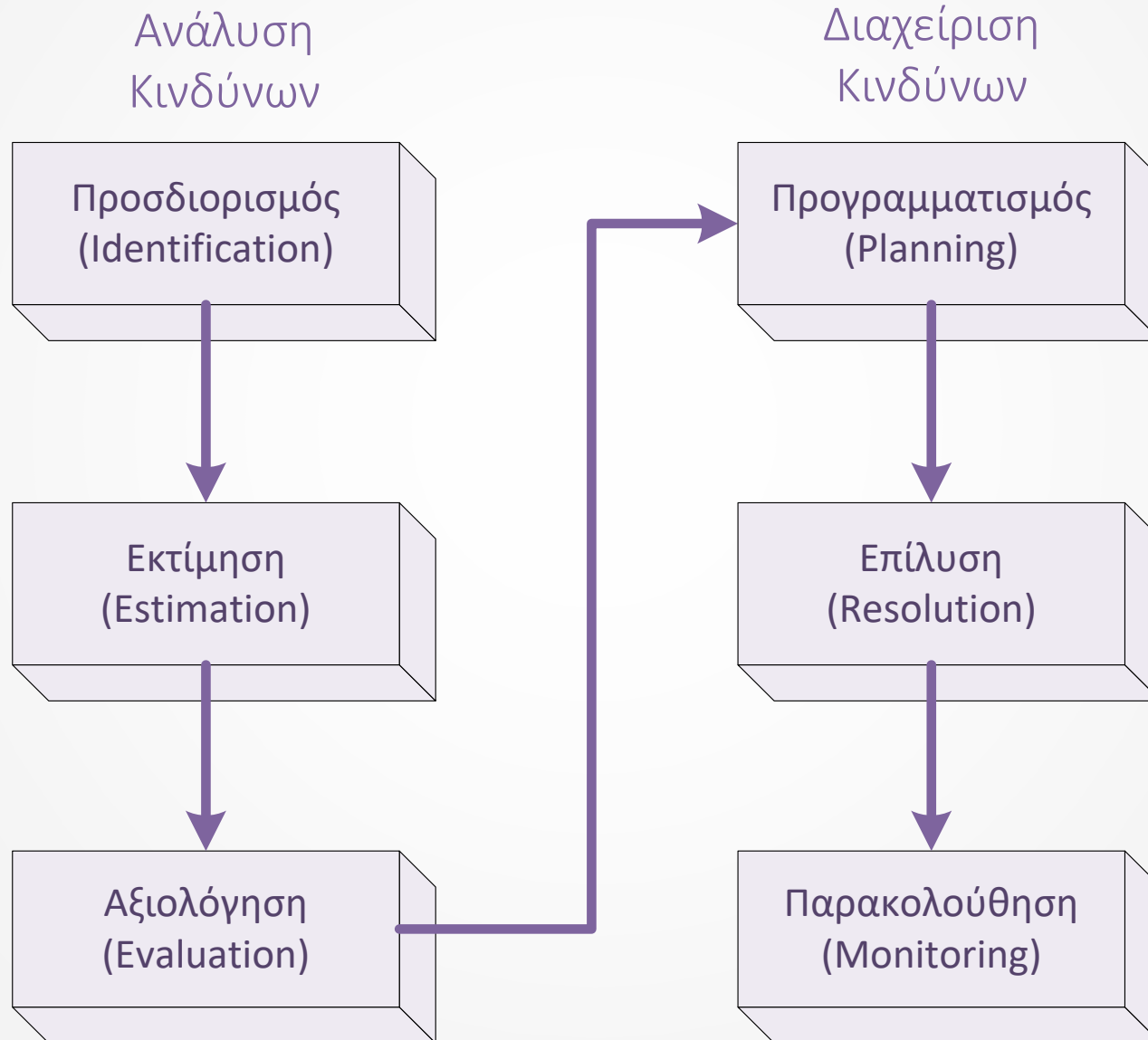
Submit  
grades

Login /  
authentication

- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

Η επιλογή μιας μεθοδολογίας ανάπτυξης ΠΣ γίνεται με γνώμονα:

- Την αύξηση της εκ των προτέρων (a priori) πιθανότητας επιτυχίας του συστήματος με βάση μετρήσιμα κριτήρια, όπως:
  - Ο προσδοκώμενος βαθμός αξιοποίησής του από τους χρήστες ή
  - Το προσδοκώμενο όφελος (χρηματικό, κτλ) για τον οργανισμό από τη χρήση του.
- Τη μείωση της εκ των προτέρων (a priori) πιθανότητας αποτυχίας του συστήματος η οποία ορίζεται ως το μέτρο της αβεβαιότητας για το τελικό αποτέλεσμα (όποιο και αν θεωρηθεί ως τελικό αποτέλεσμα).



Οι κίνδυνοι που αναφέρονται στην ανάπτυξη και πραγμάτωση ενός ΠΣ είναι διαφόρων ειδών και μπορούν να διακριθούν σε δύο κατηγορίες:

## **I. Επιχειρηματικοί κίνδυνοι**

- Νομοθεσία (απηρχαιωμένη, συχνά μεταβαλλόμενη)
- Δράσεις ανταγωνιστών
- Εγκυρότητα και βιωσιμότητα του επιχειρηματικού σχεδίου του οργανισμού
- Δυνάμεις αγοράς
- Υποστήριξη από το έργο της επιχειρηματικής και ψηφιακής στρατηγικής του οργανισμού.
- Πολιτικοί παράγοντες (κοινή γνώμη, κυβερνητικές παρεμβολές, κτλ)
- Ικανοποίηση των προσδιορισμένων απαιτήσεων από το έργο αλλά μη ικανοποίηση των προσδοκιών των πελατών.

## II. Κίνδυνοι διοίκησης έργου

- Κίνδυνοι λόγω του ανθρώπινου παράγοντα
  - Ανεπιτυχής παράδοση του έργου από τον προμηθευτή
  - Εμπειρία του προμηθευτή στον επιχειρηματικό τομέα αναφοράς
  - Έλλειψη επαρκούς εμπειρίας από το προσωπικό ανάπτυξης του συστήματος του προμηθευτή
  - Έλλειψη εμπειρίας από την ομάδα διοίκησης του έργου
  - Συμβατικοί κίνδυνοι
  - Διαθεσιμότητα πόρων
  - Έλλειψη δεξιοτήτων
  - Σχέση μεταξύ πελάτη και προμηθευτή.

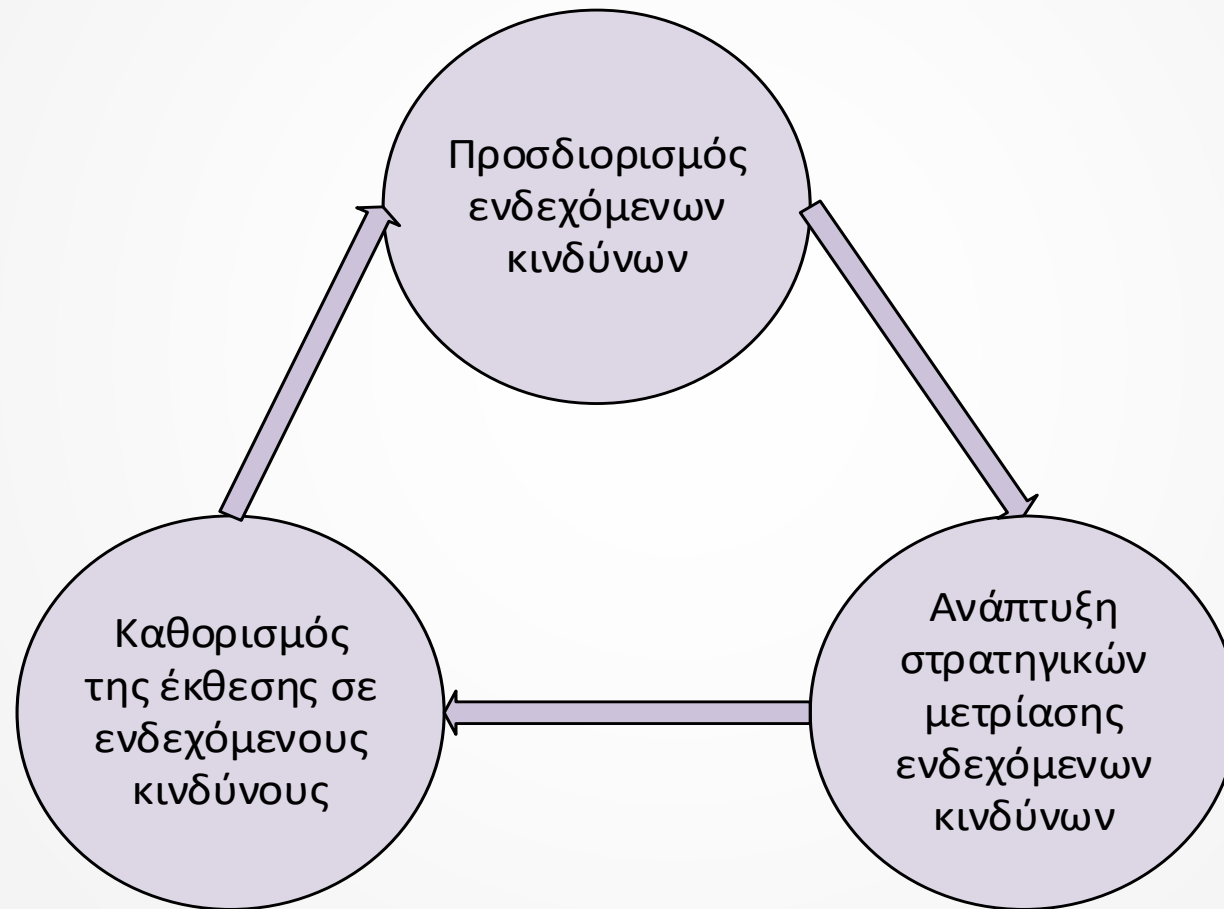
- Κίνδυνοι προγραμματισμού
  - Ανεπιτυχής προσδιορισμός απαιτήσεων
  - Μεγάλο μέγεθος του έργου
  - Ανεπιτυχής εκτίμηση κόστους
  - Ανεπιτυχής προσδιορισμός των ακριβών απαιτήσεων
  - Αnéφικτες ή ακατόρθωτες απαιτήσεις
  - Ανεπιτυχείς δοκιμές
  - Παραδοτέα που υπολείπονται των προτύπων ποιότητας.

- Τεχνικοί κίνδυνοι

- Ανάπτυξη τεχνολογικά προηγμένης ή πολύπλοκης λύσης
- Έλλειψη τεχνικής εμπειρίας της ομάδας έργου
- Γεωγραφικά κατανεμημένη ανάπτυξη
- Χρήση τεχνολογιών αιχμής.

► Οι δραστηριότητες αυτές μπορούν να προσαρμοστούν για εφαρμογή στην ανάλυση κινδύνων ενός έργου ανάπτυξης ΠΣ.





Ένας οργανισμός θεωρείται ότι βρίσκεται σε κατάσταση ετοιμότητας αναφορικά με την κυβερνοασφάλεια όταν επιτυγχάνει ορισμένους βασικούς στόχους κυβερνοασφάλειας, όπως η ικανότητα του οργανισμού να ανιχνεύει και να ανταποκρίνεται αποφασιστικά:

- σε εισβολές και παραβιάσεις της ασφάλειας υπολογιστών,
- σε κακόβουλες επιθέσεις,
- σε επιθέσεις αλίευσης δεδομένων,
- σε επιθέσεις αλλοίωσης ή κλοπής δεδομένων και πνευματικής ιδιοκτησίας από εντός και εκτός του δικτύου.

- Probability: 1-5 or medium, low, high
- Impact: 1-5 or medium, low, high

Risk ID	Description	Probability	Impact	Probability * Impact	Mitigation plan
MG-1	Shortage personnel	2	3	6	Employ additional ...
MG-2	Delays timeplan	1	2	2	Employ additional ...
TC-1	Data availability	1	5	5	Work with open data until data become available
TC-2	Complexity of interfaces	4	4		Create additional interfaces splitting the initial ones
..	Performance > 10sec	5	3		Add a load balancer
	New risk				

- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Η σύλληψη, ο προγραμματισμός, η ανάπτυξη και η συντήρηση ΠΣ μεγάλης κλίμακας τα οποία προορίζονται για την αντιμετώπιση δομημένων προβλημάτων πραγματοποιούνται στο πλαίσιο μιας διαδικασίας που έχει δομηθεί σε φάσεις και ονομάζεται **κύκλος ζωής ανάπτυξης συστημάτων (ΚΖΑΣ)**.
- Ένας ΚΖΑΣ περιλαμβάνει όλες τις δραστηριότητες που απαιτούνται για (και όλα τα παραδοτέα που παράγονται κατά) τη μελέτη, τον προγραμματισμό, την ανάλυση, το σχεδιασμό, την πραγμάτωση, τη λειτουργία και τη συντήρησή του ΠΣ κατά τρόπο ώστε να ικανοποιούνται αποτελεσματικά και αποδοτικά οι απαιτήσεις των χρηστών.

Ο ΚΖΑΣ αποτελεί μια ολοκληρωμένη τμηματική προσέγγιση εκτέλεσης ενός έργου, που αφορά στην ανάπτυξη ενός ΠΣ, η οποία κατά βάση περιλαμβάνει τις ακόλουθες έξι φάσεις:

- μελέτη εφικτότητας (*feasibility study*),
- μελέτη του συστήματος (*system investigation*),
- ανάλυση συστήματος (*system analysis*),
- σχεδιασμός συστήματος (*system design*),
- πραγμάτωση συστήματος (*system implementation*),
- λειτουργία συστήματος (*system operation*) και
- συντήρηση συστήματος (*system maintenance*).

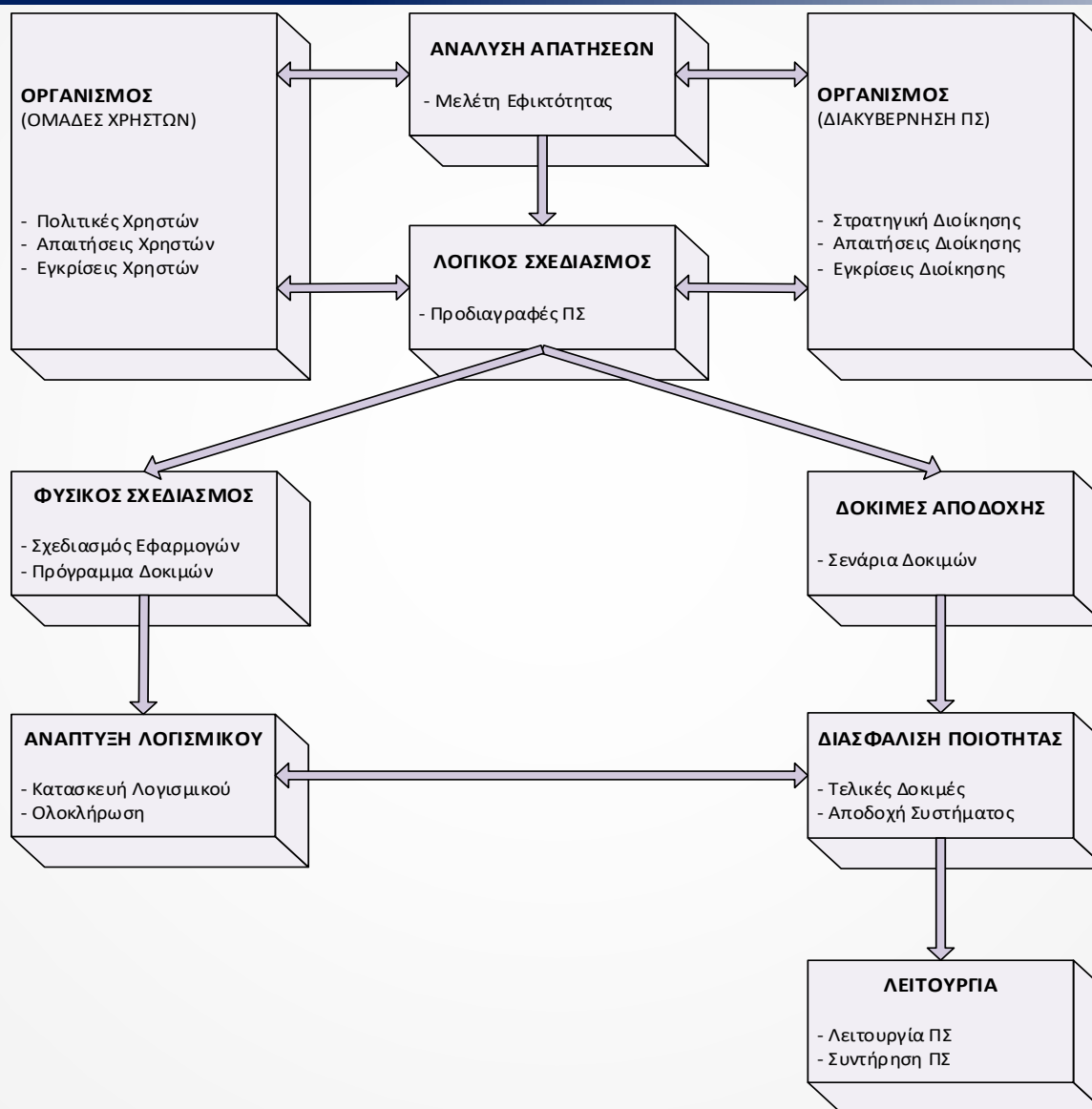
- Η υιοθέτηση αυστηρών ΚΖΑΣ που χαρακτηρίζονται από διακριτές φάσεις επιλύει σε σημαντικό βαθμό τέτοια προβλήματα.
- Πριν υιοθετηθεί ένας ΚΖΑΣ είναι ανάγκη να αποφασίσει η διοίκηση του έργου μεταξύ δύο βασικών προσεγγίσεων:
  - Της προσέγγισης που δίνει έμφαση στην τήρηση του χρονοδιαγράμματος και του προϋπολογισμού του έργου και χαρακτηρίζεται από την προσήλωση στη γραμμική εκτέλεση των φάσεων και των δραστηριοτήτων του ΚΖΑΣ κατά το καταρρακτοειδές μοντέλο
  - Της προσέγγισης που δίνει έμφαση στην τήρηση των προδιαγραφών ποιότητας του συστήματος αναφορικά με τη χρησιμότητα και την χρηστικότητα του και χαρακτηρίζεται από την μετ' επαναλήψεων εκτέλεση των φάσεων, ή δραστηριοτήτων των φάσεων, εφόσον απαιτείται.



- Περιπτώσεις στις οποίες δίδεται **ιδιαίτερη βαρύτητα στην τήρηση του χρονοδιαγράμματος** και **μικρότερη στα ποιοτικά χαρακτηριστικά του τελικού προϊόντος** ακολουθούνται μόνο εφόσον επιβάλλεται από τις επικρατούσες συνθήκες.
- Προβλήματα οδηγούν συχνά σε **υπερβατικά έργα, έργα στα οποία σημειώνονται σημαντικές υπερβάσεις προϋπολογισμού και χρονοδιαγράμματος**, ενώ από την άλλη μεριά πλείστα όσα έργα ΠΣ αποτυγχάνουν και είτε δεν χρησιμοποιούνται είτε εγκαταλείπονται.
- Όλα αυτά τα προβλήματα είναι σε μεγάλο βαθμό απόρροια της έλλειψης αυστηρών κανόνων παρακολούθησης και ελέγχου.

# Σχέσεις μεταξύ των φάσεων ενός κύκλου ζωής

42



ΦΑΣΗ ΚΥΚΛΟΥ ΖΩΗΣ		ΚΥΡΙΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ	ΠΑΡΑΔΟΤΕΑ / ΤΕΚΜΗΡΙΩΣΗ
1.	Ανάλυση Απαιτήσεων	<ul style="list-style-type: none"> <li>Αξιολόγηση αιτήματος χρήστη</li> <li>Εκπόνηση μελέτης εφικτότητας</li> <li>Προσδιορισμός απαιτήσεων συστήματος</li> <li>Κατασκευή προγράμματος εργασιών</li> </ul>	<ul style="list-style-type: none"> <li>Προκαταρκτική μελέτη και ορισμός <b>εύρους του έργου</b></li> <li>Μελέτη <b>εφικτότητας</b></li> <li>Οριστική πρόταση με εκτίμηση χρονοδιαγράμματος και προϋπολογισμού</li> <li>Απαιτήσεις συστήματος</li> <li>Πρόγραμμα εργασιών για την εκτέλεση του έργου</li> <li>Ορισμός <b>χρηστών/κατηγοριών</b> του ΠΣ</li> <li>Ανάλυση <b>υπάρχουσας κατάστασης</b></li> </ul>
2.	Λογικός Σχεδιασμός / <b>Προδιαγραφές</b>	<ul style="list-style-type: none"> <li>Μετασχηματισμός των απαιτήσεων του χρήστη σε εννοιολογικά μοντέλα</li> <li>Βελτίωση της εννοιολογικής δομής του πληροφοριακού συστήματος</li> </ul>	<ul style="list-style-type: none"> <li><b>Λογικός σχεδιασμός</b> συστήματος (<b>conceptual architecture v1, v2, v3</b>)</li> <li>Λογικός σχεδιασμός βάσεων δεδομένων</li> <li>Αρχιτεκτονική συστήματος σε ανώτερο επίπεδο</li> </ul>
3.	Φυσικός Σχεδιασμός	<ul style="list-style-type: none"> <li>Αναγνώριση της ιεραρχίας των λογικών μερών</li> <li>Αναγνώριση της επικοινωνίας μεταξύ των λογικών μερών</li> <li>Οριστικοποίηση αρχιτεκτονικής συστήματος και εξοπλισμού (υλικού και λογισμικού)</li> <li>Προετοιμασία προγράμματος ελέγχων</li> </ul>	<ul style="list-style-type: none"> <li><b>Λεπτομερής αρχιτεκτονική συστήματος (component model, sequence diagram v1, v2, v3)</b></li> <li>Λειτουργικές και τεχνικές προδιαγραφές συστήματος</li> <li>Διακήρυξη προμήθειας υλικού και λογισμικού συστήματος</li> <li><b>Φυσικός σχεδιασμός βάσεων δεδομένων</b></li> <li>Προγράμματα ελέγχων συστήματος</li> </ul>

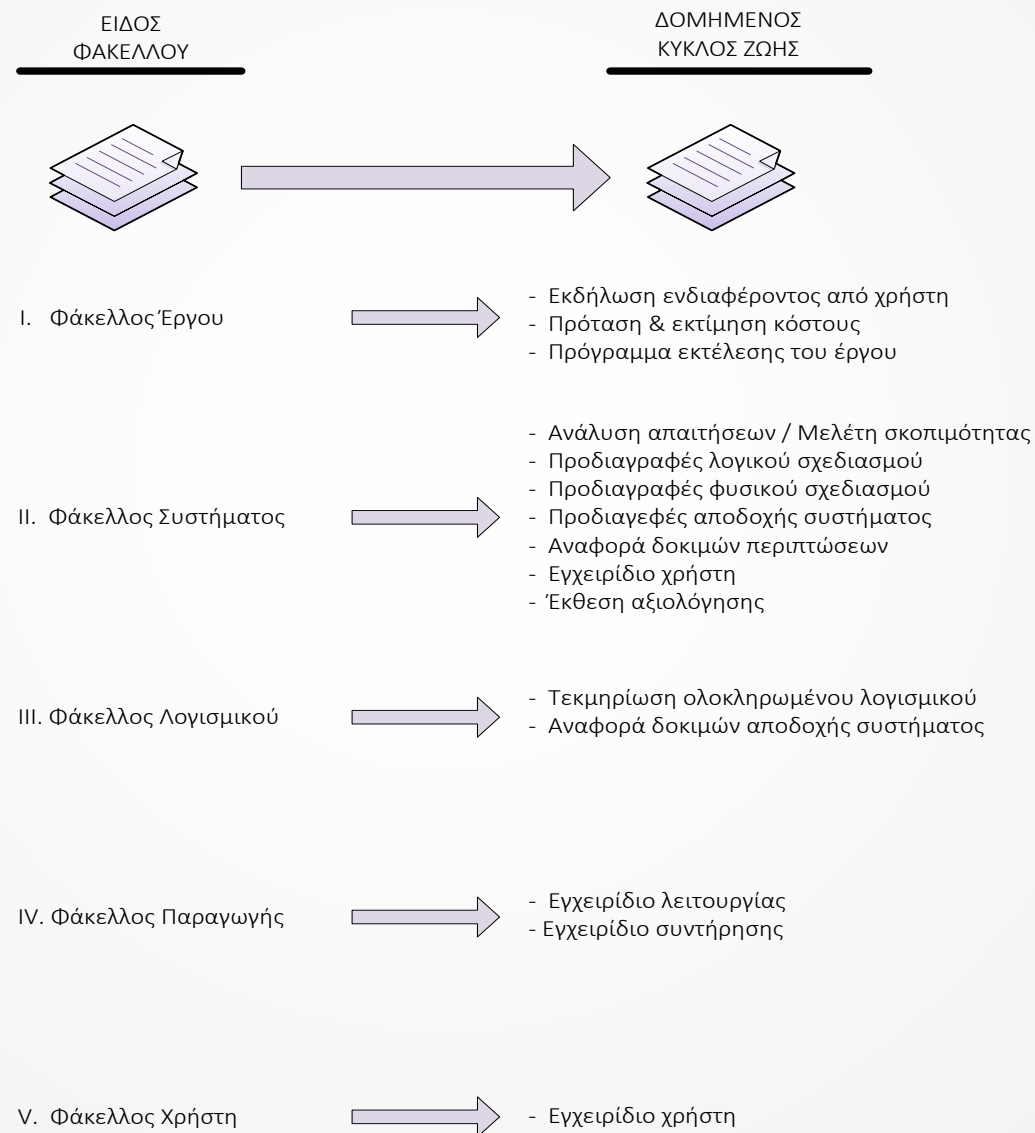
ΦΑΣΗ ΚΥΚΛΟΥ ΖΩΗΣ		ΚΥΡΙΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ	ΠΑΡΑΔΟΤΕΑ / ΤΕΚΜΗΡΙΩΣΗ
4.	Πραγμάτωση	<ul style="list-style-type: none"> <li>♦ Προγραμματισμός λογικών μερών</li> <li>♦ Ολοκλήρωση λογικών μερών</li> <li>♦ Δοκιμές λογικών μερών</li> <li>♦ Δοκιμές συστήματος</li> </ul>	<ul style="list-style-type: none"> <li>♦ Τεκμηρίωση λογισμικού εφαρμογών</li> <li>♦ Προδιαγραφές αποδοχής συστήματος</li> <li>♦ Αναφορές δοκιμών αποδοχής συστήματος</li> </ul>
5.	Λειτουργία	<ul style="list-style-type: none"> <li>♦ Εκπαίδευση χρηστών</li> <li>♦ Προσδιορισμός τρόπου μετάβασης στη νέα κατάσταση</li> <li>♦ Λειτουργία συστήματος</li> </ul>	<ul style="list-style-type: none"> <li>♦ Εγχειρίδιο χρήστη</li> <li>♦ Εκπαιδευτικό υλικό</li> <li>♦ Εγχειρίδιο λειτουργίας</li> </ul>
8.	Συντήρηση	<ul style="list-style-type: none"> <li>♦ Αξιολόγηση συστήματος</li> <li>♦ Συντήρηση συστήματος</li> </ul>	<ul style="list-style-type: none"> <li>♦ Αναφορά αξιολόγησης</li> <li>♦ Εγχειρίδιο συντήρησης</li> </ul>

<div> <div>Φάσεις Κύκλου Ζωής</div> <div>Μέθοδοι/ Τεχνικές</div> </div>	Ανάλυση Απαιτήσεων	Λογικός Σχεδιασμός	Φυσικός Σχεδιασμός	Πραγμάτωση	Λειτουργία	Συντήρηση
Διαγράμματα Ροής Δεδομένων (Data Flow Diagrams)	X	X	X	X		
Λεξικό Δεδομένων (Data Dictionary)	X	X	X	X		
HIPO (Hierarchical IPO)		X	X	X		
Διαγράμματα Οντοτήτων-Συσχετίσεων (Entity-Relationship Diagrams)		X				
Διαγράμματα Δομών (Structure Charts)			X	X		
Ψευδοκώδικας (Pseudocode)			X	X		
Προγραμματισμός (Programming)			X	X		
Έλεγχος (Testing)				X	X	X
Συντήρηση (Maintenance)						X

Μέθοδοι/ Τεχνικές \ Φάσεις Κύκλου Ζωής	Ανάλυση Απαιτήσεων	Λογικός Σχεδιασμός	Φυσικός Σχεδιασμός	Πραγμάτωση	Λειτουργία	Συντήρηση
Διαγράμματα Περιπτώσεων Χρήσης (Use Case Diagrams)	X	X				
Διαγράμματα Δραστηριοτήτων (Activity Diagrams)		X	X			
Διαγράμματα Συνιστωσών (Component Diagrams)		X	X	X		
Διαγράμματα Ακολουθιών (Sequence Diagrams)		X	X	X		
Διαγράμματα Επικοινωνίας (Communication Diagrams)		X	X	X		
Διαγράμματα Κλάσεων (Class Diagrams)		X	X	X		
Διαγράμματα Αντικειμένων (Object Diagrams)			X	X		
Διαγράμματα Πακέτων (Package Diagrams)				X	X	X
Διαγράμματα Διάταξης (Deployment Diagrams)				X	X	X

# Κατηγοριοποίηση παραδοτέων κατά την εκτέλεση των φάσεων ενός κύκλου ζωής ανάπτυξης συστημάτων

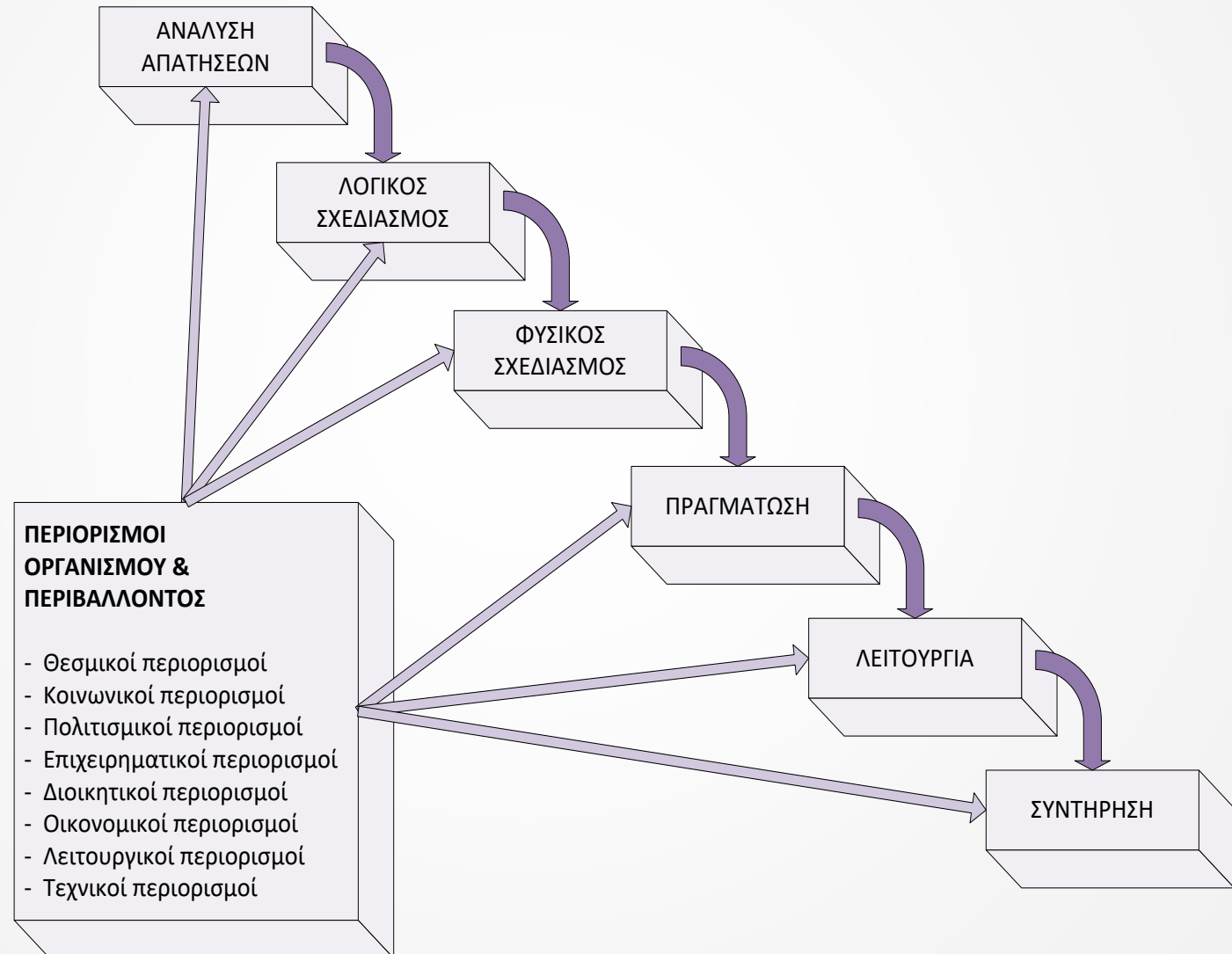
47



ΦΑΣΗ ΚΥΚΛΟΥ ΖΩΗΣ	ΠΑΡΑΔΟΤΕΟ
Συλλογή Απαιτήσεων	Τεύχος Απαιτήσεων Χρηστών
Ανάλυση Απαιτήσεων	Μελέτη Εφικτότητας
Σχεδιασμός Συστήματος	<ul style="list-style-type: none"> <li>– Τεύχος Λειτουργικών &amp; Τεχνικών Προδιαγραφών</li> <li>– Τεύχος Αρχιτεκτονικής Συστήματος</li> </ul>
Σχεδιασμός Λογισμικού Εφαρμογών	Τεύχος Προδιαγραφών Σχεδιασμού Συστήματος
Κωδικοποίηση Λογισμικού Εφαρμογών	<ul style="list-style-type: none"> <li>– Λογισμικό Εφαρμογών</li> <li>– Λογισμικό Υποβοήθησης Χρηστών</li> </ul>
Έλεγχος Μονάδων και Ολοκλήρωσης	<ul style="list-style-type: none"> <li>– Αναφορά Ελέγχου Μονάδων και Ολοκλήρωσης</li> <li>– Αναφορά Διορθωτικών Δράσε-ων</li> </ul>
Έλεγχος Συστήματος	<ul style="list-style-type: none"> <li>– Αναφορά Ελέγχου Συστήματος</li> <li>– Αναφορά Διορθωτικών Δράσε-ων</li> </ul>
Έλεγχος Αποδοχής – Δοκιμαστική Λειτουργία Συστήματος	<ul style="list-style-type: none"> <li>– Αναφορά Ελέγχου Αποδοχής</li> <li>– Αναφορά Διορθωτικών Δράσε-ων</li> </ul>
Παράδοση Συστήματος	<ul style="list-style-type: none"> <li>– Εγχειρίδιο Χρήστη</li> <li>– Εγχειρίδιο Συστήματος</li> </ul>



ΦΑΣΗ ΚΥΚΛΟΥ ΖΩΗΣ	ΠΑΡΑΔΟΤΕΟ
Εγκατάσταση & Παραγωγική Λειτουργία Συστήματος	<ul style="list-style-type: none"><li>– Εκπαίδευση Χρηστών</li><li>– Σύστημα σε Παραγωγική Λειτουργία</li><li>– Υποστήριξη Χρηστών</li></ul>
Συντήρηση Συστήματος	<ul style="list-style-type: none"><li>– Διορθώσεις Σφαλμάτων</li><li>– Τροποποιήσεις &amp; Βελτιώσεις Συστήματος</li><li>– Αναβαθμίσεις Συστήματος</li></ul>



- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Επιπλέον, στο πλαίσιο της ανάλυσης, πρέπει να κατανοηθεί τί γίνεται και **γιατί** γίνεται αυτό.
- Μόνον έτσι, η ανάπτυξη του νέου ΠΣ θα χρησιμοποιηθεί ως μοχλός προκειμένου να υπάρξει αναδιοργάνωση των διαδικασιών και του οργανισμού.
- Η αναδιοργάνωση ενός οργανισμού με μοχλό το υπό ανάπτυξη ΠΣ ονομάζεται **αναδιοργάνωση επιχειρησιακών διαδικασιών με βάση την τεχνολογία της πληροφορίας**. = **καταγραφή διαδικασίας**
- Χρησιμοποιούνται διαγραμματικές αναπαραστάσεις με χρήση τεχνικών (γλωσσών) όπως τα διαγράμματα ροής δεδομένων (*Data Flow Diagrams – DFDs*) ή τα διαγράμματα περιπτώσεων χρήσης (*Use Case Diagrams – UCDs*).
- Τα διαγράμματα αυτά είναι αρκετά εύληπτα από τους χρήστες ακόμη και χωρίς την παροχή τυπικής εκπαίδευσης σ' αυτούς.

- Η ενδελεχής μελέτη του οργανισμού μέσω της ανάλυσης που, ως διαδικασία, αρχίζει από την απόκτηση μιας πλούσιας εικόνας της λειτουργίας του οργανισμού ευθυγραμμίζεται πλήρως με δύο βασικούς κανόνες που επιβάλλουν ότι

“η ανάλυση πρέπει να εστιάζεται στον προσδιορισμό των αναγκών του οργανισμού και όχι στον προσδιορισμό των απαιτήσεων των χρηστών του που μπορεί να περιέχουν ένα βαθμό μεροληψίας” = **επικύρωση απαιτήσεων από product owner**

και ότι

“ένα σύστημα που εξυπηρετεί ένα άλλο (π.χ. ένα πληροφοριακό σύστημα που εξυπηρετεί ένα επιχειρησιακό σύστημα) δεν μπορεί να σχεδιαστεί ικανοποιητικά αν δεν κατανοηθεί, αναλυθεί και σχεδιαστεί πρώτα το εξυπηρετούμενο σύστημα”.

- Όταν κατανοηθεί επαρκώς η τρέχουσα λειτουργία του συστήματος, προσδιορίζονται οι απαιτήσεις του συστήματος.
- Οι απαιτήσεις του συστήματος αναφέρονται στο τί (όχι πώς) προβλέπεται να πράττει το σύστημα και εντός ποιων περιορισμών πρέπει να πράττει τα προβλεπόμενα και διακρίνονται σε δύο κύριες κατηγορίες:
  - **Λειτουργικές απαιτήσεις:** Αναφέρονται στις λειτουργίες τις οποίες αναμένεται να ικανοποιήσει το σύστημα.
  - **Τεχνικές απαιτήσεις:** Αναφέρονται στα τεχνικά χαρακτηριστικά που πρέπει να διαθέτει το νέο σύστημα (χωρίς να γίνεται αναφορά στις τεχνολογίες που θα χρησιμοποιηθούν για την κατασκευή του συστήματος) και στους περιορισμούς ασφάλειας, αποδοτικότητας και διαθεσιμότητας που πρέπει να εκπληρούνται.

- Κατά τη δραστηριότητα αυτή συλλέγονται, επαληθεύονται και αναλύονται οι πληροφοριακές ανάγκες του οργανισμού, ή του μέρους του οργανισμού, υπό μελέτη.
- Συνεπώς, κύριος σκοπός της δραστηριότητας είναι ο ορισμός:
  - των εισροών,
  - των αρχείων,
  - των επεξεργασιών και
  - των εκροών.

Γενικά, κατά τη φάση του προσδιορισμού των απαιτήσεων πρέπει να λαμβάνονται υπόψη τα ακόλουθα:

- Η επικοινωνία μεταξύ αναλυτών και χρηστών πρέπει να γίνεται κατά τρόπο που είναι κατανοητός από τους χρήστες.
  - Οι χρήστες δεν κατανοούν την ορολογία της πληροφορικής για την αναπαράσταση ΠΣ
  - Οι αναλυτές δεν κατανοούν συχνά την ορολογία των χρηστών και τις δραστηριότητες του οργανισμού που αναφέρονται στην καθημερινή εργασία τους.
- Πρέπει να αφιερωθεί κάποιο ποσοστό χρόνου από τους αναλυτές για την εξοικείωσή τους με τη λειτουργία και τις δραστηριότητες του οργανισμού για τον οποίο θα κατασκευαστεί το ΠΣ. Με τον τρόπο αυτό καθίσταται ευκολότερη την κατασκευή του εννοιολογικού μοντέλου δεδομένων.



- Οι διαφορετικές ομάδες χρηστών έχουν περιορισμένη και μεροληπτική αντίληψη για τις λειτουργίες, τις διεργασίες και τα δεδομένα του οργανισμού, η οποία εξαρτάται από τη φύση της εργασίας τους. Για να αποκτηθεί μια σφαιρική εικόνα του οργανισμού:
  - Οι αναλυτές πρέπει να συνεργάζονται με χρήστες που διαθέτουν σημαντική εμπειρία και ευρεία γνώση του αντικειμένου της εργασίας τους (*knowledgeable users*) από κάθε λειτουργική μονάδα ή επιχειρησιακή διεργασία του οργανισμού
  - Πρέπει να δοθεί ο απαιτούμενος χρόνος στη συνεργασία αυτή στο πλαίσιο του χρονοπρογραμματισμού του όλου έργου.

- Συχνά, οι απαιτήσεις των χρηστών είναι αντικρουόμενες, με την έννοια ότι η ικανοποίηση μιας απαίτησης μπορεί να έχει αρνητικές επιπτώσεις στην ικανοποίηση άλλων απαιτήσεων.  
**επικύρωση απαιτήσεων από product owner**
  - Οι απαιτήσεις να σταθμίζονται και αξιολογούνται ως προς τις επιπτώσεις τους σε άλλες.
  - Στο πλαίσιο του εφικτού και επιθυμητού (ευκταίου) μπορεί τελικά να ικανοποιηθούν οι απαιτήσεις, όχι σε απόλυτο αλλά σε αποδεκτό βαθμό.
- Επειδή οι απαιτήσεις συχνά εκφράζονται ή συλλέγονται σε διάφορα επίπεδα αφαίρεσης, πρέπει να αναλύονται ή να συντίθενται στα επίπεδα αφαίρεσης που εξυπηρετούν τις επόμενες φάσεις του ΚΖΑΣ ώστε να διευκολύνεται η διαδικασία της ανάπτυξης της βάσης δεδομένων.

- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Για την ανάπτυξη ενός ΠΣ που θα ικανοποιεί τις απαιτήσεις σχεδιάζονται διάφορες εναλλακτικές λύσεις σε ανώτερο επίπεδο αφαίρεσης (χαμηλό επίπεδο λεπτομέρειας).
- Η πλέον κατάλληλη λύση επιλέγεται κατόπιν αξιολόγησης των καταγεγραμμένων εναλλακτικών λύσεων.
- Η αξιολόγηση αυτή πραγματοποιείται με βάση τις ακόλουθες (υπό) μελέτες:
  - Τη Μελέτη Τεχνικής Εφικτότητας
  - τη Μελέτη Οικονομικής Εφικτότητας
  - τη Μελέτη Επιχειρησιακής Εφικτότητας

## I. Τεχνική εφικτότητα

Ένα νέο ΠΣ είναι τεχνικά εφικτό αν:

- a. όλες οι συνιστώσες του (υλικού και λογισμικού) υπάρχουν διαθέσιμες στο εμπόριο ή μπορούν να αναπτυχθούν με διαθέσιμα εργαλεία, και
- b. υπάρχει διαθέσιμο το αναγκαίο προσωπικό πληροφορικής με τις απαιτούμενες γνώσεις και εμπειρίες (σε περίπτωση εσωτερικής ανάπτυξης αναφέρεται στο προσωπικό πληροφορικής του οργανισμού και σε περίπτωση εξωτερικής ανάπτυξης αναφέρεται στο προσωπικό πληροφορικής της ελεύθερης αγοράς ή στις εταιρείες πληροφορικής).

## II. Επιχειρησιακή εφικτότητα.

Σκοπός της μελέτης επιχειρησιακής εφικτότητας είναι να καθορίσει αν το νέο ΠΣ θα είναι εναρμονισμένο με επιχειρησιακό περιβάλλον του οργανισμού και αν θα χρησιμοποιηθεί όπως προβλέπεται.

Συγκεκριμένα, η μελέτη αυτή αναφέρεται στα ακόλουθα ερωτήματα:

- Θα χρησιμοποιούν οι προβλεπόμενοι χρήστες το σύστημα στην **ολότητά** του;
- Τι αλλαγές θα επιφέρει το σύστημα στο **περιβάλλον εργασίας** των χρηστών;
- Θα δημιουργήσει, ενδεχομένως, η προτεινόμενη λύση **νέα προβλήματα**;
- Θα παρεισδύει το σύστημα στις **πολιτικές ή στους κανονισμούς του οργανισμού**;

## III. Οικονομική εφικτότητα.

- Σκοπός της μελέτης οικονομικής εφικτότητας είναι η οικονομική αιτιολόγηση και η διερεύνηση της βιωσιμότητας του νέου ΠΣ, όπως ακριβώς συμβαίνει με κάθε έργο.
- Δηλαδή, στη διάρκεια ζωής του συστήματος τα οικονομικά, και όχι μόνον, οφέλη πρέπει υπερκεράζουν τις δαπάνες.
- Η διερεύνηση αυτή πραγματοποιείται μέσω μιας ανάλυσης κόστους/οφέλους (*cost/benefit analysis*) που, πολύ απλά, μπορεί να είναι ένα λογιστικό φύλο στο οποίο έχουν καταγραφεί όλες οι δαπάνες που επισύρει το σύστημα και όλα τα αναμενόμενα οφέλη από τη λειτουργία του.

- Η πιο ακριβής μέθοδος οικονομικής ανάλυσης είναι η εκτίμηση της απόδοσης επί της επένδυσης – ATE (*Return on Investment – ROI*) η οποία είναι ένας υπολογισμός της διαφοράς μεταξύ μιας δέσμης οφελών και της δέσμης των δαπανών στη διάρκεια ζωής του συστήματος, μειωμένη κατά τα αντίστοιχα επιτόκια.
- Για την εκτίμηση της ATE, υπολογίζεται η καθαρή παρούσα αξία του συστήματος, με βάση την καθαρή παρούσα αξία των δαπανών και την καθαρή παρούσα αξία των οφελών του συστήματος, υπολογίζοντας τις ετήσιες δαπάνες και τα οφέλη και απομειώνοντάς τα με βάση κατάλληλα επιτόκια.
- Αν η ATE είναι θετική, τότε η ανάπτυξη του ΠΣ είναι οικονομικά εφικτή ή είναι δικαιολογημένη η δαπάνη που επισύρεται για την ανάπτυξη του ΠΣ.



- Κατά τη διάρκεια του χρόνου για τη ανάπτυξη του ΠΣ, που μπορεί να είναι αρκετοί μήνες, δεν υπάρχουν οφέλη αλλά μόνο δαπάνες που αναφέρονται στην ανάπτυξη του ΠΣ.
- Κατά τη διάρκεια της ζωής του ΠΣ, μεταξύ των λειτουργικών δαπανών περιλαμβάνονται:
  - οι δαπάνες για το προσωπικό συντήρησης,
  - οι τηλεπικοινωνιακές δαπάνες,
  - οι δαπάνες ηλεκτρικού ρεύματος
  - οι προμήθειες πάγιου και αναλώσιμου υλικού πληροφορικής (π.χ. εκτυπωτικό χαρτί, ανταλλακτικά υπολογιστών, αναβαθμίσεις λογισμικού συστήματος και εφαρμογών)
  - η δαπάνη για την αναβάθμιση και τον εμπλουτισμό του ιστοτόπου.

- Στον παρακάτω πίνακα παρουσιάζεται ένα απλοποιημένο παράδειγμα λογιστικού φύλου ανάλυσης δαπάνης/οφέλους για ένα ΠΣ.
- Επειδή η καθαρή παρούσα αξία του συστήματος είναι θετική, που σημαίνει ότι τα οφέλη υπερβαίνουν την επένδυση, δικαιολογείται η ανάπτυξη του ΠΣ.
- Όμως, το έτος 2008 φαίνεται ότι η καθαρή παρούσα αξία αρχίζει να μειώνεται.
- Καθώς συνεχίζει να μειώνεται αυτή η αξία, ο οργανισμός πρέπει να θεωρήσει την περίπτωση ανάπτυξης ενός νέου ΠΣ με βελτιωμένα τεχνικά και λειτουργικά χαρακτηριστικά.
- Αν το σύστημα δεν αντικατασταθεί ή δεν αναβαθμιστεί δραστικά, θα μετατραπεί σταδιακά σε μαύρη τρύπα για τον οργανισμό και ο οργανισμός θα απωλέσει την ευκαιρία να κεφαλαιοποιήσει επί της νέας τεχνολογίας.

# Εκτίμηση κόστους/οφέλους ενός ΠΣ

67

Έτος	2005	2006	2007	2008	2009	2010
<b>Οφέλη</b>						
Αύξηση πωλήσεων			55.000	45.000	30.000	10.000
Μείωση προσωπικού			20.000	20.000	20.000	20.000
<b>Σύνολο οφελών (α)</b>	0	0	75.000	65.000	50.000	30.000
<b>Δαπάνες</b>						
Ανάλυση	15.000					
Σχεδιασμός	37.500					
Πραγμάτωση	0	55.000				
Υλικό / Λογισμικό Συστήματος	0	20.000				
Λειτουργία / Συντήρηση	0	0	5.000	5.000	5.000	5.000
<b>Σύνολο δαπανών (β)</b>	52.000	75.000	5.000	5.000	5.000	5.000
<b>Διαφορά (α)-(β)</b>	-52.000	-75.000	71.000	60.000	45.000	25.000
Κατόπιν έκπτωσης 5%	-49.524	-68.934	61.332	49.362	32.259	18.657
<b>Καθαρή παρούσα αξία για 6 έτη</b>	43.152					

# Business plan

68

Year	2020	2021	2022	2023	2024
Foreseen users (from market analysis)	2	8	20	63	
Income/ benefits					
Flat rate	100	400			
License / user	20	80			
Distributors ...					
Costs					
Planning	3000				
Development	5000				
Testing	1000				
Deployment	500				
Operational costs / Cloud	100	1000	10000		
Marketing (lines: Google Adwords, Website, Promotional ...)	0	50000	20000		
Benefits - costs	-	-	+		

- Η μελέτη εφικτότητας καταλήγει σε μια διαδικασία αξιολόγησης των εναλλακτικών λύσεων που έχουν καταγραφεί και μελετηθεί κατά τα προηγούμενα.
- Κατά τη διαδικασία της αξιολόγησης των εναλλακτικών λύσεων διαγράφονται αμέσως οι ανέφικτες λύσεις.
- Αν οι εφικτές λύσεις είναι περισσότερες από μία θα πρέπει να επιλεγεί η πλέον εφικτή μεταξύ αυτών.
- Σε μερικές περιπτώσεις, λαμβάνονται υπόψη πρόσθετοι λόγοι που μπορεί να επιβάλλουν την επιλογή μιας, εκ πρώτης όψεως, λιγότερο εφικτής λύσης (π.χ. γενικότεροι οικονομικοί, στρατηγικοί, κοινωνικοί ή πολιτικοί λόγοι).

Η αναφορά της μελέτης εφικτότητας που υποβάλλεται στη διοίκηση του οργανισμού για τη λήψη της τελικής απόφασης πρέπει να περιέχει:

- Συμπεράσματα και προτάσεις που θα συνοδεύονται από την αντίστοιχη τεκμηρίωση. Στα συμπεράσματα πρέπει να περιλαμβάνονται:
  - οι δαπάνες για το υπάρχον και το προτεινόμενο ΠΣ,
  - οι κύριες απαιτήσεις του προτεινόμενου ΠΣ σε ανθρώπινους και υλικούς πόρους,
  - μια ανάλυση των πιθανών οφελών και
  - η λογική βασιμότητα των προτάσεων.

- Μια ανάλυση του προτεινόμενου και των εναλλακτικών ΠΣ σύμφωνα με εκτιμήσεις σχετικά με το επαναλαμβανόμενο (λειτουργικό) και μη επαναλαμβανόμενο (πάγιο) κόστος που υπεισέρχεται καθώς και με τα άμεσα (μετρήσιμα) και έμμεσα (μη μετρήσιμα) οφέλη που εκτιμάται ότι θα προκύψουν.
- Ένα προσωρινό χρονοδιάγραμμα για την προετοιμασία των λειτουργικών και τεχνικών προδιαγραφών καθώς και όλα τα απαιτούμενα στοιχεία που θα βοηθήσουν τη διοίκηση του οργανισμού στη λήψη της απόφασης σχετικά με τη συνέχιση ή μη του έργου.

- **Τα ορόσημα.** Είναι χρονικά σημεία κατά τη διάρκεια εξέλιξης του έργου που σηματοδοτούν:
  - την έναρξη ή την περάτωση των επιμέρους εργασιών,
  - την επίτευξη των στόχων,
  - τις διοικητικές ανασκοπήσεις και εγκρίσεις και τα σημαντικά επιτεύγματα.

Τα ορόσημα αποτελούν κατάλληλα χρονικά σημεία για την υποβολή εκθέσεων σχετικά με τη πορεία του όλου έργου ή για την μέτρηση και αξιολόγηση της προόδου του.

Milestones: **Decision (ON/OFF)**



- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Κατά τη φάση του λογικού σχεδιασμού (ή φάση του γενικού σχεδιασμού) του κύκλου ζωής καθορίζεται η λεπτομερής εννοιολογική δομή του ΠΣ με στόχο την ικανοποίηση των απαιτήσεων του οργανισμού.
- Στη φάση αυτή, οι πληροφοριακές απαιτήσεις του οργανισμού μετασχηματίζονται σε σειρά εννοιολογικών μοντέλων που καλύπτουν την αρχιτεκτονική και κάθε άποψη (στατική ή δυναμική) του νέου ΠΣ.
- Σύμφωνα με τη δομημένη προσέγγιση οι δραστηριότητες αυτής της φάσης εκτελούνται χρησιμοποιώντας κάποια **δομημένη μεθοδολογία ανάλυσης και σχεδιασμού**.

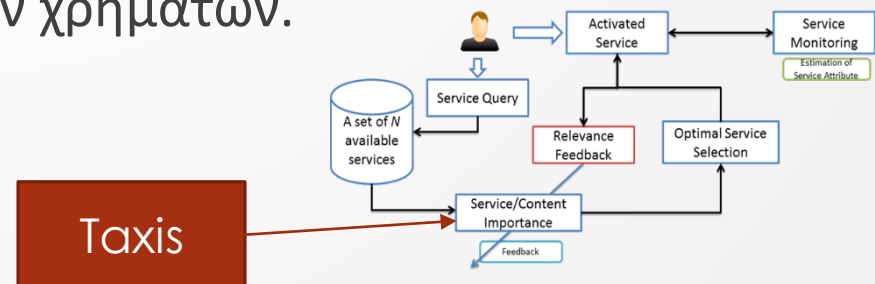
- Η δομημένη προσέγγιση του κύκλου ζωής επιβάλλει το **σχεδιασμό της εννοιολογικής δομής του νέου ΠΣ** κατά την προσέγγιση **από-το-γενικό-στο-ειδικό** που σημαίνει ότι ένα σύστημα (ανωτέρου επιπέδου) διασπάται σε υποσυστήματα (κατωτέρου επιπέδου) τα οποία μελετώνται ανεξάρτητα μεταξύ τους αλλά εντός του πλαισίου που προσδιορίζεται από το ευρύτερο σύστημα.
- Κατά την εκτέλεση αυτής της δραστηριότητας πρέπει να καταβάλλεται κάθε προσπάθεια ώστε να καθοριστούν επακριβώς οι **εισροές**, οι **εκροές** και οι **διεργασίες** που απαιτούνται **για κάθε λειτουργική** (όχι κατ' ανάγκη ταυτιζόμενη με οργανωσιακή) **μονάδα ή λογικό τμήμα του υπό ανάπτυξη ΠΣ.**

- Κάθε **λογικό τμήμα** πρέπει να περιγράφεται στον βαθμό λεπτομέρειας που απαιτείται από το αντίστοιχο επίπεδο του λογικού σχεδιασμού.
- Κατά παρόμοιο τρόπο πρέπει επίσης να περιγράφονται οι **ροές** και οι **δραστηριότητες** που αναλαμβάνονται για τους μετασχηματισμούς των δεδομένων (οι εισροές μετασχηματίζονται σε εκροές).
- Τα **δεδομένα** μπορεί να αναπαριστούν τις απαιτήσεις επικοινωνίας μεταξύ των λογικών τμημάτων.
- Τα **σημεία επικοινωνίας** πρέπει να προσδιορίζονται επακριβώς πριν το σχεδιασμό του κάθε λογικού τμήματος.

Για τη δημιουργία μιας ιεραρχίας λογικών τμημάτων μπορεί να χρησιμοποιηθεί η τεχνική της λειτουργικής διάσπασης.

- Κάθε λειτουργία διασπάται σε διαρκώς κατώτερα επίπεδα λεπτομέρειας μέχρις ότου καθοριστούν πλήρως όλες οι υπολειτουργίες που τη συνιστούν.
- Το ανώτερο επίπεδο της ιεραρχίας περιγράφει ολόκληρο το υπό μελέτη σύστημα ως μία λειτουργία.
- Κάθε λειτουργία κατωτέρου επιπέδου αποτελεί ένα "υποσύνολο" της λειτουργίας του αμέσως ανωτέρου επιπέδου.
- Οι λειτουργίες ανωτέρων επιπέδων περιέχουν τη λογική ελέγχου που προσδιορίζει **πότε** και σε **ποια** σειρά θα γίνεται η επίκληση των λειτουργιών κατωτέρων επιπέδων.
- Αντίθετα, οι λειτουργίες κατωτέρων επιπέδων είναι κατά κύριο λόγο εκτελεστικές.

- Οι απαιτούμενες βάσεις δεδομένων, πίνακες και αρχεία, στα οποία περιγράφονται τα δεδομένα του συστήματος καθώς και οι μέθοδοι και έλεγχοι που προβλέπεται να εφαρμόζονται επί των δεδομένων.
- Προδιαγραφές των εκροών, στις οποίες περιγράφεται κάθε χειρόγραφο ή ψηφιοποιημένη εκροή από το σύστημα.
- Διαλειτουργικότητα / Σημεία επικοινωνίας του ΠΣ και/ή των χρηστών του με κατακευμασμένα κέντρα επεξεργασίας δεδομένων τα οποία δέχονται δεδομένα από το σύστημα και σημεία επικοινωνίας με άλλα χειρόγραφα ή ψηφιακά συστήματα.
- Απαιτήσεις σε πόρους, συμπεριλαμβανομένων των υπολογιστικών πόρων, του προσωπικού και των χρημάτων.



- Ο όρος “**εκροή**” χρησιμοποιείται για οποιαδήποτε πληροφορία παράγεται από το ΠΣ ανεξάρτητα από τη μορφή της.
- Οι εκροές από ένα ΠΣ αποτελούν τα κύρια σημεία επαφής μεταξύ αυτού και των περισσοτέρων χρηστών του.
- Επομένως, η χρησιμοποίηση και αξιοποίηση του ΠΣ από τους χρήστες, που αποτελούν βασικά κριτήρια για την επιτυχία του, εξαρτάται σε σημαντικό βαθμό από την ποιότητα και τη χρησιμότητα των εκροών.

- Για τον προσδιορισμό του περιεχομένου και της μορφής των εκροών (πληροφοριών ή αναφορών) απαιτείται η συγκέντρωση πληροφοριών σχετικά με τα ακόλουθα ερωτήματα:
  - ποιος θα δέχεται τις εκροές;
  - ποια είναι η προγραμματισμένη χρήση των εκροών;
  - σε ποια λεπτομέρεια απαιτούνται οι εκροές;
  - πότε και πόσο συχνά απαιτούνται οι εκροές;
  - σε ποια μορφή θέλουν τις εκροές οι χρήστες;



- Ο όρος “εισροή” χρησιμοποιείται για οποιαδήποτε πληροφορία χρησιμοποιείται από το ΠΣ ανεξάρτητα από τη μορφή της.
- Ο σχεδιασμός των εισροών αφορά στον καθορισμό των προδιαγραφών και των διεργασιών που απαιτούνται για την προετοιμασία των δεδομένων και την εισαγωγή τους στο υπολογιστικό σύστημα.
- Ο σχεδιασμός των εισροών στοχεύει στον έλεγχο της ποσότητάς τους, στην αποφυγή τυχόν καθυστερήσεων, στην αποφυγή λαθών, στην αποφυγή περιττών διεργασιών και στην απλούστευση των αναγκαίων διεργασιών.
- Η κωδικοποίηση των δεδομένων συμβάλλει σημαντικά στην επίτευξη αυτών των στόχων.
- Όταν είναι δυνατό, τα δεδομένα εισάγονται στο υπολογιστικό σύστημα σε κωδικοποιημένη μορφή.

Οι συνηθέστερες μορφές κωδικοποίησης είναι:

- Ο **κωδικός ταξινόμησης**, που χρησιμοποιείται για την ταξινόμηση διαφόρων οντοτήτων (π.χ. γεγονότων, ανθρώπων ή αντικειμένων) σε διακριτές κατηγορίες.
- Ο **ακολουθιακός κωδικός**, που αποτελείται από αριθμούς ή γράμματα τοποθετημένα κατά μία ακολουθία για την αναπαράσταση της σειράς πραγματοποίησης των γεγονότων (π.χ. δοσοληψιών).
- Ο **μνημονικός κωδικός**, που αποτελείται από γράμματα τα οποία αναφέρονται στα χαρακτηριστικά κάποιου αντικειμένου με στόχο τη συνοπτική περιγραφή του. Για παράδειγμα, ο κωδικός TV-LED-21 μπορεί περιγράφει τις τηλεοράσεις 21 ιντσών τεχνολογίας LED.

- Ο **κωδικός σημαντικού ψηφίου**, που περιέχει υποκώδικες από χαρακτήρες ή αριθμούς παρέχοντας έτσι πολλές πληροφορίες στους χρήστες. Για παράδειγμα, ο κωδικός 90-213-01 περιγράφει ένα φοιτητή αναφορικά με το έτος εισαγωγής του, τον αύξοντα αριθμό επιτυχίας του και το τμήμα του πανεπιστημίου στο οποίο εγγράφηκε.
- Ο **κωδικός λειτουργίας**, που χρησιμοποιείται για την περιγραφή κάποιας δραστηριότητας ή εργασίας κατά μη αφηγηματικό τρόπο. Για παράδειγμα, μπορεί να χρησιμοποιείται ο κωδικός 1 για εισαγωγή, ο κωδικός 2 για διαγραφή και κωδικός 3 για ενημέρωση.

- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Η έννοια της τμηματοποίησης του λογισμικού εφαρμογών οδηγεί στο ακόλουθο θεμελιώδες ερώτημα:

**Πώς πρέπει να πραγματοποιηθεί η διάσπαση μιας εφαρμογής ώστε να παραχθεί το καλύτερο δυνατό σύνολο λογικών τμημάτων υπό την έννοια των παραπάνω χαρακτηριστικών;**

- Σύμφωνα με την **αρχή της απόκρυψης πληροφοριών** κάθε λογικό τμήμα πρέπει να διαθέτει χαρακτηριστικά τα οποία το αποκρύπτουν από όλα τα άλλα λογικά τμήματα. // **separation of concerns**
- Δηλαδή, τα λογικά τμήματα πρέπει να προσδιορίζονται και να σχεδιάζονται κατά τρόπο ώστε τα περιεχόμενά τους (λειτουργίες και δεδομένα) να μην είναι προσπελάσιμα από λογικά τμήματα που δεν τα χρειάζονται για τη λειτουργία τους.

- Τα λογικά τμήματα που έχουν κατασκευαστεί κατά τα παραπάνω μπορούν να συνενωθούν εύκολα σε συνιστώσες που πιθανότατα έχουν προ-σχεδιαστεί προκειμένου να ενταχθούν σε ένα συγκεκριμένο αρχιτεκτονικό πρότυπο για κάποιο πεδίο εφαρμογής.
- Έτσι, η εφαρμογή συναρμολογείται τελικά χρησιμοποιώντας αυτές τις συνιστώσες.
- Το λογισμικό που έχει κατασκευαστεί κατ' αυτόν τον τρόπο ονομάζεται **λογισμικό που βασίζεται σε συνιστώσες** και αποτελεί μια προσέγγιση που, κατά κύριο λόγο, προσβλέπει στην επαναχρησιμοποίηση του λογισμικού.

- Η οργάνωση ή αρχιτεκτονική της **βάσης δεδομένων** καθώς και οι σχετικές μέθοδοι προσπέλασης του χειρισμού των δεδομένων πρέπει να περιγραφούν με κάθε λεπτομέρεια κατά το φυσικό σχεδιασμό.
- Η περιγραφή αυτή πρέπει να περιλαμβάνει τα ακόλουθα:
  - Λογική οργάνωση των πινάκων:
    - συνδέσεις μεταξύ των πινάκων,
    - ευελιξία της βάσης δεδομένων, και
    - λεξικό δεδομένων.
- Database (“structure” of data), data, dataset (subset of the data in the database usually)

- Φυσική οργάνωση των αρχείων:
  - απαιτήσεις φόρτωσης δεδομένων αρχικοποίησης στους πίνακες,
  - τεχνικές προσπέλασης των δεδομένων,
  - διαχείριση του χώρου αποθήκευσης των δεδομένων,
  - τεχνικές σύμπτυξης των δεδομένων,
  - προβλεπόμενα μεγέθη αρχείων δεδομένων και ευρετηρίων,
  - ενεργοποίηση τρόπων λήψης αντιγράφων ασφαλείας
  - δυνατότητες υποστήριξης πολλαπλών χρηστών, και
  - επιπτώσεις από την αναδιοργάνωση της βάσης δεδομένων.



- Τα αποτελέσματα από τη φάση αυτή θα είναι:
  - οι **προδιαγραφές** του δομημένου φυσικού σχεδιασμού και
  - το βασικό **πρόγραμμα δοκιμών** του λογισμικού εφαρμογών.
- Στο φυσικό σχεδιασμό πρέπει να καθοριστεί κάθε λογικό τμήμα του λογισμικού εφαρμογών καθώς και τα σημεία επικοινωνίας μεταξύ τους.
- Επίσης, το νέο ΠΣ πρέπει να προσαρμοστεί στο διαθέσιμο υπολογιστικό σύστημα (αν υπάρχει) ή πρέπει να γίνει προμήθεια του κατάλληλου υλικού και λογισμικού συστήματος.
- Τέλος, πρέπει να προετοιμαστεί ένα σχέδιο στο οποίο να περιγράφονται οι απαιτήσεις για τη δοκιμή των εφαρμογών του ΠΣ, κατά την φάση της πραγμάτωσής τους σύμφωνα με την προσέγγιση από-το-γενικό-στο-ειδικό.

# Περιεχόμενα

- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Κατά τη φάση της λειτουργίας του ΠΣ:
  - Ολοκληρώνεται η πλήρης δοκιμή του λογισμικού εφαρμογών
  - Πραγματοποιείται η εκπαίδευση των χρηστών
  - Εκτελείται η διαδικασία μετάβασης από το υπάρχον ΠΣ στο νέο.
    - Εγκατάσταση του νέου υπολογιστικού συστήματος,
    - Εισαγωγή των νέων αυτοματοποιημένων μεθόδων λειτουργίας
    - Μετατροπή των παλαιών αρχείων δεδομένων (χειρόγραφων ή αυτοματοποιημένων) σε εκείνα που απαιτούνται για τη λειτουργία του νέου ΠΣ.

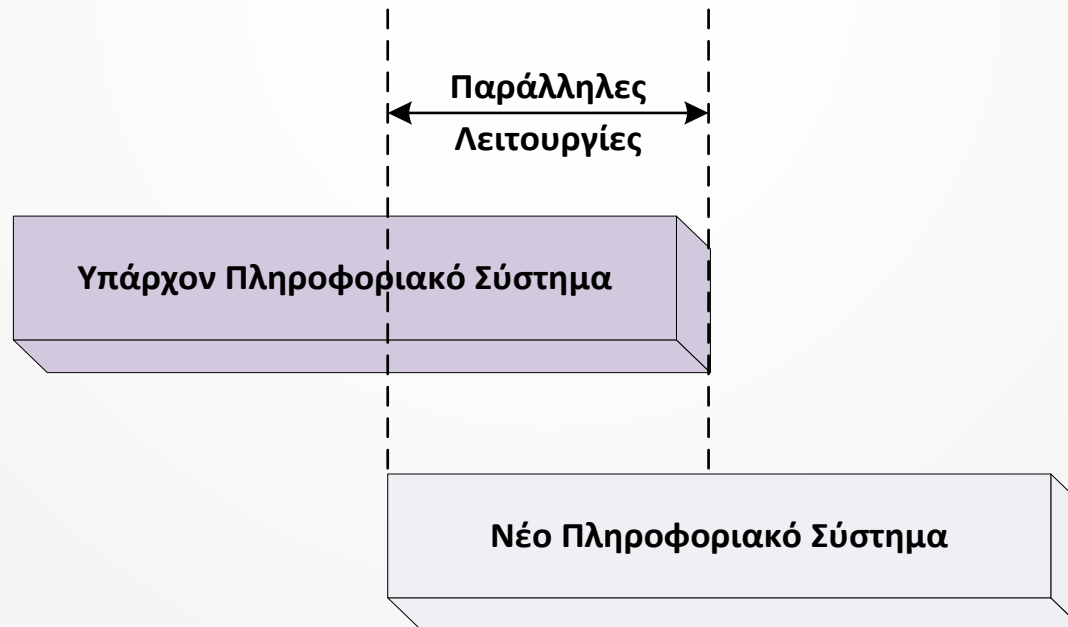
- Κατά τη φάση της λειτουργίας του ΠΣ παράγεται η ακόλουθη τεκμηρίωση υπο μορφή παραδοτέων:
  - Η τελική έκδοση του εγχειριδίου λειτουργίας
  - Το εγχειρίδιο συντήρησης
  - Η έκθεση αξιολόγησης

## ► Λειτουργικοί και μη-λειτουργικοί:

- Ο **έλεγχος χωρητικότητας** (*capacity*) με τον οποίο εξετάζεται αν είναι επαρκής η χωρητικότητα των μέσων δευτερεύουσας αποθήκευσης και του υπολογιστικού συστήματος και/ή των αρχείων.
- Ο **έλεγχος μεγίστου φόρτου** με τον οποίο εξετάζεται αν το ΠΣ μπορεί να ανταπεξέλθει στον όγκο των εργασιών που απαιτούνται σε περιόδους αιχμής. = **stress testing**
  - Ο **έλεγχος χρόνου απόκρισης** με τον οποίο προσδιορίζεται ο χρόνος που χρειάζεται το ΠΣ για να επεξεργαστεί δοσοληψίες δεδομένων διαφόρων τύπων. = **performance testing**
- Ο **έλεγχος ανάκαμψης** με τον οποίο εξετάζεται η ικανότητα των χρηστών να επαναλειτουργούν το υπολογιστικό σύστημα μετά από κάποια βλάβη. = **recovery testing**

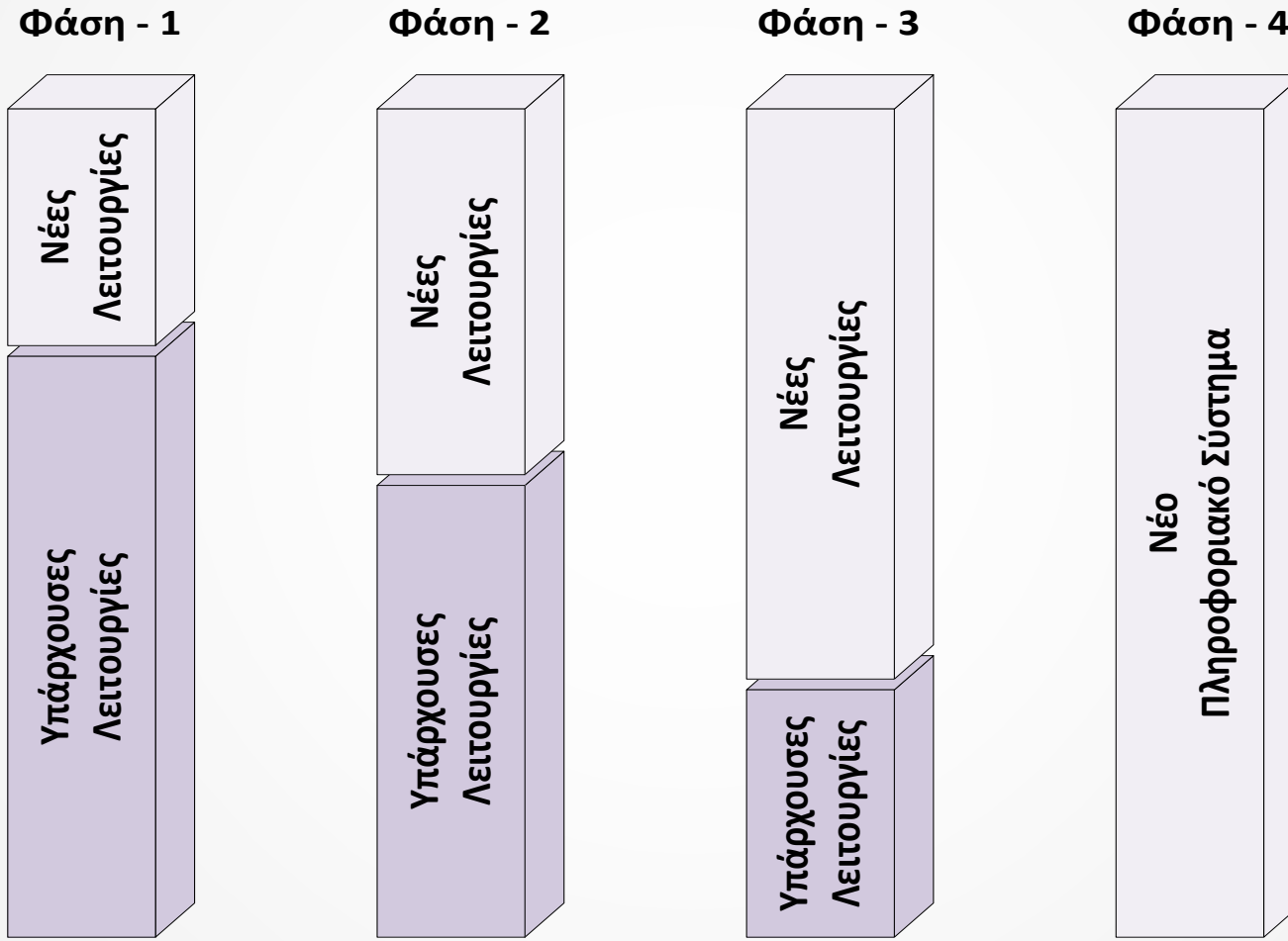
- Η διαδικασία μετάβασης από το υπάρχον πληροφοριακό σύστημα στο νέο μπορεί να γίνει με μία από τις ακόλουθες προσεγγίσεις:
  - την παράλληλη
  - την τμηματική
  - την πιλοτική
  - την άμεση
- Οι προσεγγίσεις αυτές δεν είναι αμοιβαία αποκλειόμενες και συνεπώς είναι δυνατό να χρησιμοποιηθεί κάποιος συνδυασμός τους.

- Goal: 0 downtime
- Data migration at the handover point
  - ▶ Stored procedures:
    - ETL (extract – transform – load)



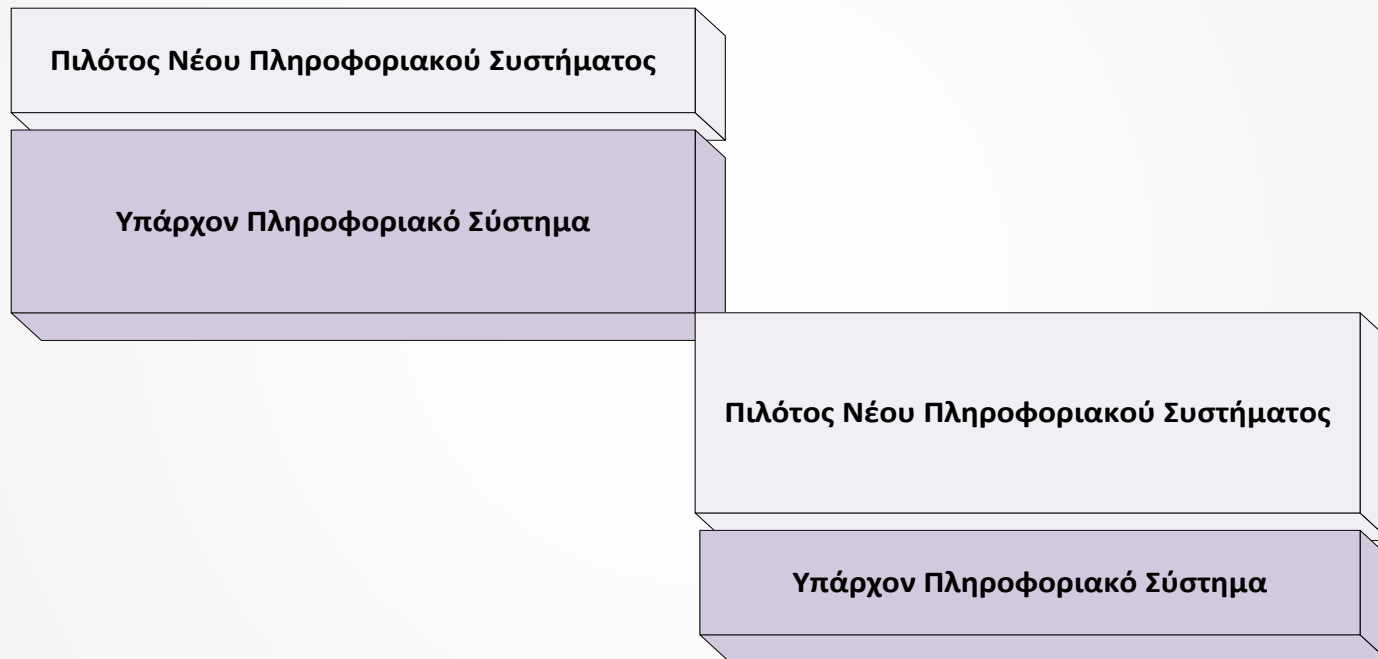
# Λειτουργία: Η τμηματική προσέγγιση μετάβασης από το υπάρχον στο νέο πληροφοριακό σύστημα

96



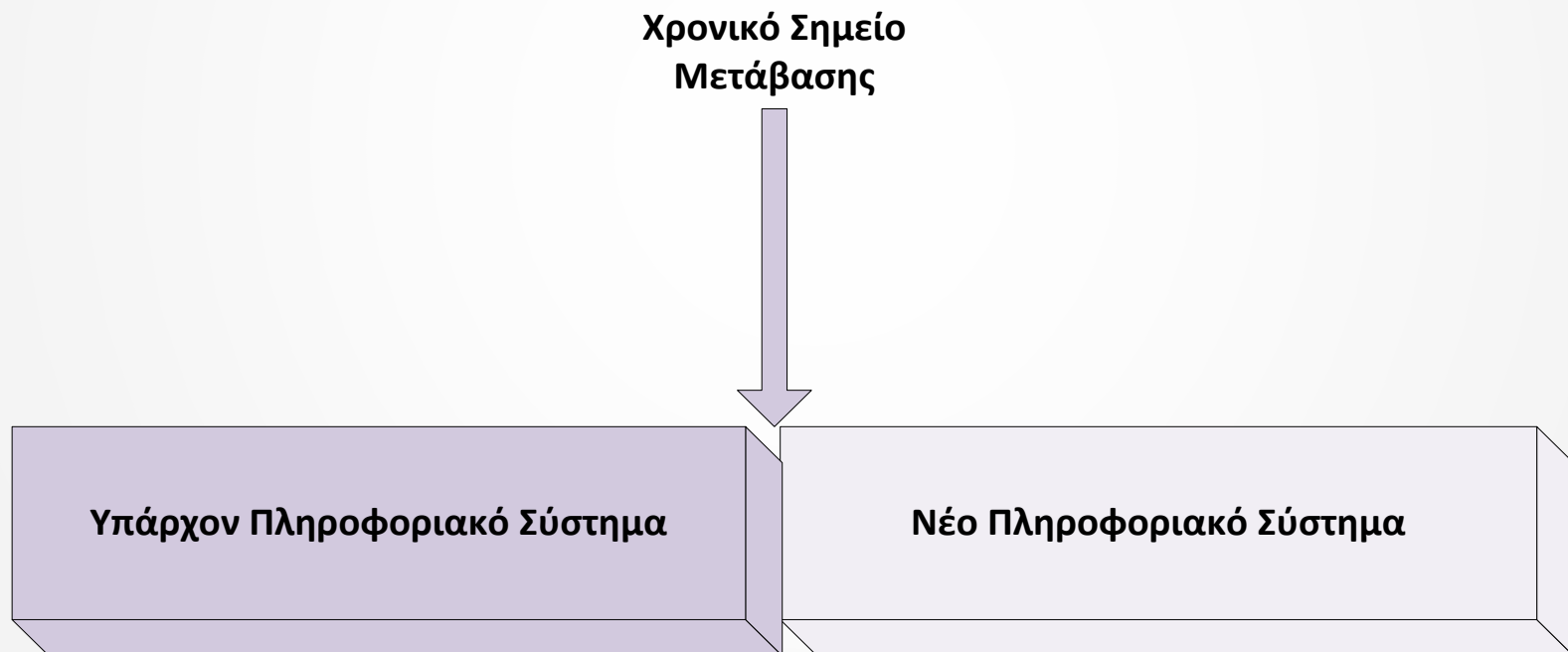


➡ Optimal for complex IS (e.g. ebanking)



➡ + 0 double maintenance

➡ - “slow start”



- Η συντήρηση του ΠΣ:
  - είναι είτε **προληπτική** είτε **επανορθωτική**
  - Περιλαμβάνει τη **συντήρηση του υλικού, του λογισμικού συστήματος και του λογισμικού εφαρμογών του ΠΣ.**
- Οι διορθώσεις προγραμματιστικών σφαλμάτων πραγματοποιούνται σε περιπτώσεις εμφάνισης σφαλμάτων μικρής κλίμακας (π.χ. δεν εκτυπώνεται ορθά η επικεφαλίδα μιας εκτυπωτικής κατάστασης).
  - Τα προβλήματα αυτά είναι καλά ορισμένα
  - Για την αντιμετώπιση των προβλημάτων αυτών δεν απαιτείται τροποποίηση της δομής του λογισμικού εφαρμογών. Δηλαδή, δεν απαιτείται κάποια διαδικασία εξαρχής ανάλυσης και σχεδιασμού.
  - Οι διορθώσεις πραγματοποιούνται αμέσως μετά τον εντοπισμό των σφαλμάτων.

## ➤ Ο εμπλουτισμός:

- Αφορά τη βελτίωση της αποδοτικότητας των εφαρμογών του υπάρχοντος πληροφοριακού συστήματος και/ή την ανάπτυξη νέων (π.χ. βελτίωση ενός υπολογιστικού αλγορίθμου ή προσθήκη της δυνατότητας για παροχή νέων αναφορών).
- Γίνεται σε χρονικά διαστήματα που επιλέγει ο οργανισμός, λαμβάνοντας υπόψη τις πληροφοριακές του ανάγκες και την απαιτούμενη δαπάνη για την ικανοποίησή τους. Έτσι, μπορεί να αναβάλλεται χρονικά χωρίς σοβαρές επιπτώσεις στη λειτουργία του υπάρχοντος πληροφοριακού συστήματος.

- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

Τα διαγράμματα ροής δεδομένων - ΔΡΔ (*Data Flow Diagrams - DFD*):

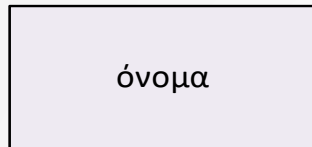
- Είναι μια από τις περισσότερο διαδεδομένες μεθόδους ή τεχνικές στα πλαίσια μεθοδολογιών δομημένης ανάλυσης και σχεδιασμού ΠΣ.
- Παρέχουν τη δυνατότητα αναπαράστασης των επιμέρους συστατικών ενός συστήματος και των σχέσεων μεταξύ τους.
- Μπορούν να χρησιμοποιηθούν για την αναπαράσταση της παρούσας και της προτεινόμενης λειτουργίας ενός ΠΣ κατά τρόπο εύληπτο προς του χρήστες που καλούνται να αποδεχτούν ή να απορρίψουν τις προτεινόμενες από τους αναλυτές λύσεις.

- Τα ΔΡΔ συνιστούν ένα εργαλείο για την αναπαράσταση ΠΣ χωρίς αναφορά σε υπερβολικές λεπτομέρειες που αναφέρονται είτε σε αυτοματοποιημένες είτε σε χειρόγραφες δραστηριότητες.
- Η μέθοδος ή τεχνική των ΔΡΔ:
  - Συνιστά μία από τις πιο σημαντικές του είδους της ακόμη και σήμερα που έχουν επινοηθεί και χρησιμοποιούνται ευρύτατα νέες μέθοδοι ή τεχνικές ανάλυσης και σχεδιασμού συστημάτων που βασίζονται στην αντικειμενοστρεφή προσέγγιση, όπως η ενοποιημένη γλώσσα μοντελοποίησης (*Unified Model Language – UML*).
  - Για την αναπαράσταση ενός υπάρχοντος (*as-is*) ή ενός προτεινόμενου (*to-be*) ΠΣ.

Γενικά, τα ΔΡΔ ως εργαλεία ανάλυσης και σχεδιασμού συστημάτων παρέχουν τη δυνατότητα επιτέλεσης των ακόλουθων καθοριστικών ενεργειών κατά τη διαδικασία εκτέλεσης ενός κύκλου ζωής ανάπτυξης ΠΣ:

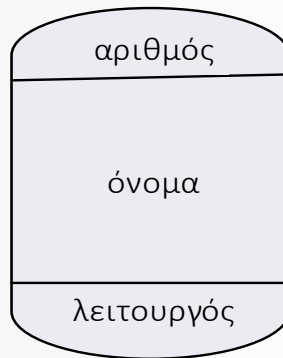
- Τη διάσπαση του συστήματος σε υποσυστήματα, λειτουργίες, εργασίες και επιχειρησιακές διεργασίες και την αναπαράσταση της δομής του σε ιεραρχικά δομημένες σχηματικές αναπαραστάσεις από-το-γενικό-προς-το-ειδικό.
- Την σχηματική τεκμηρίωση των διεπαφών μεταξύ των υποσυστημάτων, των λειτουργιών, των εργασιών και των επιχειρησιακών διεργασιών.
- Την τεκμηρίωση του ΠΣ και της χρήσης του κατά τις φάσεις του κύκλου ζωής που έχει επιλεγεί.





## Πηγή/προορισμός δεδομένων (Data source/destination):

Οντότητα που βρίσκεται εκτός του υπό ανάπτυξη ΠΣ (εξωτερική οντότητα) και η οποία διοχετεύει (δέχεται) δεδομένα προς (από) το σύστημα. Προσδιορίζει τα όρια του υπό ανάπτυξη ΠΣ.



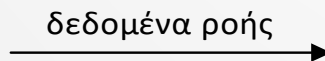
## Επεξεργασία δεδομένων (Process):

Διεργασία μετασχηματισμού των δεδομένων. Το σύμβολο διακρίνεται σε τρεις ενότητες – τον αριθμό της επεξεργασίας, το όνομα της επεξεργασίας και τον λειτουργό της επεξεργασίας (π.χ. φυσικό πρόσωπο ή ρόλο).



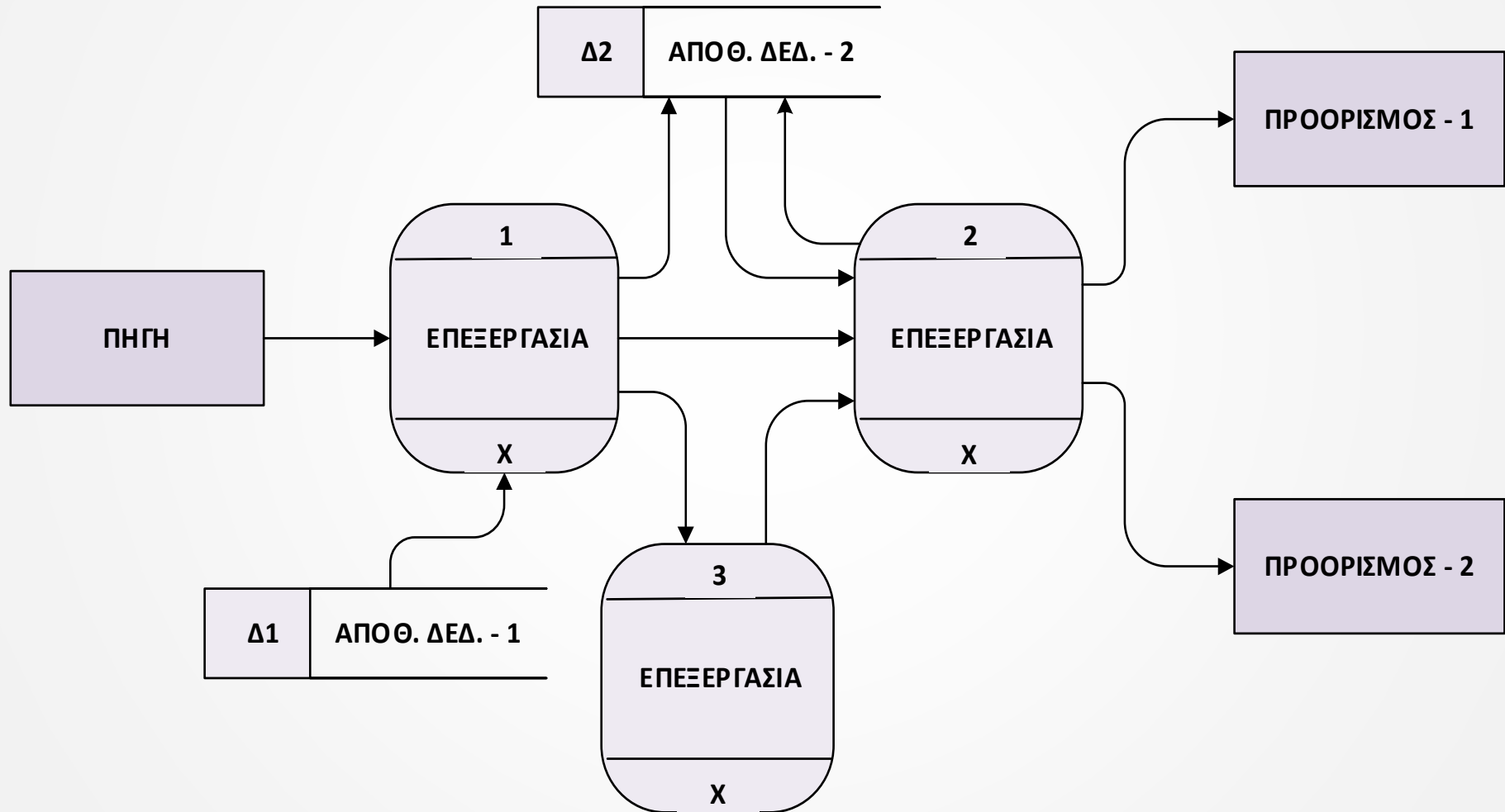
## Αποθηκεύσεις δεδομένων (Data store):

Συλλογή (ή αρχείο ή σύνολο αρχείων) δεδομένων που χρησιμοποιείται (read) ή ενημερώνεται (write) από επεξεργασίες. Το σύμβολο διακρίνεται σε δύο ενότητες – τον αριθμό της αποθήκευσης και το όνομα της αποθήκευσης.



## Ροή δεδομένων (Data flow):

Εισροή (εκροή) δεδομένων προς (από) επεξεργασίες ή εξωτερικές οντότητες. Επί του συμβόλου καταγράφονται τα ονόματα των διακινούμενων δεδομένων.

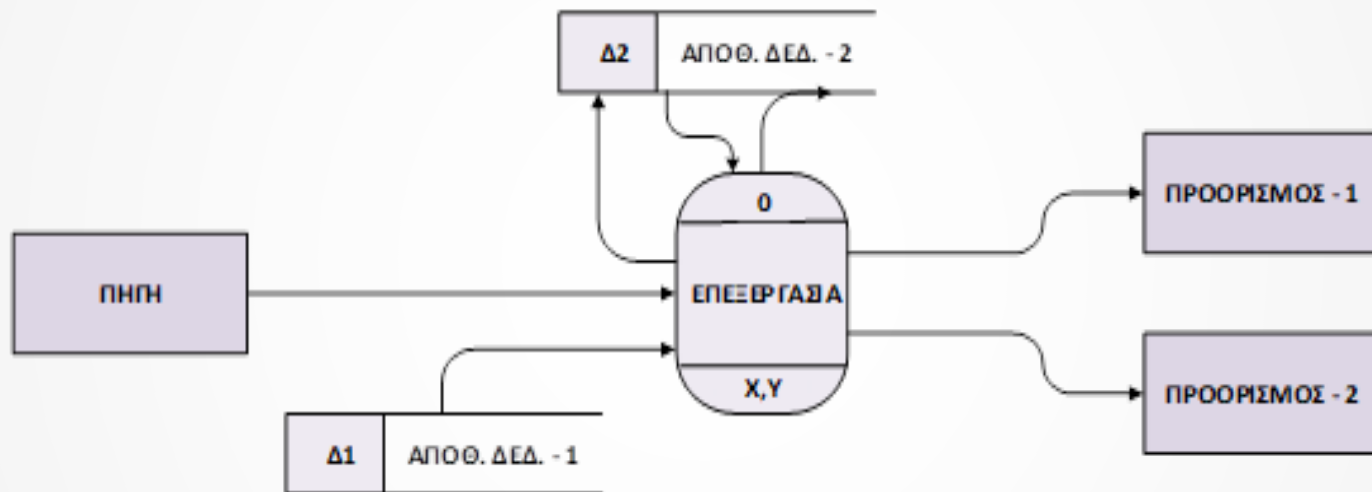


- Τα ΔΡΔ αποσκοπούν στην αναπαράσταση της λειτουργίας ενός συστήματος μέσω της αναπαράστασης των ροών των δεδομένων που παράγονται από ή χρησιμοποιούνται σε αυτό. Τα διαγράμματα ροών προγραμμάτων δείχνουν την ακολουθία των βημάτων που εκτελούνται σε έναν αλγόριθμο. Τα ΔΡΔ, αντίθετα από τα διαγράμματα ροών προγραμμάτων, δεν αναπαριστούν δομές τυχόν κυκλωτικών κινήσεων (loops) και δομές αποφάσεων αιτίου-αιτιατού (if-then-else).
- Τα ΔΡΔ μπορούν να αναπαριστούν επεξεργασίες που έχουν σημαντικά διαφορετικό χρονισμό.

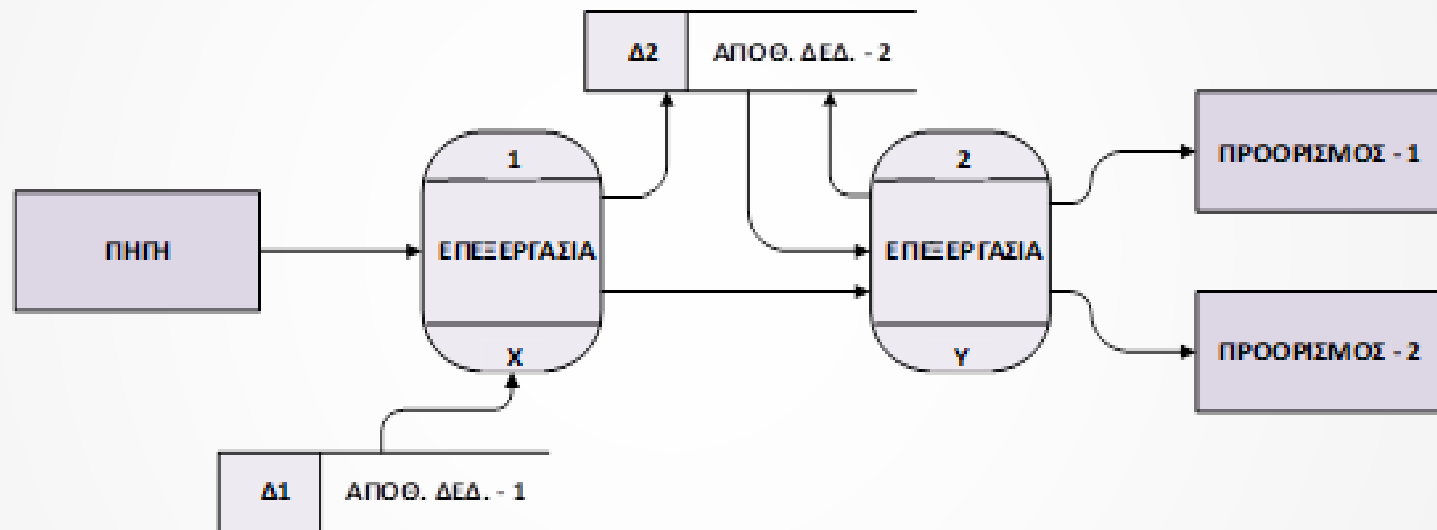
- Μια επεξεργασία δεδομένων μπορεί να δέχεται πολλές εισροές και να παράγει πολλές εκροές δεδομένων. Η συχνότητα εμφάνισης αυτών μπορεί να ποικίλει.
- Μερικές εκροές δεδομένων μπορεί να εμφανίζονται (παράγονται) κάθε φορά που εκτελείται η επεξεργασία ενώ άλλες μπορεί να εμφανίζονται προαιρετικά υπό συγκεκριμένες συνθήκες.
- Τα ΔΡΔ δεν καταδεικνύουν ποιες εκροές δεδομένων είναι υποχρεωτικές και επιθυμητές και ποιες είναι προαιρετικές ή με ποιους συνδυασμούς εμφανίζονται. Οι λεπτομέρειες αυτές δεν αναφέρονται και δεν είναι δυνατόν να αναπαρασταθούν μέχρι να προσδιοριστούν οι εσωτερικές επιμέρους διαδικασίες των επεξεργασιών.

- Δύο συχνά σφάλματα αναπαράστασης ενός συστήματος με ΔΡΔ είναι:
  - η ύπαρξη επεξεργασιών που παράγουν εκροές δεδομένων χωρίς να δέχονται εμφανώς εισροές δεδομένων και
  - η ύπαρξη επεξεργασιών που δεν παράγουν εκροές δεδομένων παρά το γεγονός ότι δέχονται εισροές δεδομένων.

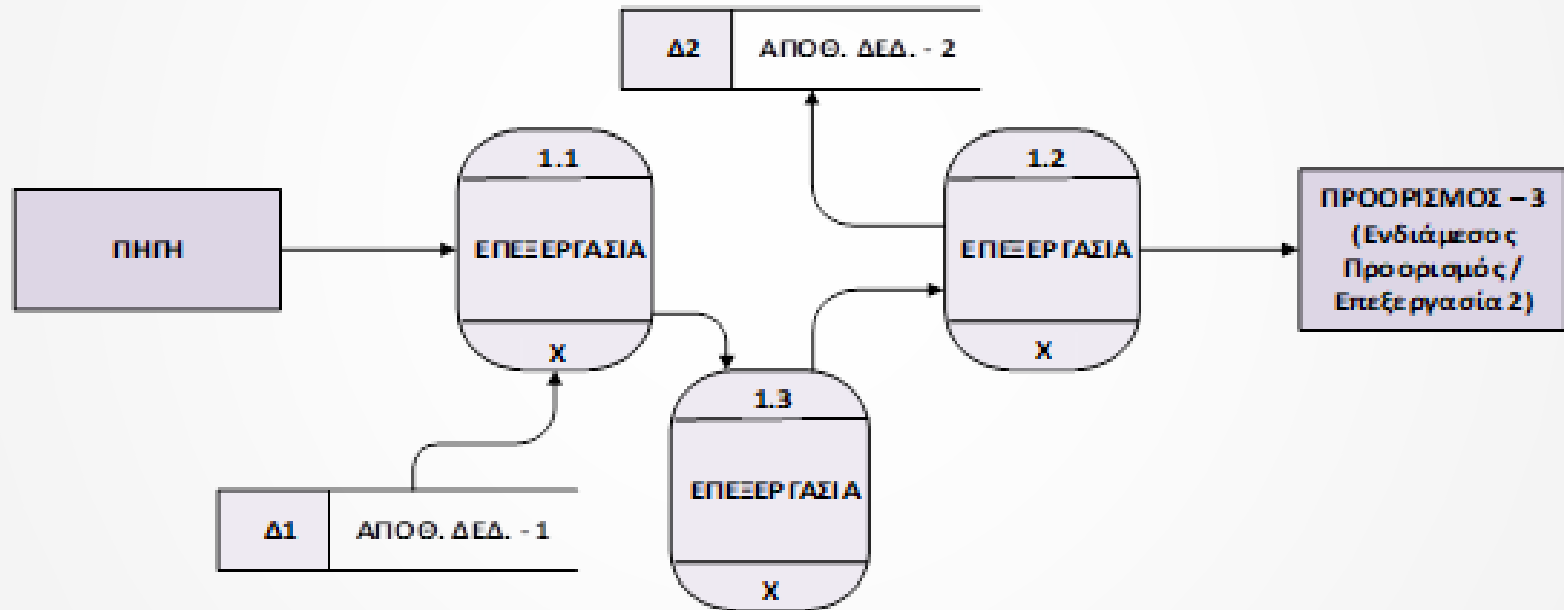
ΕΠΙΠΕΔΟ - 0: Η επεξεργασία 0 αναφέρεται σε ολόκληρο το σύστημα



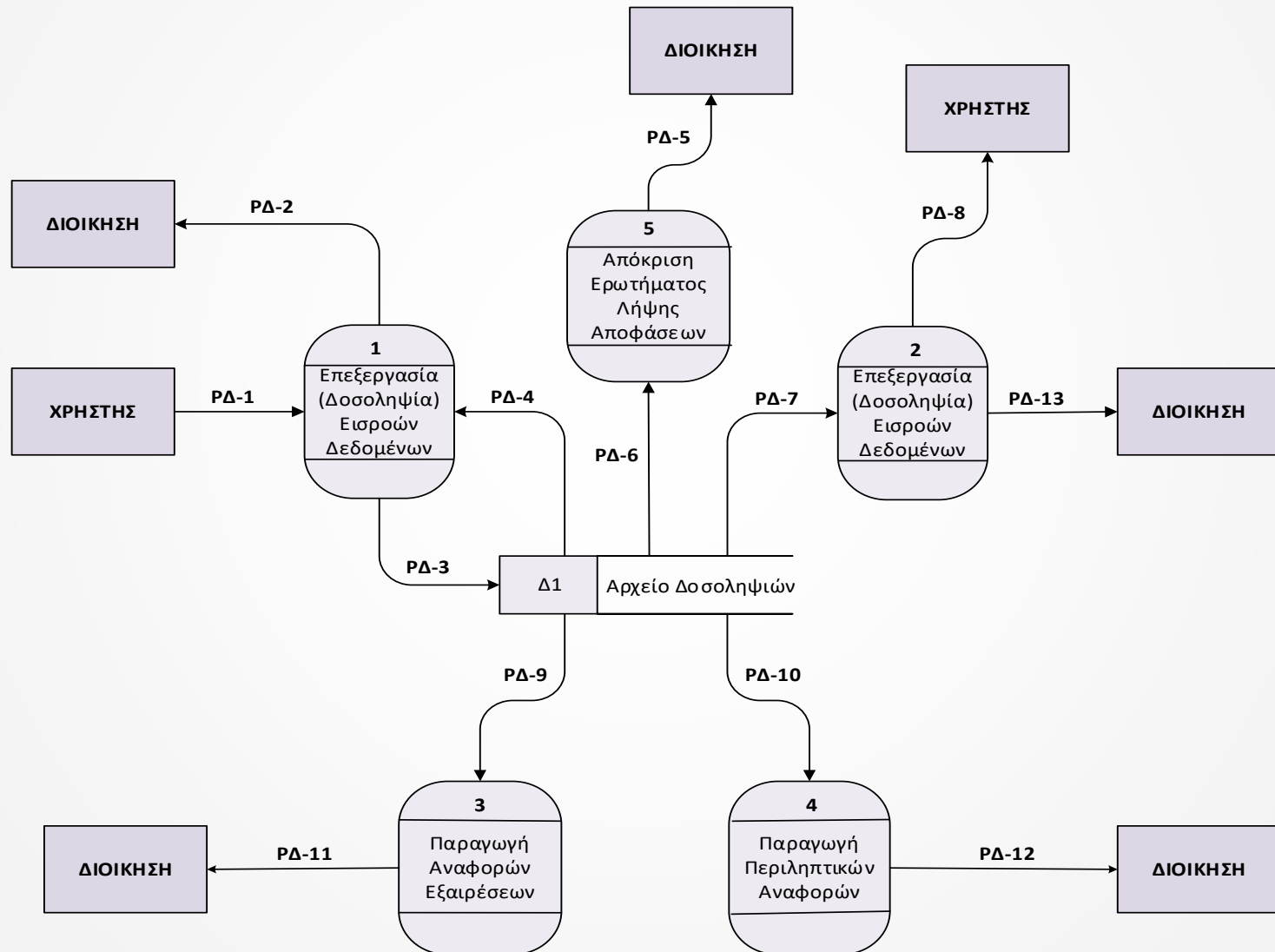
ΕΠΙΓΕΔΟ - 1: Διάσπαση της επεξεργασίας 0 στις επεξεργασίες 1 & 2



ΕΠΙΠΕΔΟ - 2: Διάσπαση της επεξεργασίας 1 στις επεξεργασίες 1.1, 1.2 & 1.3







- Παρόλο που τα ΔΡΔ αναπαριστούν όλες τις συνιστώσες ενός ΠΣ, η αναπαράσταση αυτή πραγματοποιείται σε σχετικά υψηλό επίπεδο αφαίρεσης (χαμηλό επίπεδο λεπτομέρειας).
  - Για παράδειγμα, δεν εμφανίζεται το συγκεκριμένο περιεχόμενο των ροών δεδομένων και των αποθηκεύσεων δεδομένων καθώς και οι ακριβείς πολιτικές και διαδικασίες που πραγματώνουν οι επεξεργασίες.
- Για την διαγραμματική αναπαράσταση των δομών των αποθηκεύσεων δεδομένων, δηλαδή της δομής της βάσης δεδομένων του ΠΣ, χρησιμοποιούνται άλλες τεχνικές όπως τα διαγράμματα οντοτήτων-συσχετίσεων (*entity-relationship diagrams*).

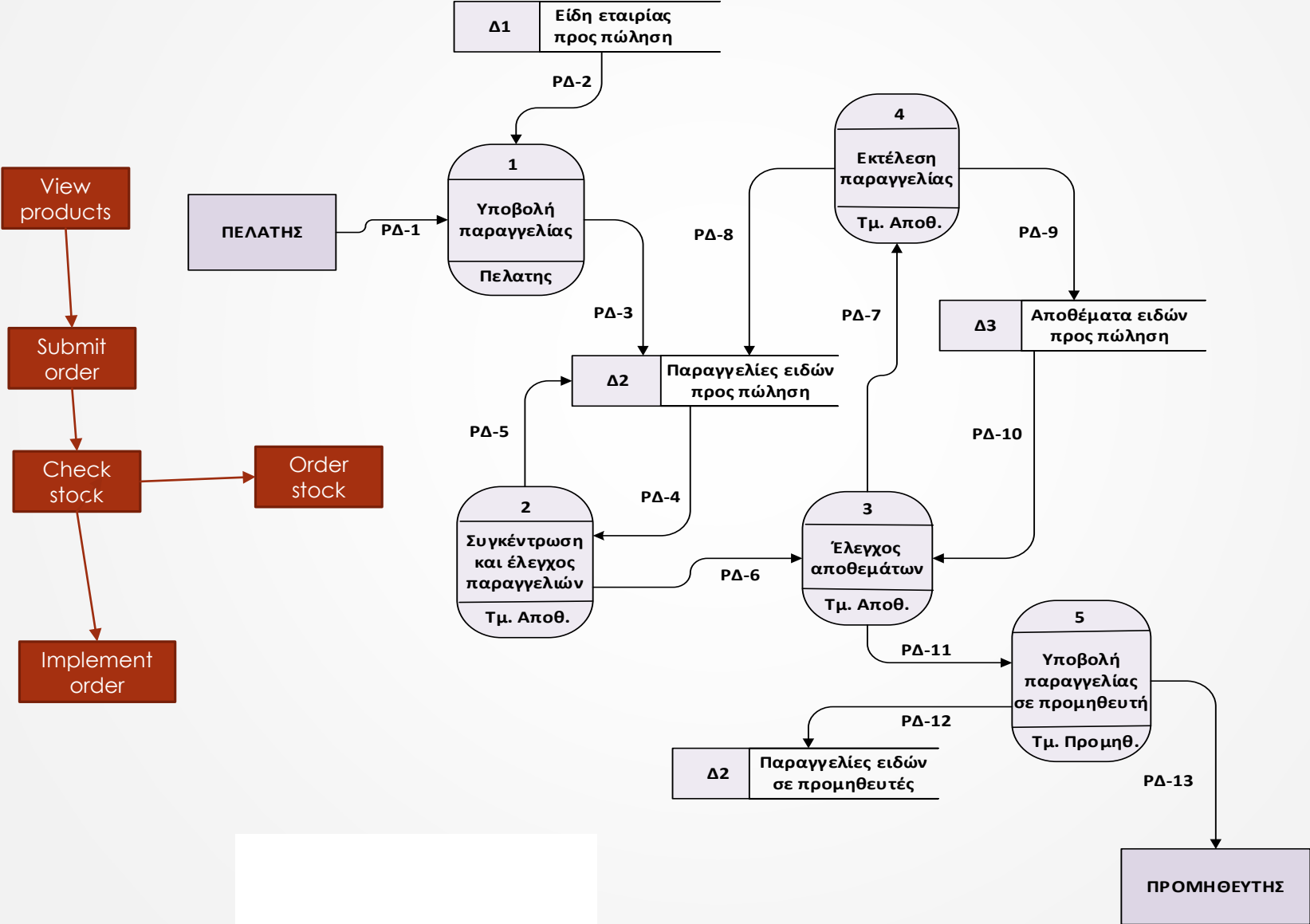
## Διαδικασία διαχείρισης παραγγελιών εμπορικής εταιρίας

*Σε μια εμπορική εταιρία, οι παραγγελίες των πελατών υποβάλλονται μέσω του διαδικτύου.*

*Κάθε παραγγελία υποβάλλεται με τη συμπλήρωση ενός ηλεκτρονικού εντύπου που υπάρχει στον ιστότοπο της εταιρίας και την καταχώρησή του στο αρχείο εκκρεμών παραγγελιών.*

*Προκειμένου να εκτελεστεί μια παραγγελία, το προσωπικό της αποθήκης ελέγχει τα αποθέματα για να αποφανθεί αν υπάρχουν τα αιτούμενα είδη στις αιτούμενες ποσότητες.*

*Σε καταφατική περίπτωση εκτελείται η παραγγελία ενώ σε αρνητική περίπτωση παραγγέλλονται τα ελλειμματικά είδη στις ποσότητες που προσδιορίζονται από την  $(s, S)$  πολιτική της εταιρίας ανά είδος (η πολιτική αυτή σημαίνει ότι όταν το απόθεμα του είδους πέσει κάτω από  $s$  παραγγέλλεται ποσότητα τόση ώστε το απόθεμα να γίνει  $S$ ).*



## Ροές Δεδομένων (ΡΔ)

**ΡΔ-1:** Στοιχεία πελάτη (καταγραφή)

**ΡΔ-2:** Είδη εταιρίας (ανάκτηση)

**ΡΔ-3:** Παραγγελία (καταγραφή)

**ΡΔ-4:** Παραγγελίες (ανάκτηση)

**ΡΔ-5:** Επιβεβαίωση ορθότητας παραγγελίας (καταγραφή)

**ΡΔ-6:** Εκκρεμείς παραγγελίες (διαβίβαση)

**ΡΔ-7:** Εκκρεμείς ελεγμένες παραγγελίες (διαβίβαση)

**ΡΔ-8:** Επιβεβαίωση εκτέλεσης παραγγελίας (καταγραφή)

**ΡΔ-9:** Ενημέρωση αποθέματος (καταγραφή)

**ΡΔ-10:** Αποθέματα παραγγελθέντων ειδών (ανάκτηση)

**ΡΔ-11:** Διαπίστωση ελλείμματος αποθέματος για συγκεκριμένα είδη

**ΡΔ-12:** Παραγγελία σε προμηθευτή για αναπλήρωση αποθέματος (καταγραφή)

**ΡΔ-13:** Παραγγελία σε προμηθευτή για αναπλήρωση αποθέματος (αποστολή)

Για την κατασκευή ενός ΔΡΔ, αρχικά:

- Εντοπίζονται οι τυπικές ροές δεδομένων που συνιστούν τις εισροές και τις εκροές του συστήματος από και προς το περιβάλλον του, αντίστοιχα.
- Οι ροές αυτές αναπαριστώνται στο αρχικό ή γενικό ΔΡΔ, μηδενικού (όπως συχνά χαρακτηρίζεται) επιπέδου λεπτομέρειας (υψηλότετου δυνατού επιπέδου αφαίρεσης). Δηλαδή, το αρχικό ή γενικό ΔΡΔ αναπαριστά μόνο τα όρια του υπό μελέτη συστήματος και τις εισροές και εκροές δεδομένων από και προς αυτά.

Στο επόμενο (πρώτο) επίπεδο λεπτομέρειας του συστήματος

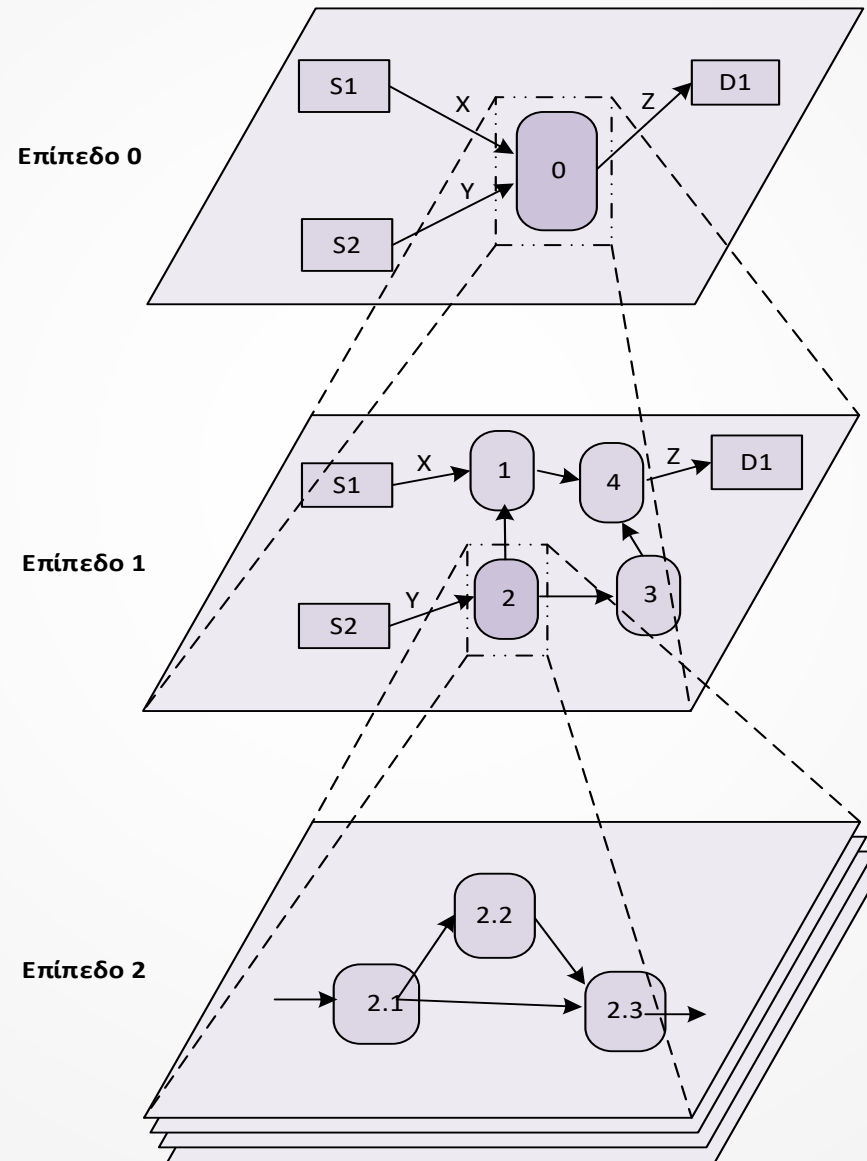
- Περιγράφονται οι **κύριες επεξεργασίες του συστήματος** καθώς και οι **ροές και οι αποθηκεύσεις δεδομένων** που σχετίζονται με αυτές. Έτσι, σχηματίζεται η πρώτη γενική θεώρηση του συστήματος (ΔΡΔ πρώτου επιπέδου λεπτομέρειας).
- Κάθε επεξεργασία του ΔΡΔ πρώτου επιπέδου λεπτομέρειας, ή ενός ΔΡΔ οποιουδήποτε ανωτέρου επιπέδου λεπτομέρειας, διασπάται για την δημιουργία ενός νέου ΔΡΔ ακόμη ανωτέρου επιπέδου λεπτομέρειας. Αυτό αναπαριστά μία πιο λεπτομερή θεώρηση κάθε επεξεργασίας και των εισροών και εκροών δεδομένων.

- Σε κάθε επίπεδο ενός ΔΡΔ περιγράφονται συνήθως πέντε έως επτά επεξεργασίες προκειμένου το διάγραμμα να είναι κατανοητό από τους χρήστες του συστήματος.
- Σε γενικές γραμμές, τα βασικά βήματα για την κατασκευή ενός ΔΡΔ είναι:
  - i. Ο καθορισμός των πηγών και των προορισμών του των ροών δεδομένων του συστήματος (που συνιστούν το περιβάλλον του συστήματος).
  - ii. Ο καθορισμός των εισροών και εκροών του συστήματος και ο σχεδιασμός του αρχικού ΔΡΔ μηδενικού επιπέδου.



- iii. Στη συνέχεια σχεδιάζονται οι πηγές και οι τελικοί προορισμοί των ροών δεδομένων του συστήματος (με φορά από τα αριστερά προς τα δεξιά), οι επεξεργασίες και οι αποθηκεύσεις των δεδομένων. Εναλλακτικά, η διαδικασία μπορεί να αρχίσει από την αναπαράσταση των επεξεργασιών και να συνεχιστεί με την αναπαράσταση των πηγών και των προορισμών που έχουν ήδη καθοριστεί στο βήμα (i).
- iv. Η ονομασία των ροών, αποθηκεύσεων και επεξεργασιών των δεδομένων.
- v. Η ολοκλήρωση ενός προκαταρκτικού ΔΡΔ και ο έλεγχος των εισροών και εκροών που καθορίστηκαν στο βήμα (ii).

- vi. Η ανασκόπηση του ΔΡΔ με τη συμμετοχή των χρηστών προκειμένου να καθοριστεί εάν αναπαρίσταται σωστά το υπό μελέτη σύστημα (είτε το υπάρχον είτε το υπό δημιουργία). Εάν όχι, πραγματοποιούνται οι αναγκαίες τροποποιήσεις.
- vii. Η διάσπαση κάθε επεξεργασίας του ΔΡΔ ενός επιπέδου για τη δημιουργία του ΔΡΔ υψηλότερου επιπέδου λεπτομέρειας (εφόσον είναι εφικτό). Κατά τη διάσπαση αυτή πρέπει να χρησιμοποιείται ιεραρχική αρίθμηση των επεξεργασιών του ΔΡΔ. Εάν απαιτείται, τα ΔΡΔ ανωτέρων επιπέδων επανεξετάζονται και πραγματοποιούνται οι αναγκαίες τροποποιήσεις.
- viii. Η επανάληψη του βήματος (vii) για κάθε επεξεργασία του ΔΡΔ μέχρις ότου κάθε επεξεργασία να χρησιμοποιεί και να παράγει τις εισροές και τις εκροές, αντίστοιχα, στο αντίστοιχο επίπεδο λεπτομέρειας.



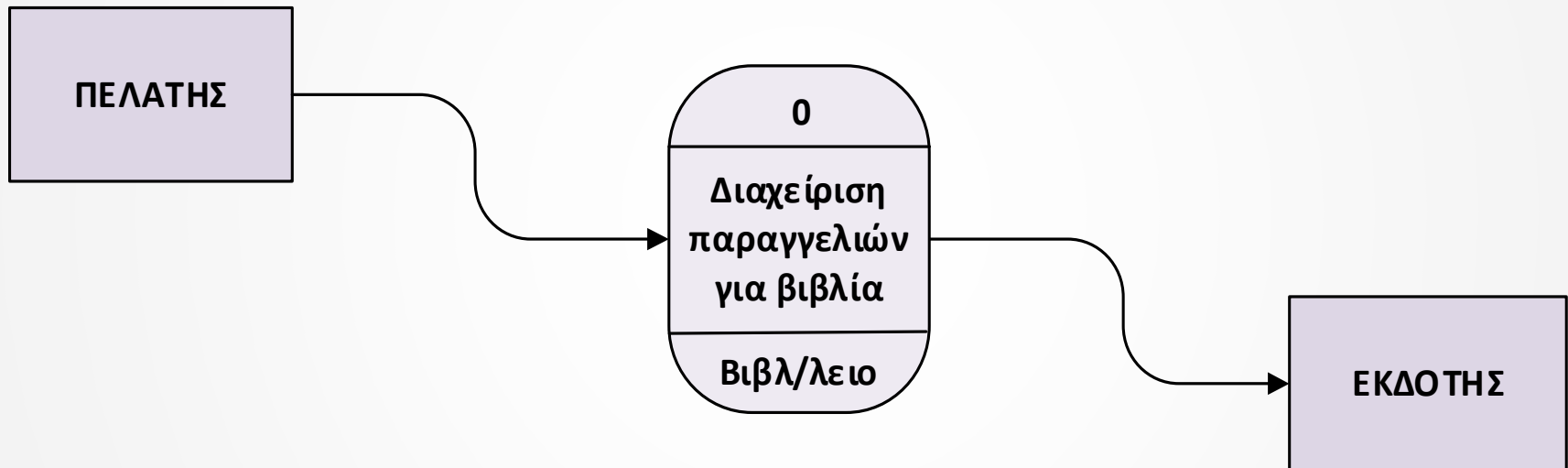
Για τη βελτίωση της δυνατότητας αναπαράστασης ενός ΠΣ μέσω ΔΡΔ μπορεί να χρησιμοποιηθούν εκφράσεις του τύπου  $A \theta B$ , όπου  $A$  και  $B$  είναι δύο επεξεργασίες και  $\theta$  μπορεί να λάβει τις Boolean τιμές AND, OR και XOR. Έτσι, στο παράδειγμα που αφορά **τη διαχείριση παραγγελιών εμπορικής εταιρίας**, από το αποτέλεσμα της επεξεργασίας με αριθμό 3 (“Έλεγχος αποθεμάτων”) μπορεί να επακολουθήσει:

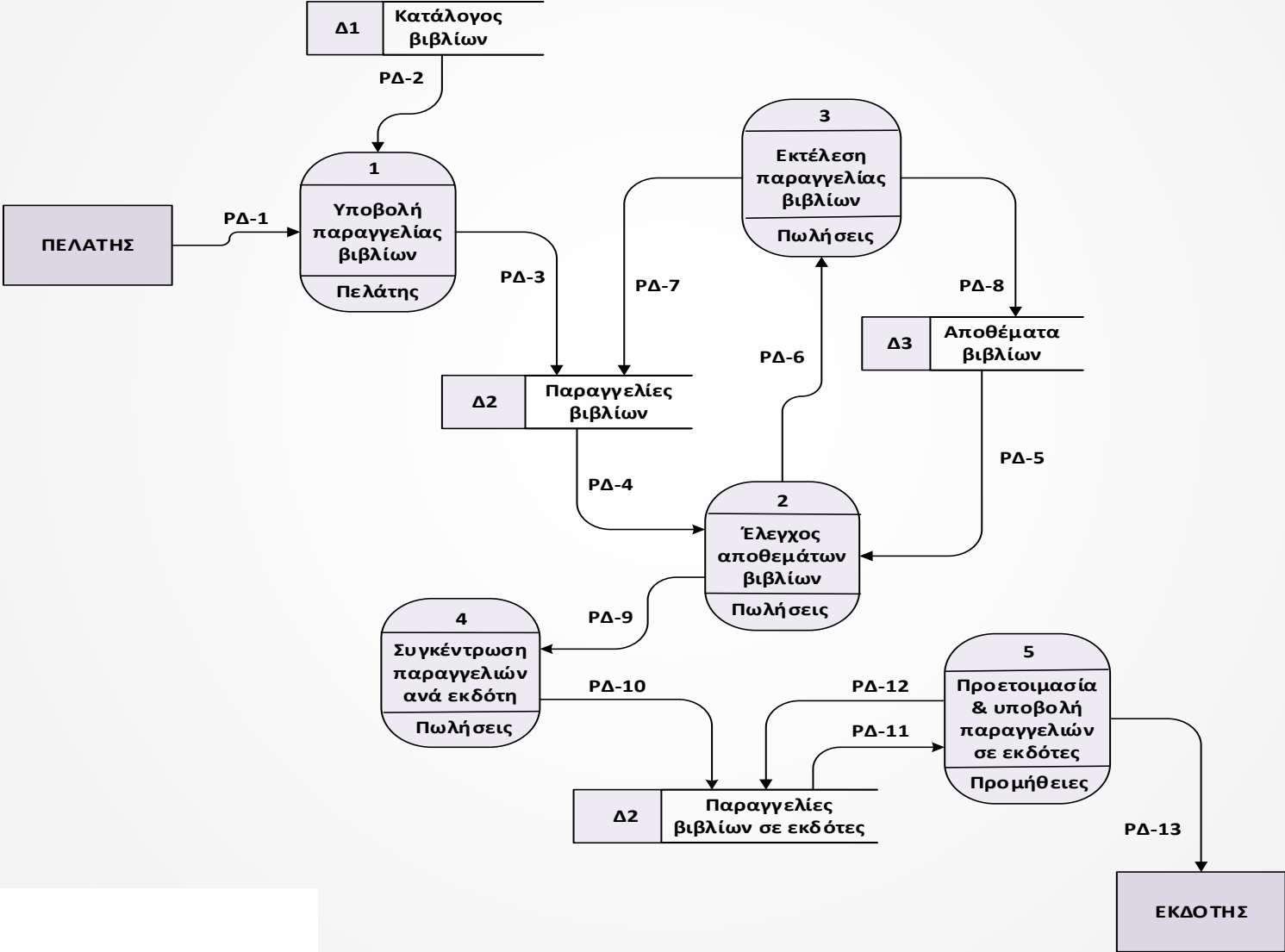
- i. Μόνο η ροή ΡΔ-7 και, στη συνέχεια, η εκτέλεση της επεξεργασίας με αριθμό 4 (“Εκτέλεση παραγγελίας”), εφόσον επαρκεί το υπάρχον απόθεμα και δεν παραβιάζεται η  $(s, S)$  πολιτική της εταιρίας, ή
- ii. Μόνο η ροή ΡΔ-11 και, στη συνέχεια, η εκτέλεση της επεξεργασίας με αριθμό 5 (“Υποβολή παραγγελίας σε προμηθευτή”), εφόσον δεν επαρκεί το υπάρχον απόθεμα (στην περίπτωση που η αιτούμενη ποσότητα προϊόντος υπερβαίνει το  $S$ ), ή

- iii. Και οι δύο ροές και εκτελέσεις επεξεργασιών, εφόσον επαρκεί το υπάρχον απόθεμα αλλά παραβιάζεται η  $(s, S)$  πολιτική της εταιρίας με την εκτέλεση της παραγγελίας.

### Διαχείρισης παραγγελιών βιβλίων ενός βιβλιοπωλείου

Ένα βιβλιοπωλείο δέχεται ηλεκτρονικές παραγγελίες για βιβλία από διάφορους πελάτες (π.χ. φυσικά πρόσωπα και βιβλιοθήκες). Για την εκτέλεση των παραγγελιών, το βιβλιοπωλείο συγκεντρώνει τις παραγγελίες των πελατών ανά εκδότη, τις διαβιβάζει προς αυτούς, παραλαμβάνει τα αντίστοιχα βιβλία και τα παραδίδει στους πελάτες. Η διαδικασία που αφορά στη διαχείριση των παραγγελιών βιβλίων από τους πελάτες του βιβλιοπωλείου φαίνεται στα ιεραρχικά δομημένα (από-το-γενικό-στο-ειδικό) ΔΡΔ των ακόλουθων σχημάτων.

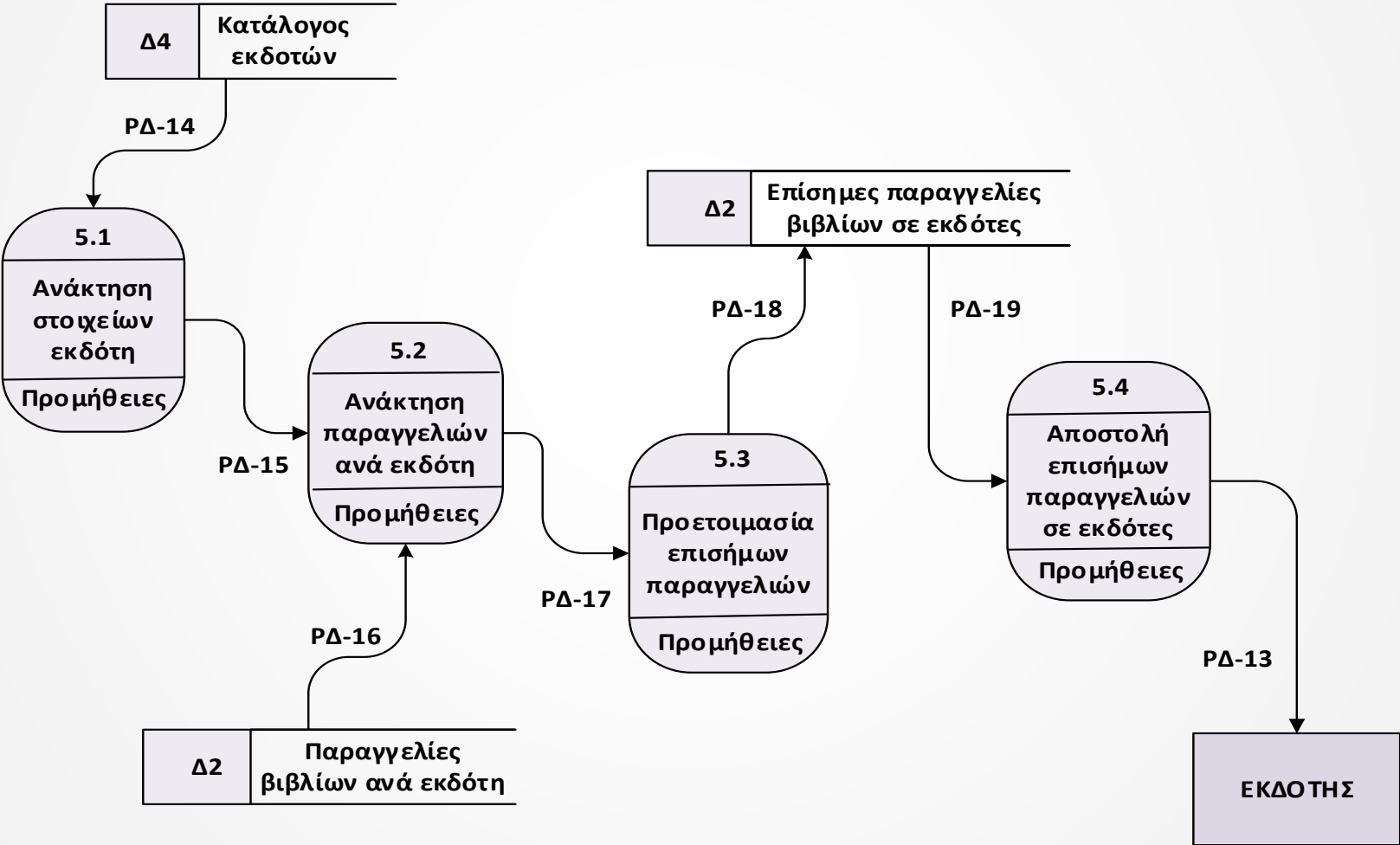






### Ροές Δεδομένων (ΡΔ)

- ΡΔ-1:** Στοιχεία πελάτη (καταγραφή)
- ΡΔ-2:** Διαθέσιμα βιβλία (ανάκτηση)
- ΡΔ-3:** Παραγγελία βιβλίων (καταγραφή)
- ΡΔ-4:** Παραγγελίες βιβλίων (ανάκτηση)
- ΡΔ-5:** Αποθέματα παραγγελθέντων βιβλίων (ανάκτηση)
- ΡΔ-6:** Επιβεβαίωση ύπαρξης ικανών αποθεμάτων (διαβίβαση)
- ΡΔ-7:** Ενημέρωση εκτέλεσης παραγγελίας (καταγραφή)
- ΡΔ-8:** Ενημέρωση αποθέματος με διατεθέντα βιβλία (καταγραφή)
- ΡΔ-9:** Διαπίστωση ελλειματικών αποθεμάτων για συγκεκριμένα βιβλία (διαβίβαση)
- ΡΔ-10:** Παραγγελίες βιβλίων ανά εκδότη (καταγραφή)
- ΡΔ-11:** Παραγγελίες βιβλίων ανά εκδότη (ανάκτηση)
- ΡΔ-12:** Παραγγελία σε εκδότη για αναπλήρωση αποθέματος (καταγραφή)
- ΡΔ-13:** Παραγγελία σε εκδότη για αναπλήρωση αποθέματος (αποστολή)



**Ροές Δεδομένων (ΡΔ)**

**ΡΔ-14:** Στοιχεία εκδότη (ανάκτηση)

**ΡΔ-15:** Στοιχεία εκδότη (διαβίβαση)

**ΡΔ-16:** Παραγγελίες βιβλίων ανά εκδότη (ανάκτηση)

**ΡΔ-17:** Παραγγελίες βιβλίων ανά εκδότη (διαβίβαση)

**ΡΔ-18:** Επίσημες παραγγελίες βιβλίων ανά εκδότη (καταγραφή)

**ΡΔ-19:** Επίσημες παραγγελίες βιβλίων ανά εκδότη (ανάκτηση)

**ΡΔ-13:** Αποστολή επισήμων παραγγελιών σε εκδότη (αποστολή)

## Διαχείριση φαρμακευτικών συνταγών

Έστω μια διαδικασία διαχείρισης φαρμακευτικών συνταγών πρωτοβάθμιας φροντίδας υγείας από ασφαλιστικούς φορείς που χρησιμοποιεί “ηλεκτρονικό βιβλιάριο” συγγραφής συνταγών υπό μορφή έξυπνης κάρτας (*smart card*). Στη έξυπνη κάρτα έχουν καταγραφεί τα προσωπικά στοιχεία και τα κρίσιμα στοιχεία υγείας κάθε ασφαλισμένου (τα οποία είναι δυνατόν να ενημερώνονται). Η διαδικασία διαχείρισης φαρμακευτικών συνταγών αφορά στη συγγραφή, στον έλεγχο και στην εκτέλεση συνταγών.

Οι κύριες λειτουργίες των τριών βασικών υποσυστημάτων του συστήματος μπορεί, ανάλογα με το σχεδιασμό, να είναι οι ακόλουθες:

## *1. Θεράπων Ιατρός: Συγγραφή Συνταγών*

- Ο ασφαλισμένος είτε επισκέπτεται εκούσια κάποιον γιατρό είτε διακομίζεται σε κάποιον γιατρό είτε ο γιατρός επισκέπτεται τον ασφαλισμένο κατ' οίκον. Η έξυπνη κάρτα του ασφαλισμένου παραδίδεται στο γιατρό (ΡΔ-1) ο οποίος την εισάγει στον αναγνώστη έξυπνων καρτών του υπολογιστή του που είναι συνδεδεμένος στο αντίστοιχο διαδικτυακό ΠΣ.
- Σε περίπτωση επιτυχούς αυθεντικοποίησης ο γιατρός μπορεί, εφόσον το επιθυμεί, να ανακτήσει τα προσωπικά και ιατρικά στοιχεία του ασθενούς που είναι αποθηκευμένα στην έξυπνη κάρτα του (ΡΔ-2).

- Ο γιατρός ανακτά και επιλέγει τα απαιτούμενα δεδομένα (π.χ. κωδικοποιήσεις διαγνώσεων, φάρμακα) (ΡΔ-3) και καταγράφει τα ιατρικά στοιχεία του ασφαλισμένου (νέους κρίσιμους παράγοντες υγείας, διαγνώσεις, φάρμακα) που αφορούν στη συγκεκριμένη επίσκεψη (ΡΔ-4). Τα ιατρικά στοιχεία καταγράφονται στην Κεντρική Βάση Δεδομένων του Ασφαλιστικού Φορέα ενώ, ταυτόχρονα, καταγράφονται μόνο τα στοιχεία της φαρμακευτικής συνταγής στην έξυπνη κάρτα του.
- Αφού ελεγχθεί η καταγραφή των στοιχείων παραδίδεται η έξυπνη κάρτα στον ασφαλισμένο ή τον συγγενή του.

## *II. Ελεγκτής Ιατρός: Έλεγχος Συνταγών*

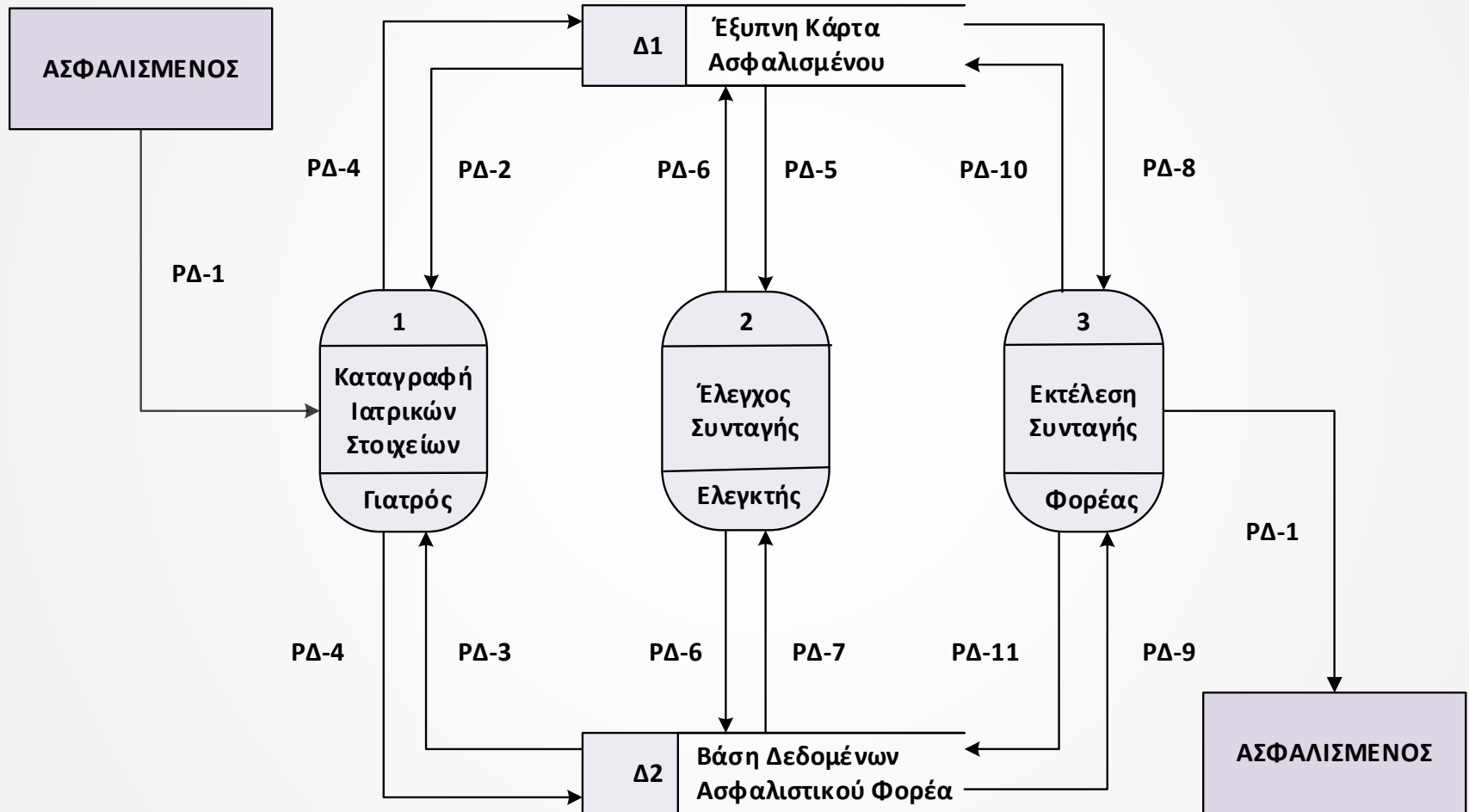
- Ο ασφαλισμένος ή συγγενής του παραδίδει την έξυπνη κάρτα του στον ελεγκτή γιατρό και ακολουθεί η διαδικασία αυθεντικοποίησης.
- Σε περίπτωση επιτυχούς αυθεντικοποίησης, ο ελεγκτής γιατρός διαβάζει τα στοιχεία που έχουν καταγραφεί στην έξυπνη κάρτα του ασφαλισμένου (ΡΔ-5). Επίσης, διαβάζει του τυχόν περιορισμούς του Ασφαλιστικού Φορέα αναφορικά με τη συνταγογράφηση (ΡΔ-7).
- Κατόπιν, ο ελεγκτής γιατρός εγκρίνει ή απορρίπτει τα φάρμακα και υπογράφει ψηφιακά (ΡΔ-6). Οι αντίστοιχες εγγραφές γίνονται στην κεντρική βάση δεδομένων του Ασφαλιστικού Φορέα και στην έξυπνη κάρτα του ταυτόχρονα.
- Αφού ελεγχθεί η ορθότητα της καταγραφής των στοιχείων παραδίδεται η έξυπνη κάρτα στον ασφαλισμένο ή στον συγγενή του.

## *III. Φαρμακείο: Εκτέλεση Συνταγών*

- Ο ασφαλισμένος ή συγγενής του παραδίδει την έξυπνη κάρτα του στον φαρμακοποιό και ακολουθεί η διαδικασία αυθεντικοποίησης τους.
- Σε περίπτωση επιτυχούς αυθεντικοποίησης, ο φαρμακοποιός διαβάζει από την έξυπνη κάρτα του ασθενούς μόνο τα στοιχεία που τον αφορούν για την εκτέλεση της συνταγής (δεν έχει πρόσβαση στα λοιπά) και εκτελεί τη συνταγή (ΡΔ-8). Ταυτόχρονα ανακτώνται τα στοιχεία της συνταγής και από τη βάση δεδομένων του Ασφαλιστικού φορέα (ΡΔ-9)
- Διαγράφονται από την έξυπνη κάρτα του ασθενούς τα φάρμακα της συνταγής που χορηγήθηκαν (ΡΔ-10) ενώ, ταυτόχρονα, καταγράφεται στην Κεντρική Βάση Δεδομένων του Ασφαλιστικού Φορέα η εκτέλεση της συνταγής (ΡΔ-11).



- Αφού ελεγχθεί η ορθότητα της καταγραφής των στοιχείων παραδίδεται η έξυπνη κάρτα του στον ασφαλισμένο ή στον συγγενή του.



### Ροές Δεδομένων (ΡΔ)

- ΡΔ-1:** Στοιχεία έξυπνης κάρτας (διαβίβαση)
- ΡΔ-2:** Υπάρχοντα στοιχεία ασθενούς (ανάκτηση)
- ΡΔ-3:** Στοιχεία φαρμάκων και περιορισμοί συνταγογράφησης (ανάκτηση)
- ΡΔ-4:** Νέα στοιχεία ασθενούς, διαγνώσεις και φάρμακα (καταγραφή)
- ΡΔ-5:** Ενημερωμένα στοιχεία έξυπνης κάρτας (ανάκτηση)
- ΡΔ-6:** Εγκεκριμένα φάρμακα εκκρεμούς συνταγής έξυπνης κάρτας (καταγραφή)
- ΡΔ-7:** Περιορισμοί συνταγογράφησης (ανάκτηση)
- ΡΔ-8:** Στοιχεία έξυπνης κάρτας (ανάκτηση)
- ΡΔ-9:** Υπάρχοντα στοιχεία ασθενούς (ανάκτηση)
- ΡΔ-10:** Στοιχεία εκτέλεσης συνταγής (καταγραφή)
- ΡΔ-11:** Υπάρχοντα στοιχεία ασθενούς και συνταγές (ανάκτηση)

- Σύνοψη προηγούμενης διάλεξης
- Ανάλυση και διαχείριση κινδύνων ανάπτυξης ΠΣ
- Κύκλος ζωής Πληροφοριακών Συστημάτων: Δραστηριότητες & Παραδοτέα
  - ▶ Ανάλυση / προσδιορισμός απαιτήσεων
  - ▶ Μελέτη εφικτότητας
  - ▶ Λογικός σχεδιασμός
  - ▶ Φυσικός σχεδιασμός
  - ▶ Λειτουργία / έλεγχοι & τρόποι μετάβασης
- Διαγράμματα ροής δεδομένων
- Η μέθοδος HIPO

- Η μέθοδος HIPO (*Hierarchical Input-Process-Output*), δηλαδή ιεραρχική εισροή - επεξεργασία- εκροή, μπορεί να χρησιμοποιηθεί ως διαγραμματικό εργαλείο αναπαράστασης μιας γενικής αλλά και λεπτομερούς θεώρησης ενός ΠΣ.
- Αρχικά, η μέθοδος HIPO αναπτύχθηκε από την εταιρία IBM για την τεκμηρίωση των επιμέρους λειτουργιών των εφαρμογών του ΠΣ έτσι ώστε να καθίσταται πιο εύκολος ο εντοπισμός των λειτουργιών που πρόκειται να τροποποιηθούν με άμεσο αποτέλεσμα την επιτάχυνση την διαδικασίας συντήρησης των εφαρμογών του ΠΣ.

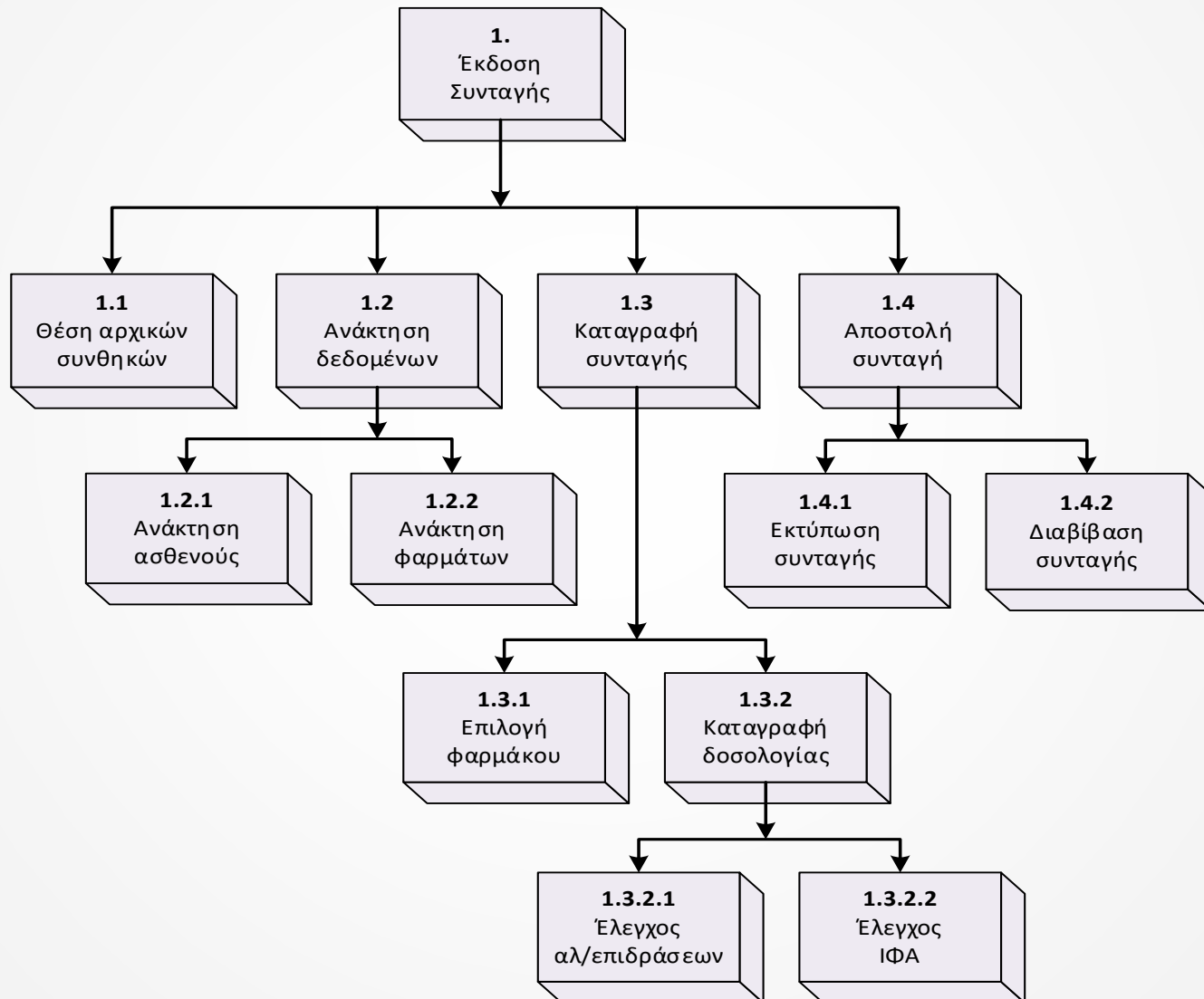
Οι κύριοι στόχοι της μεθόδου HIPO ως τεχνικής ανάπτυξης και τεκμηρίωσης συστημάτων είναι:

- Να παρέχει μια δομή για την καλύτερη κατανόηση των λειτουργιών που εκτελεί μια εφαρμογή του ΠΣ.
- Να καταγράφει τις επιμέρους λειτουργίες που εκτελούνται στο πλαίσιο μιας εφαρμογής του ΠΣ χωρίς να εμπλέκεται στον προσδιορισμό της δομής των προγραμμάτων εφαρμογών (της ροής εκτέλεσης των επιμέρους λειτουργιών).
- Να παρέχει μία οπτική περιγραφή των δεδομένων που χρησιμοποιούνται (εισροές) και των δεδομένων που παράγονται από κάθε λειτουργία (εκροές).

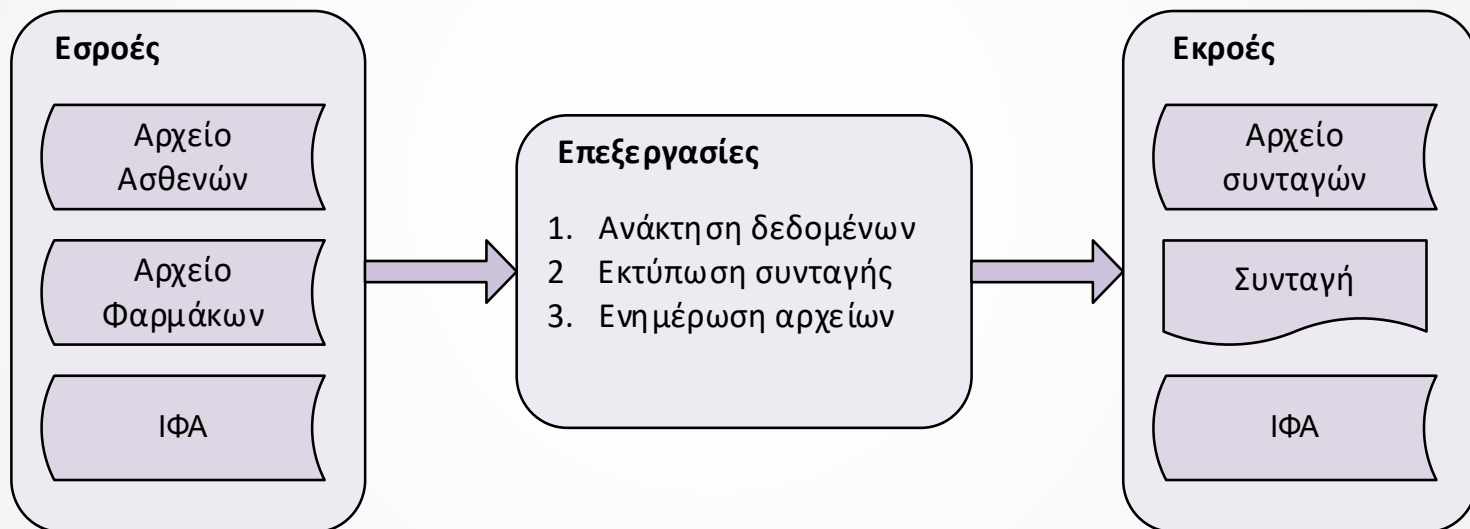
Έτσι, η μέθοδος HIPO παρέχει την δυνατότητα διαγραμματικής αναπαράστασης της διαδικασίας μετατροπής των εισροών σε εκροές. Για το σκοπό αυτό χρησιμοποιούνται τρία βασικά είδη διαγραμμάτων:

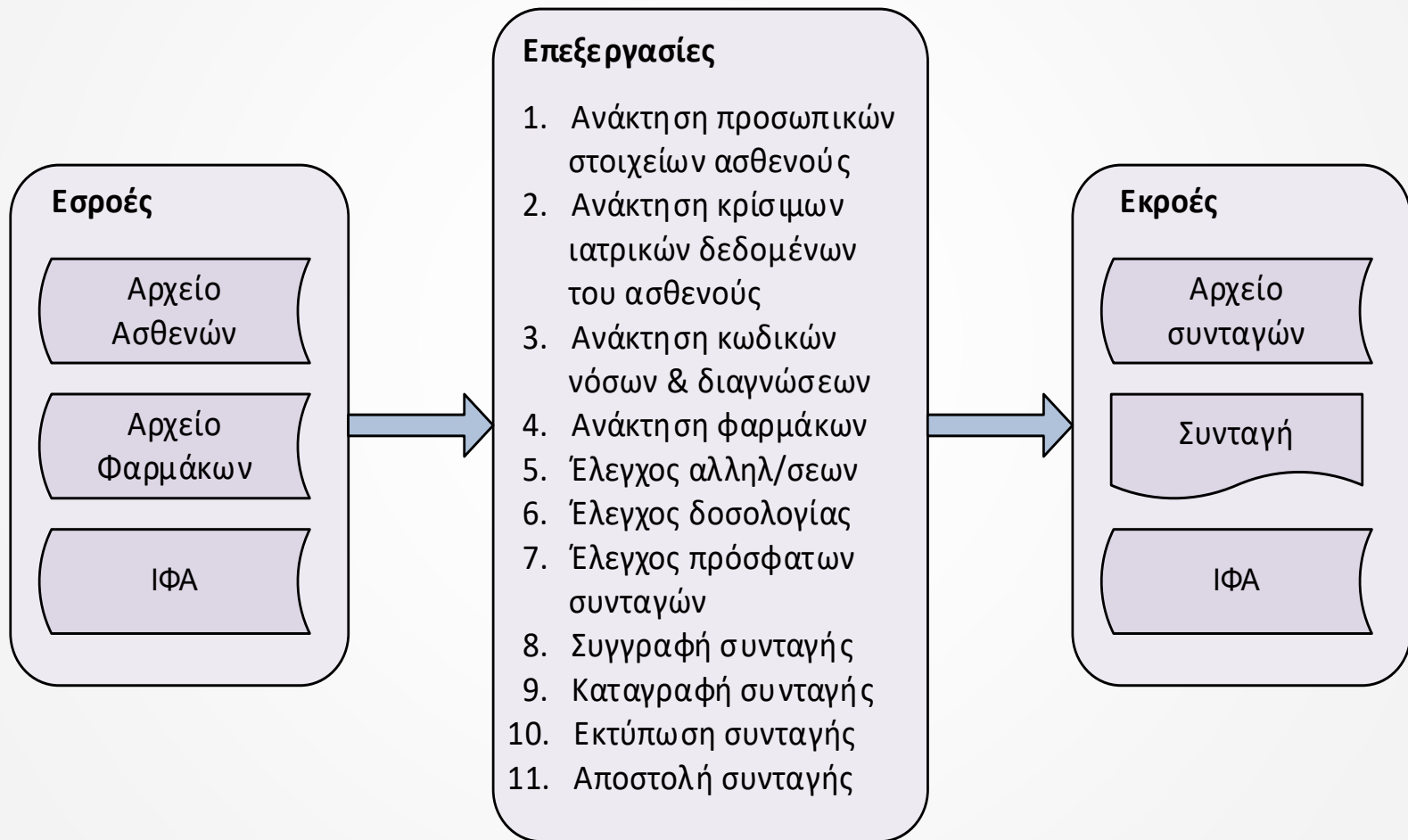
- Ο οπτικός πίνακας περιεχομένων (visual table of contents - VTOC).
- Τα γενικά διαγράμματα.
- Τα λεπτομερή διαγράμματα.

*Οι μορφές των παραπάνω διαγραμμάτων για ένα ΠΣ ηλεκτρονικής συνταγογράφησης φαίνονται στα ακόλουθα σχήματα*









- Ο οπτικός πίνακας περιεχομένων είναι ένα ιεραρχικό διάγραμμα στο οποίο περιγράφονται οι κύριες λειτουργίες (των εφαρμογών) του συστήματος και αποτελείται από αριθμημένα τετράγωνα κάθε ένα από τα οποία αναπαριστά ένα υποσύστημα.
- Τα γενικά και λεπτομερή διαγράμματα παρέχουν περισσότερες πληροφορίες για τα συστατικά των επιμέρους λειτουργιών των εφαρμογών του ΠΣ από αυτές που περιλαμβάνονται στον οπτικό πίνακα περιεχομένων και αποτελούνται από τρία τμήματα:
  - στο αριστερό τμήμα περιγράφονται οι εισροές δεδομένων στη λειτουργία,
  - στο μεσαίο τμήμα περιγράφεται η επεξεργασία των εισροών δεδομένων, και
  - στο δεξιό τμήμα περιγράφονται οι εκροές δεδομένων από τη λειτουργία.

Τα γενικά διαγράμματα διαφέρουν από τα λεπτομερή διαγράμματα στον βαθμό λεπτομέρειας για τις λειτουργίες που περιγράφουν.

- Ένα γενικό διάγραμμα περιγράφει τις εισροές δεδομένων, τις επεξεργασίες τους και τις εκροές δεδομένων μιας λειτουργίας ανωτέρου επιπέδου.
- Ένα λεπτομερές διάγραμμα παρέχει όλες τις απαραίτητες πληροφορίες για την πληρέστερη κατανόηση των λειτουργιών του αμέσως ανωτέρου επιπέδου του.

Η κατασκευή των τριών ειδών διαγραμμάτων της μεθόδου HIPO πραγματοποιείται ως εξής:

- i. Προσδιορίζονται οι λειτουργίες των εφαρμογών του συστήματος σε διάφορα επίπεδα λεπτομέρειας και τοποθετούνται ιεραρχικά σε έναν οπτικό πίνακα περιεχομένων.
- ii. Για κάθε λειτουργία ανωτέρου επιπέδου κατασκευάζεται ένα γενικό διάγραμμα και ένα λεπτομερές διάγραμμα αφού καταγραφούν οι εκροές στα αντίστοιχα δεξιά τμήματά τους
- iii. Προσδιορίζονται οι εισροές και οι επεξεργασίες που απαιτούνται για την παραγωγή τους.

Η διαδικασία αυτή επαναλαμβάνεται μέχρι την κατασκευή μιας λεπτομερούς περιγραφής του συστήματος.