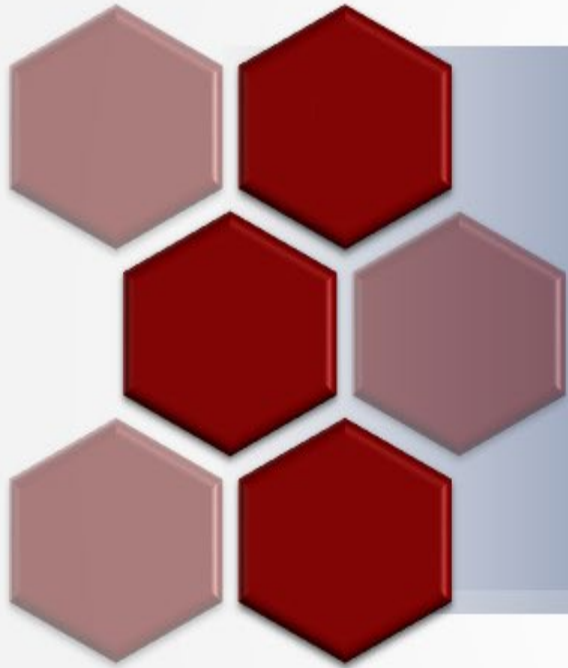


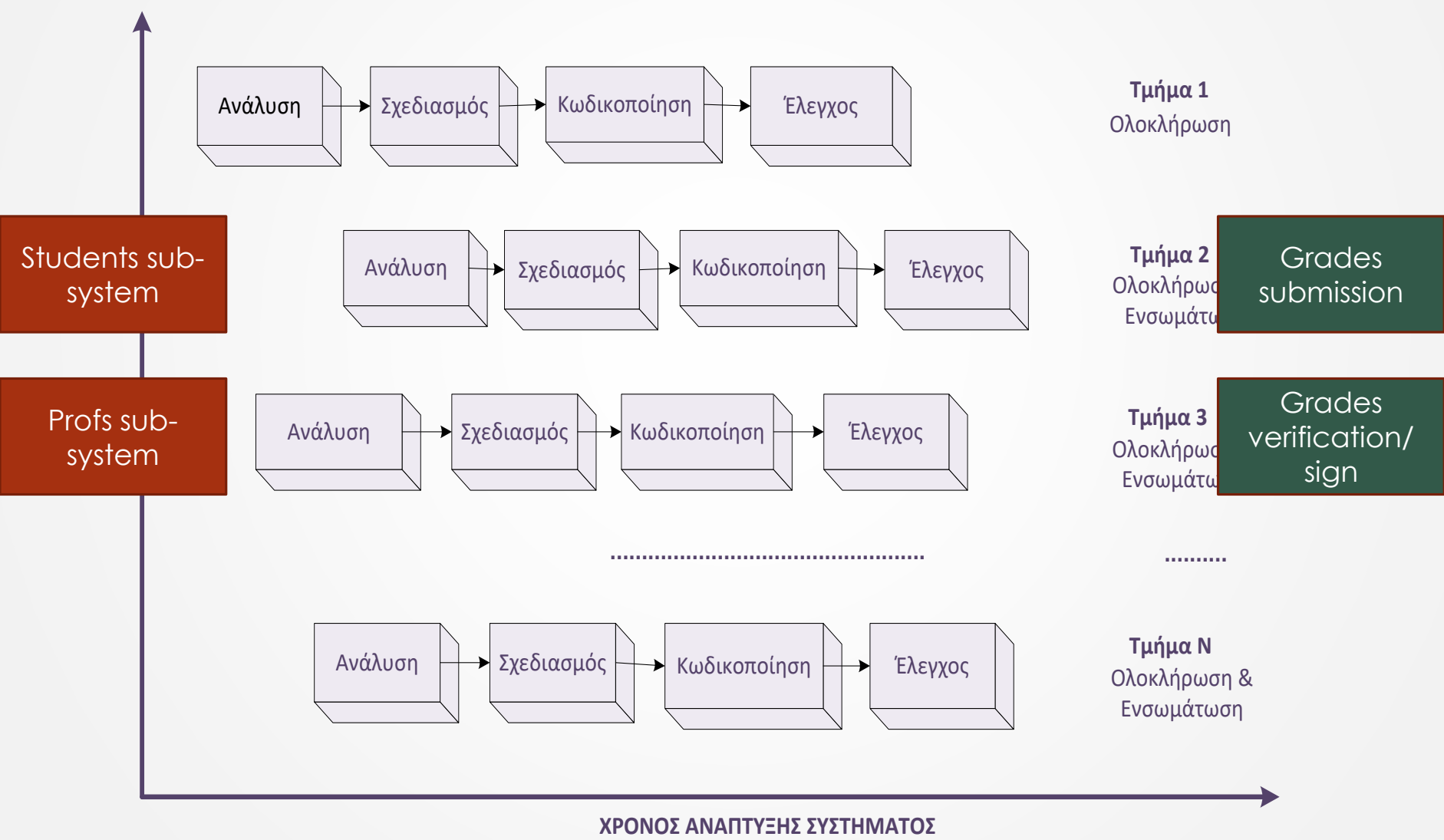


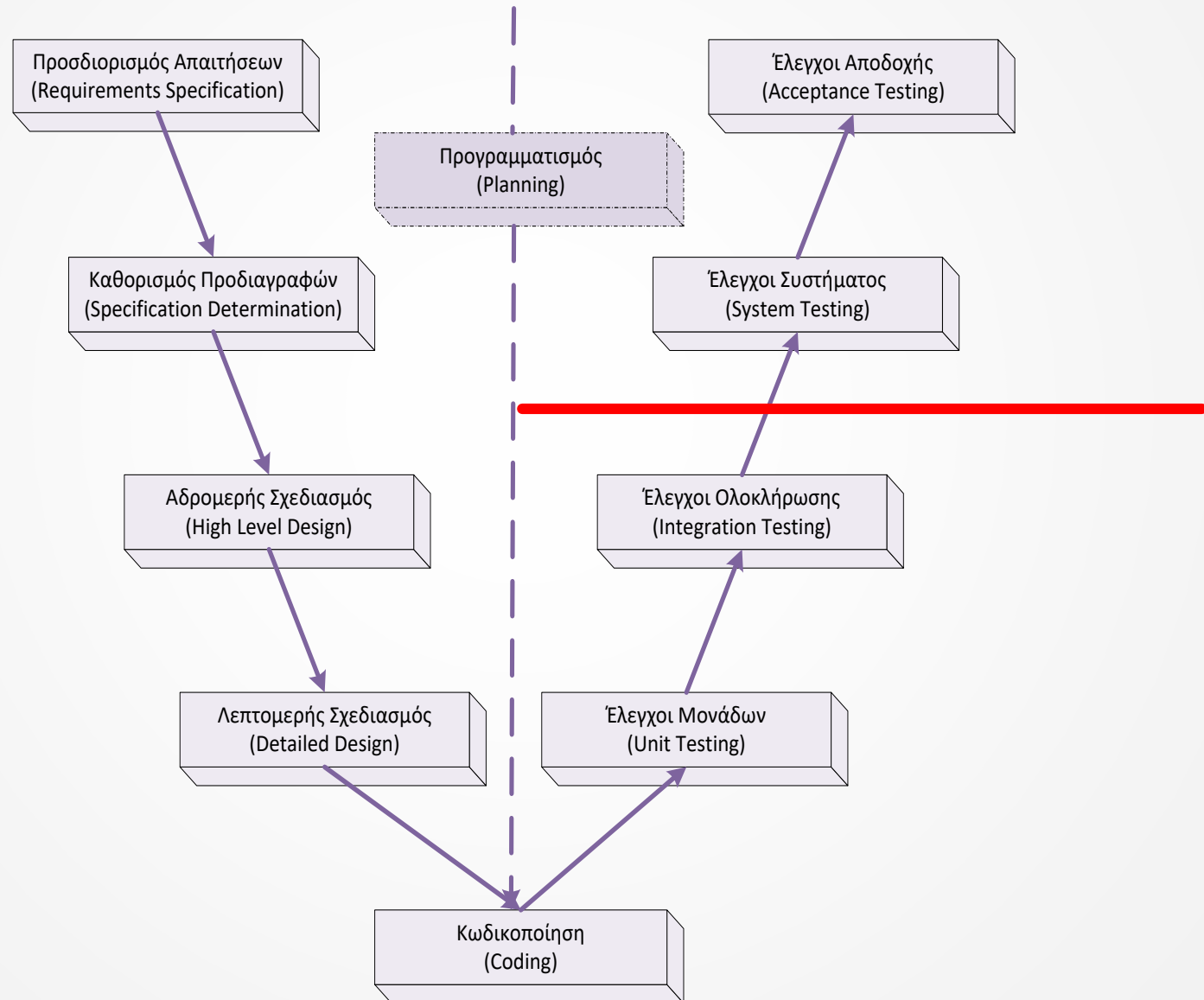
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

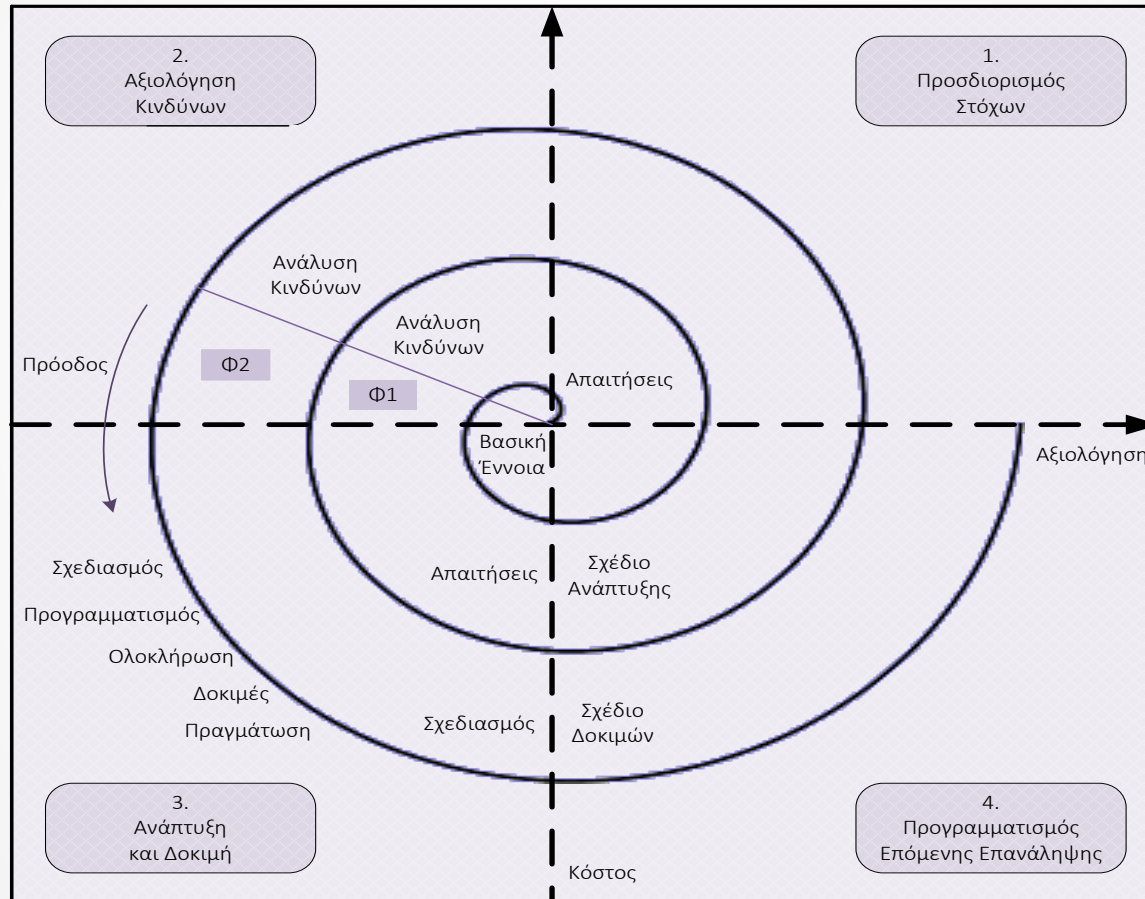


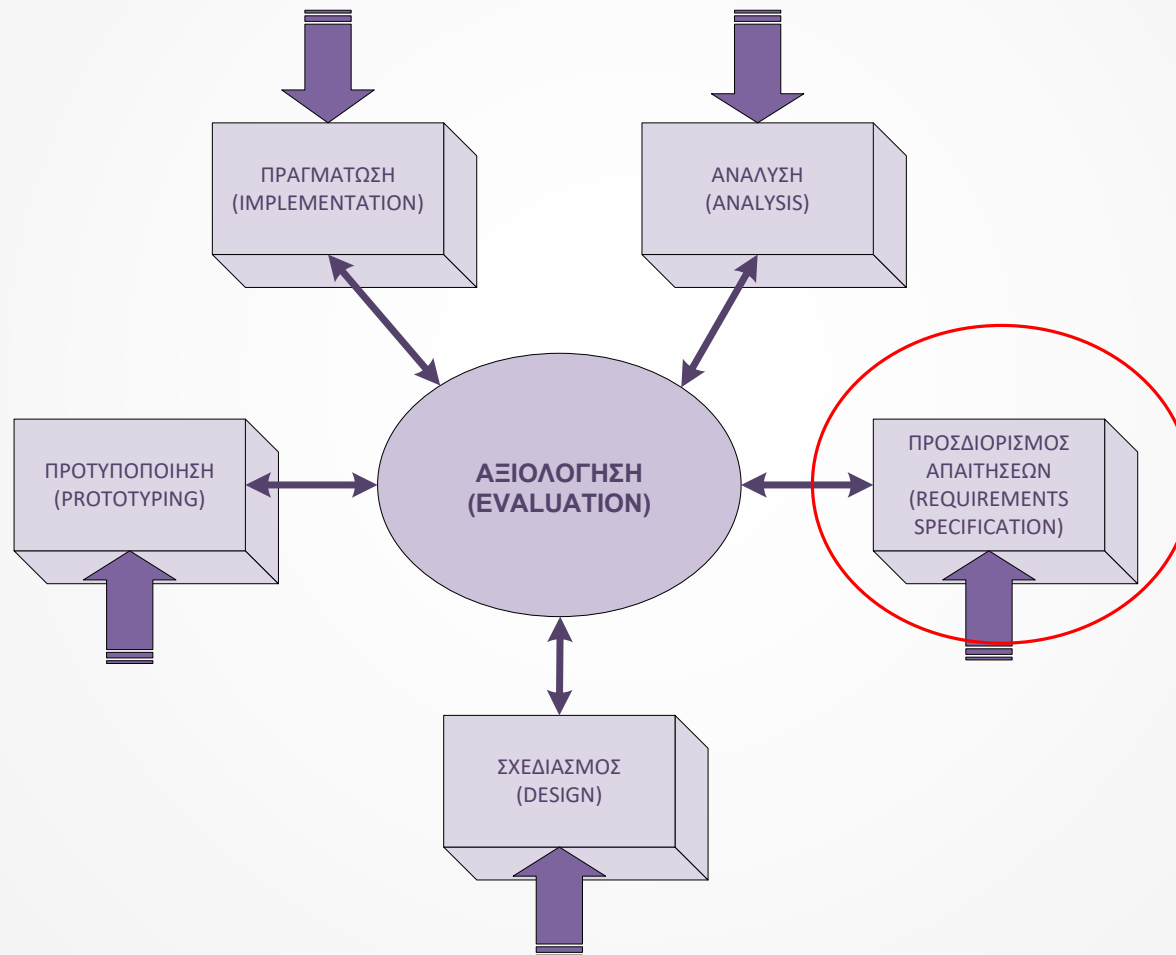
Πληροφορικά Συστήματα

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - ▶ Εισαγωγή
 - ▶ **Agile Software Development**
 - ▶ Η μεθοδολογία των κρυστάλλινων μεθόδων
 - ▶ Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - ▶ **Η μεθοδολογία Scrum**
 - ▶ Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

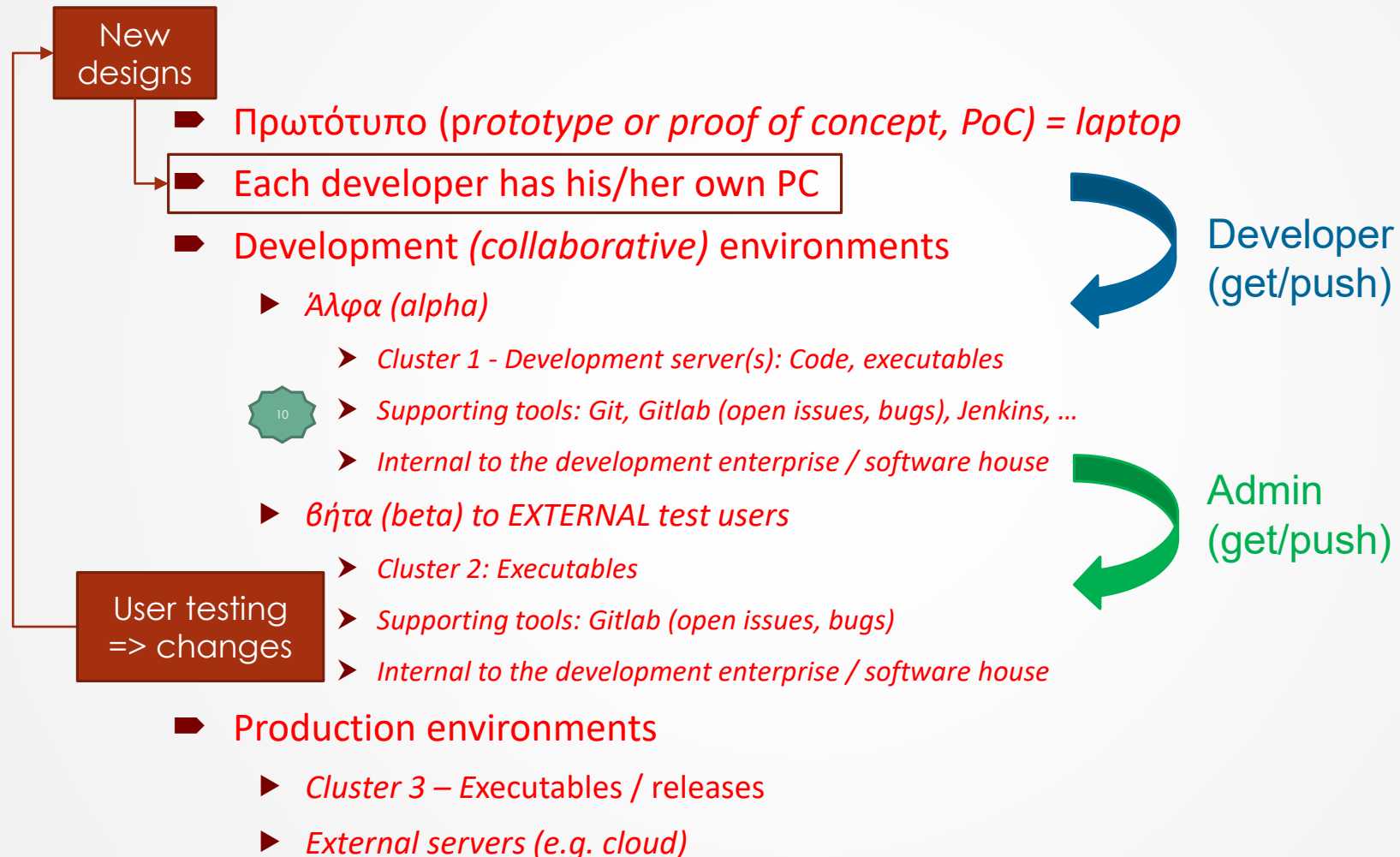


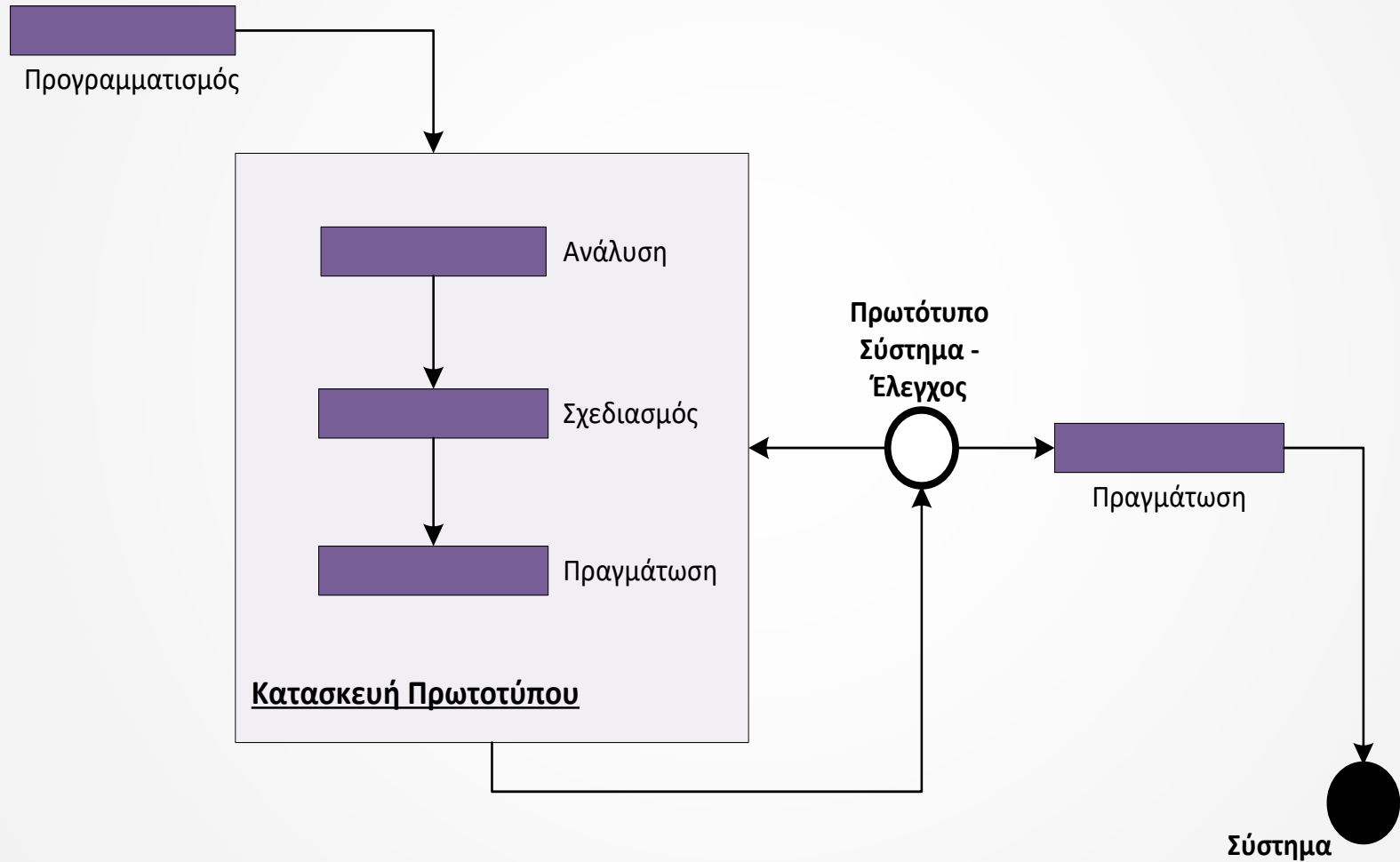


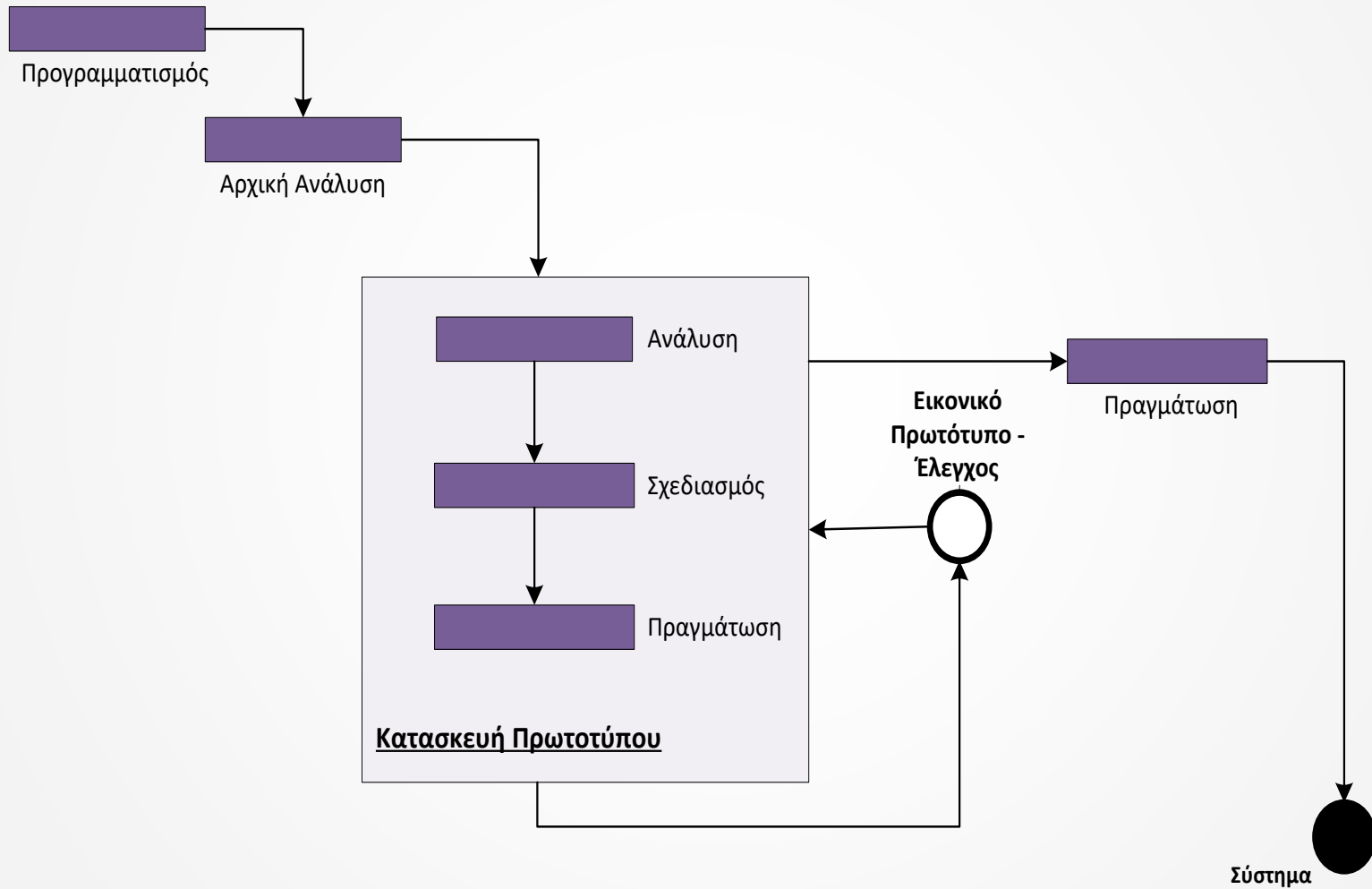


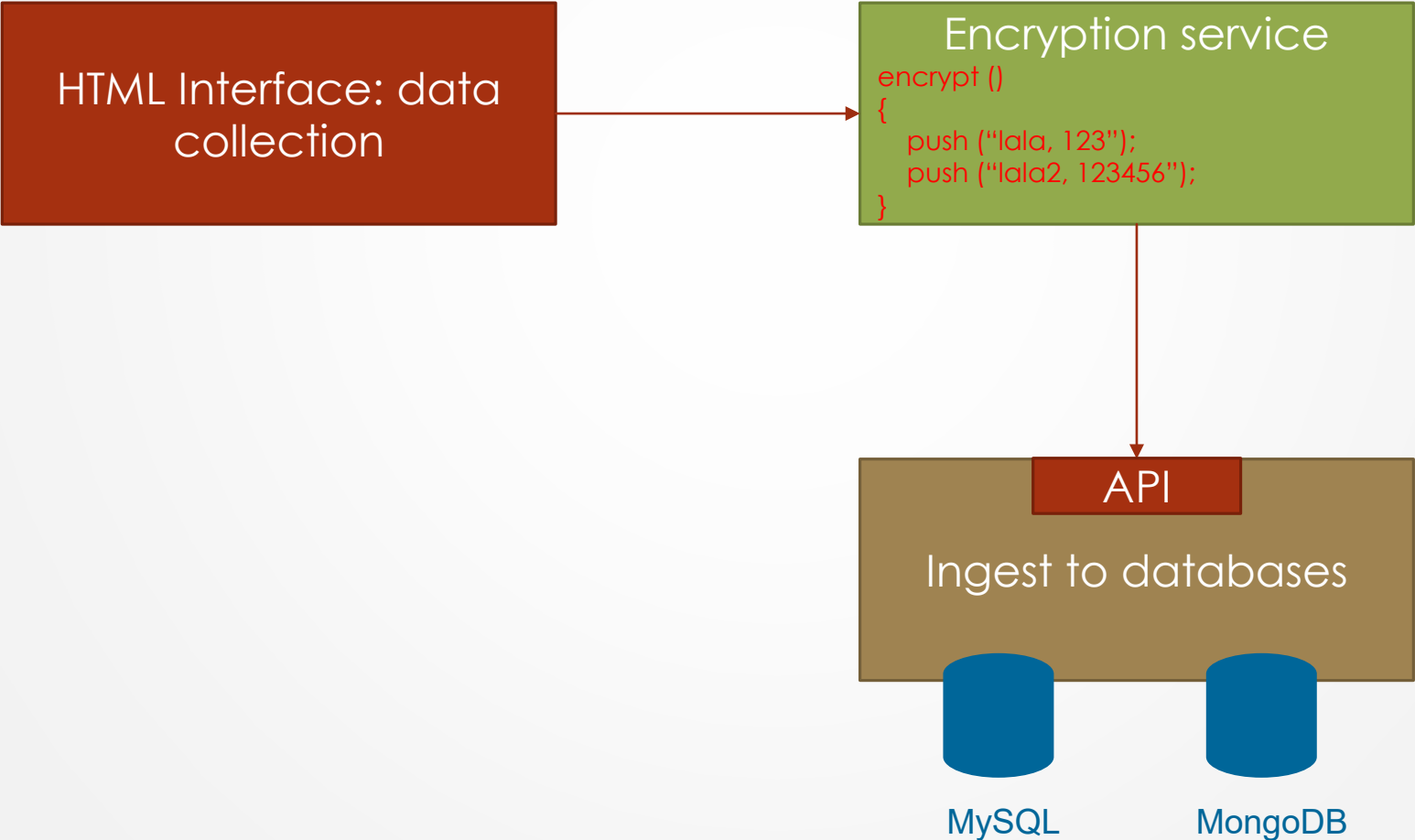


- Οι εφαρμογές ακολουθούν την πορεία:
 - ▶ πρωτότυπο(*prototype or proof of concept, PoC*)
 - ▶ άλφα(*alpha*) *not to end-users but used for testing & bug fixing!!!*
 - ▶ βήτα (*beta*) *to test users (closed group)*
 - ▶ έκδοση (*releaseSSSSSS*).









- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

- Η Ταχεία Ανάπτυξη Συστημάτων – ΤΑΣ (*Rapid Application Development – RAD*) συνιστά μια γενική προσέγγιση για την ανάπτυξη συστημάτων που επιστρατεύει εργαλεία, τεχνικές και μεθοδολογίες που έχουν σχεδιαστεί έτσι ώστε να επιταχύνεται η ανάπτυξη συστημάτων.
- Η διαδικασία της ανάπτυξης του συστήματος υπαγορεύει τη **γρήγορη ανάπτυξη κάποιου μέρους του συστήματος** (ίσως, του πιο σημαντικού) και την **παράδοσή του στους χρήστες προς σχολιασμό και υποβολή παρατηρήσεων** προκειμένου να κατασκευαστεί η επόμενη έκδοση του συστήματος που θα τις ικανοποιεί.

- Μερικά είδη (ή παραλλαγές) της ΤΑΣ προσπαθούν να προσαρμόζονται σε πιθανές τροποποιήσεις των επιχειρησιακών διεργασιών με εκτέλεση όλων των φάσεων της ανάπτυξης του συστήματος σχεδόν ταυτόχρονα.
- Τούτο συμβαίνει σε διάφορες μεθοδολογίες που αποσκοπούν στη γρήγορη ανάπτυξη συστημάτων προς αξιολόγηση από τους χρήστες, όπως:
 - η μεθοδολογία ταχείας ανάπτυξης πρωτοτύπων εφαρμογών (*Prototyping RAD*) και
 - Η μεθοδολογία ευέλικτης ανάπτυξης συστημάτων (*Agile Systems Development Methodology*).

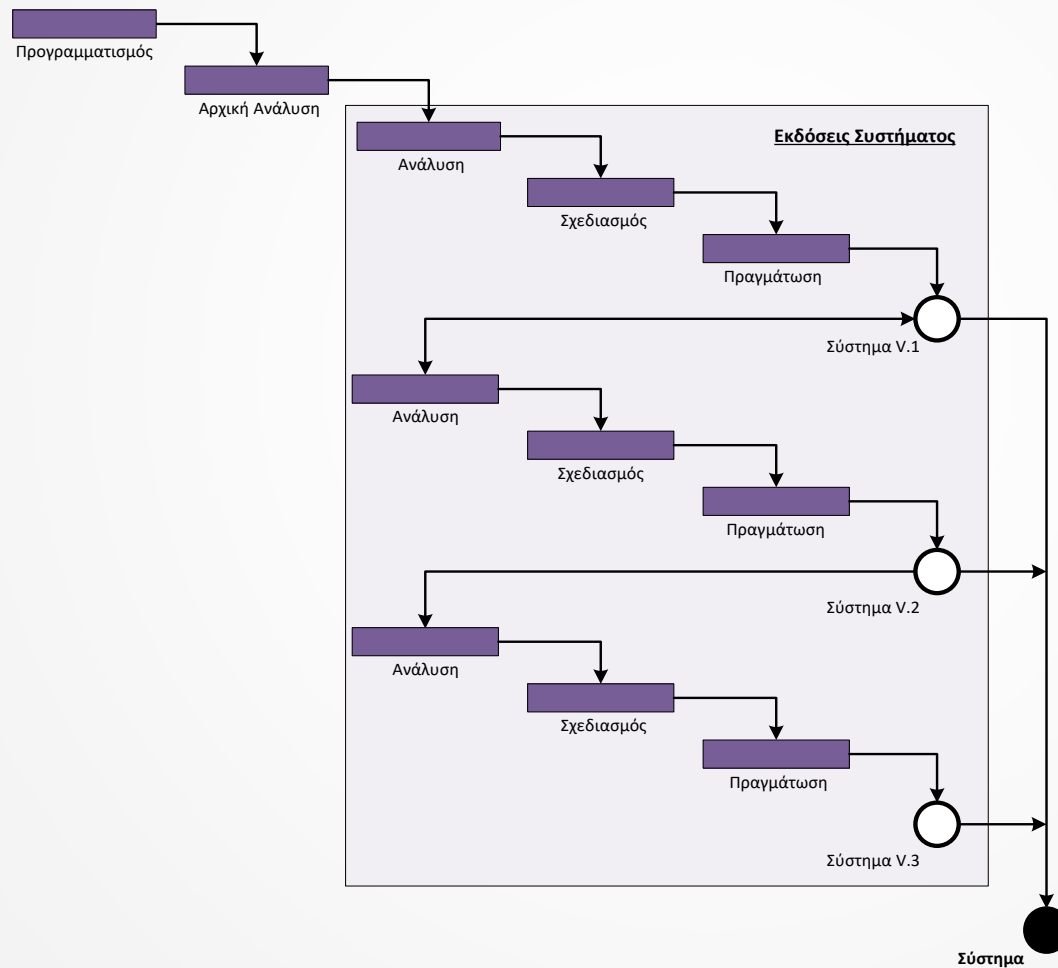
- Η γενική μεθοδολογία ΤΑΣ εισήγαγε τη χρήση προηγμένων εργαλείων ανάπτυξης συστημάτων, όπως:
 - οι **γεννήτριες κώδικα** (*low -> zero / no code programming*) και
 - Google AppMaker -> AppSheet, Microsoft PowerApps
 - οι **οπτικές γλώσσες προγραμματισμού**.
 - NodeRED
- Με τη χρήση αυτών των εργαλείων:
 - επιταχύνεται η διαδικασία ανάπτυξης του συστήματος και,
 - σε κάποιο βαθμό, παράγεται υψηλότερη ποιότητα κώδικα αναφορικά με τη δόμηση και την ικανοποίηση των απαιτήσεων.
- Η επιτάχυνση της παράδοσης του συστήματος, τουλάχιστον σε μία προχωρημένη πρωτότυπη μορφή, δίνει τη δυνατότητα στους χρήστες, να τροποποιούν και να επεκτείνουν τις αρχικές απαιτήσεις τους με αποτέλεσμα να βελτιώνονται τα χαρακτηριστικά του

Η γενική μεθοδολογία ΤΑΣ μπορεί να συμπεριλάβει πολλές κατηγορίες επιμέρους μεθοδολογιών. Στο πλαίσιο της γενικής μεθοδολογίας ΤΑΣ περιγράφονται τρεις επιμέρους μέθοδοι:

- Η Μέθοδος Σταδιακής Ανάπτυξης Συστημάτων (*Phased Development*),
- Η Μέθοδος Κοινής Ανάπτυξης Εφαρμογών (*Joint Application Development – JAD*), και
- Η Μέθοδος Ανάπτυξης Συστημάτων από Τελικούς Χρήστες (*End-User Development*).

- Κατά τη μέθοδο της σταδιακής ανάπτυξης συστημάτων (*phased development*), η ανάπτυξη του συστήματος πραγματοποιείται σε **διαδοχικές εμπλουτισμένες εκδοχές (*versions*)**, ή **αποδεσμεύσεις (*releases*)**, του με τις πιο βασικές και σημαντικές λειτουργίες να ενσωματώνονται στην πρώτη εκδοχή.
- Η ανάπτυξη της **επόμενης εκδοχής του συστήματος αρχίζει μόνον όταν η προηγούμενη εκδοχή έχει εγκριθεί** ως προς το σχεδιασμό της, έχει πραγματωθεί και επιδοκιμασθεί ως προς την κατασκευή της.

- Κάθε εκδοχή περιλαμβάνει τις δικές της βασικές φάσεις ενός κύκλου ζωής που ουσιαστικά συμπυκνώνονται στις φάσεις **Ανάλυση – Σχεδιασμός – Πραγμάτωση – Έλεγχος** (σε μικρότερη κλίμακα απ' ότι για το όλο σύστημα).
- Έτσι, κατασκευάζεται πολύ σύντομα, με τεχνικές και εργαλεία ταχείας ανάπτυξης συστημάτων, και παραδίδεται στους χρήστες ένα “χρήσιμο” για ουσιαστικό σχολιασμό σύστημα αν και, εν γνώσει των κατασκευαστών, δεν καλύπτει ακόμη όλες τις προβλεπόμενες, και προδιαγραφείσες, λειτουργίες ή επιχειρησιακές διεργασίες του οργανισμού ούτε όλες τις λειτουργικές και τεχνικές απαιτήσεις του συστήματος.



- Η μέθοδος της σταδιακής ανάπτυξης συστημάτων διαφέρει από τις μεθόδους της επαυξητικής ανάπτυξης στο ότι θέτει στο επίκεντρο την γρήγορη ανάπτυξη του συστήματος και, γι' αυτόν το λόγο, χρησιμοποιεί τις αρχές της ταχείας ανάπτυξης συστημάτων (αναφορικά με τις τεχνικές και τα εργαλεία) προκειμένου να παράξει την αρχική και ενδεχομένως τις επόμενες εκδοχές του συστήματος.
- Έτσι, επιταχύνεται η ανάπτυξη του συστήματος αποφέροντας σημαντικό όφελος στον οργανισμό.

- Η μεθοδολογία κοινής ανάπτυξης εφαρμογών – ΚΑΕ (Joint Application Development – JAD):
 - ανήκει στην κατηγορία των συνεργατικών μεθοδολογιών ανάπτυξης συστημάτων
 - δημιουργήθηκε προκειμένου να επισπευσθεί το στάδιο του σχεδιασμού του συστήματος.
 - Shorten overall time λόγω σωστού fragmentation
- Ο σχεδιασμός του βασικού κορμού του συστήματος, αναφορικά με τις εισροές (δεδομένα και οθόνες) και τις εκροές (αναφορές) του, πραγματοποιείται κατά τη διάρκεια εντατικών συναντήσεων εργασίας μεταξύ χρηστών, διευθυντικών στελεχών και τεχνικών πληροφορικής.
- Η συμμετοχή όλων των ενδιαφερομένων μερών σε συναντήσεις έχει ως αποτέλεσμα την ταχύτερη σύνθεση διαφορετικών

Βασικοί λόγοι που ερμηνεύουν τη χρήση της μεθοδολογίας:

1. Η συσσώρευση μεγάλου αριθμού αιτημάτων για υποψήφια έργα ανάπτυξης ΠΣ από διάφορες επιχειρησιακές μονάδες προς την μονάδα πληροφορικής του οργανισμού με αποτέλεσμα την σημαντική καθυστέρηση στην αξιολόγηση των αιτημάτων και την, κατ' ακολουθία, καθυστέρηση έναρξης της ανάπτυξης των αντιστοίχων συστημάτων (εφαρμογών).
2. Οι αυξανόμενες δυνατότητες των σύγχρονων εργαλείων λογισμικού από άποψης χρησιμότητας και χρηστικότητας. Έτσι, καθίσταται δυνατή η ανάπτυξη μικρών και χρήσιμων εφαρμογών από τους χρήστες εντός μερικών μόλις ημερών, αλλά και η ανάπτυξη πιο σύνθετων εφαρμογών αξιοποιώντας τις δυνατότητες ισχυρότερων και πληρέστερων εργαλείων λογισμικού.

Κύρια πλεονεκτήματα:

- Το γεγονός ότι οι χρήστες παίρνουν αυτό που θέλουν πολύ πιο σύντομα, κατά παρόμοιο τρόπο με άλλες μεθόδους που αποσκοπούν στη γρήγορη ανάπτυξη του συστήματος.
- Το γεγονός ότι οι χρήστες παίρνουν αυτό που θέλουν χωρίς τη συμβολή τεχνικών πληροφορικής, οπότε δεν απαιτείται η περιγραφή της λειτουργίας του οργανισμού σ' αυτούς που μπορεί να μην είναι επιθυμητό.

Κύρια μειονεκτήματα:

- Απαιτείται σημαντική εκπαίδευση των τελικών χρηστών στην ανάπτυξη συστημάτων (εφαρμογών) μέσω των συγκεκριμένων εργαλείων που πρόκειται να χρησιμοποιηθούν. Έτσι:
 - Συχνά, οι εφαρμογές κατασκευάζονται από έναν χρήστη για την ικανοποίηση των προσωπικών του πληροφοριακών αναγκών στο πλαίσιο της εργασίας του (οι οποίες μπορεί να διαφέρουν από χρήστη σε χρήστη). Αυτό σημαίνει ότι τα νέα συστήματα δεν είναι προσαρμοσμένα στους ρόλους αλλά στα πρόσωπα κάτι το οποίο δυσχεραίνει τη λειτουργία τους στο πλαίσιο του οργανισμού ανεξάρτητα του προσώπου που καταλαμβάνει τη συγκεκριμένη θέση.

- Συχνά, οι χρήστες που έχουν αναπτύξει τις εφαρμογές δεν καταγράφουν την τεκμηρίωσή τους οπότε δυσκολεύεται η χρήση των εφαρμογών από άλλους χρήστες. Επίσης δυσχεραίνεται η συντήρηση και η επέκταση των εφαρμογών για την κάλυψη νέων αναγκών με το πέρασμα του χρόνου και την πρόοδο της τεχνολογίας.
- Συχνά, οι χρήστες δυσκολεύονται να ελέγξουν την ορθότητα και πληρότητα των εφαρμογών τους λόγω της ελλιπούς εκπαίδευσης που έχουν λάβει. Ιδιαίτερες δυσκολίες μπορεί να εμφανίζονται σε θέματα ασφάλειας αναφορικά με την αποθήκευση και τη διακίνηση των πληροφοριών αλλά και με την απόκρουση κακόβουλων επιθέσεων κατά του συστήματος.

- Στα παραπάνω μειονεκτήματα πρέπει να προστεθεί και το βασικό μειονέκτημα των ενδεχόμενων αποσπασματικών βάσεων δεδομένων που έχουν κατασκευαστεί χωρίς ενιαίο σχεδιασμό και/ή χωρίς ενιαίους κανόνες με αποτέλεσμα να δυσχεραίνεται η διαλειτουργικότητα των δεδομένων σε σημασιολογικό και σε τεχνικό επίπεδο. Το γεγονός αυτό είναι σημαντικά επιβλαβές για την διαδικασία λήψης αποφάσεων στον οργανισμό, ιδιαίτερα στη σύγχρονη εποχή κατά την οποία η ύπαρξη ομογενοποιημένων (κατ' επιθυμία) και ανοικτών δεδομένων έχει καταστεί σημαντικός παράγων στη διαδικασία λήψης εδραιωμένων αποφάσεων με βάση σύγχρονα μαθηματικά, κυρίως, μοντέλα.

► Design phase: datastore(s) DECISION – ερώτημα για sample datasets

► Choices (2 baseis dedomenon + 1 datastore):

1. Transactions (OLTP): SQL
2. Analytics (OLAP): NoSQL
3. Archive: Data lakes (object store: CEPH)

- ETL: Extract (1) Transform Load (2)

► Choices (basei arithmou xriston -> arithmo

nodes -> SQL / NoSQL):

- MySQL (SQL database)
- MongoDB (NoSQL database)

- name: Nikos
- tel: 21041..
- add: [
- home: gr. lampraki
- office: kissamou

- 1

UserID	Name	Tel
1	Nikos	21041...
4	Maria	21098...



UserID	Home Address	Office Address
1	Gr. Lampraki	Kissamou ...

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

- Οι μεθοδολογίες ευέλικτης ανάπτυξης συστημάτων – ΕΑΣ (**Agile Systems Development** – ASD) αναπτύχθηκαν περί τα μέσα της δεκαετίας του '90 ως αντίδραση ενάντια στην παραδοσιακή καταρρακτοειδή μεθοδολογία των ΚΖΑΣ η οποία εθεωρείτο ανελαστική, αργή και ασυνεπής αναφορικά με τον τρόπο λειτουργίας των κατασκευαστών λογισμικού στην πράξη.

Με βάση τη βιβλιογραφία κάποιοι ορισμοί για την ευέλικτη ανάπτυξη συστημάτων είναι οι ακόλουθοι:

- ➤ Ευέλικτη ονομάζεται μια μεθοδολογία ανάπτυξης συστημάτων λογισμικού η οποία είναι:
 - Ανθρωποκεντρική και επικοινωνιοστρεφής (εστιάζει στην επικοινωνία μεταξύ των εμπλεκόμενων στην ανάπτυξη του συστήματος ανθρώπων (π.χ. κατασκευαστών και χρηστών)
 - Ευέλικτη (είναι έτοιμη για προσαρμογή σε προβλέψιμες αλλαγές του συστήματος σε οποιονδήποτε χρόνο κατά τη διάρκεια της ανάπτυξης ή λειτουργίας του),
 - Ταχεία (ενθαρρύνει τη γρήγορη και επαναληπτική ανάπτυξη του συστήματος σε μικρές εκδόσεις),

- Εύλογη (εστιάζει στη μείωση του χρόνου και του κόστους κατασκευής του συστήματος με παράλληλη βελτίωση της ποιότητάς του),
- Αποκριτική (ανταποκρίνεται κατάλληλα σε αναμενόμενες και μη αναμενόμενες αλλαγές του συστήματος), και
- Μαθησιακή (εστιάζει στη διαρκή βελτίωση του συστήματος, και κατά τη διαδικασία ανάπτυξής του και κατόπιν αυτής).
 - **FEEDBACK LOOP**

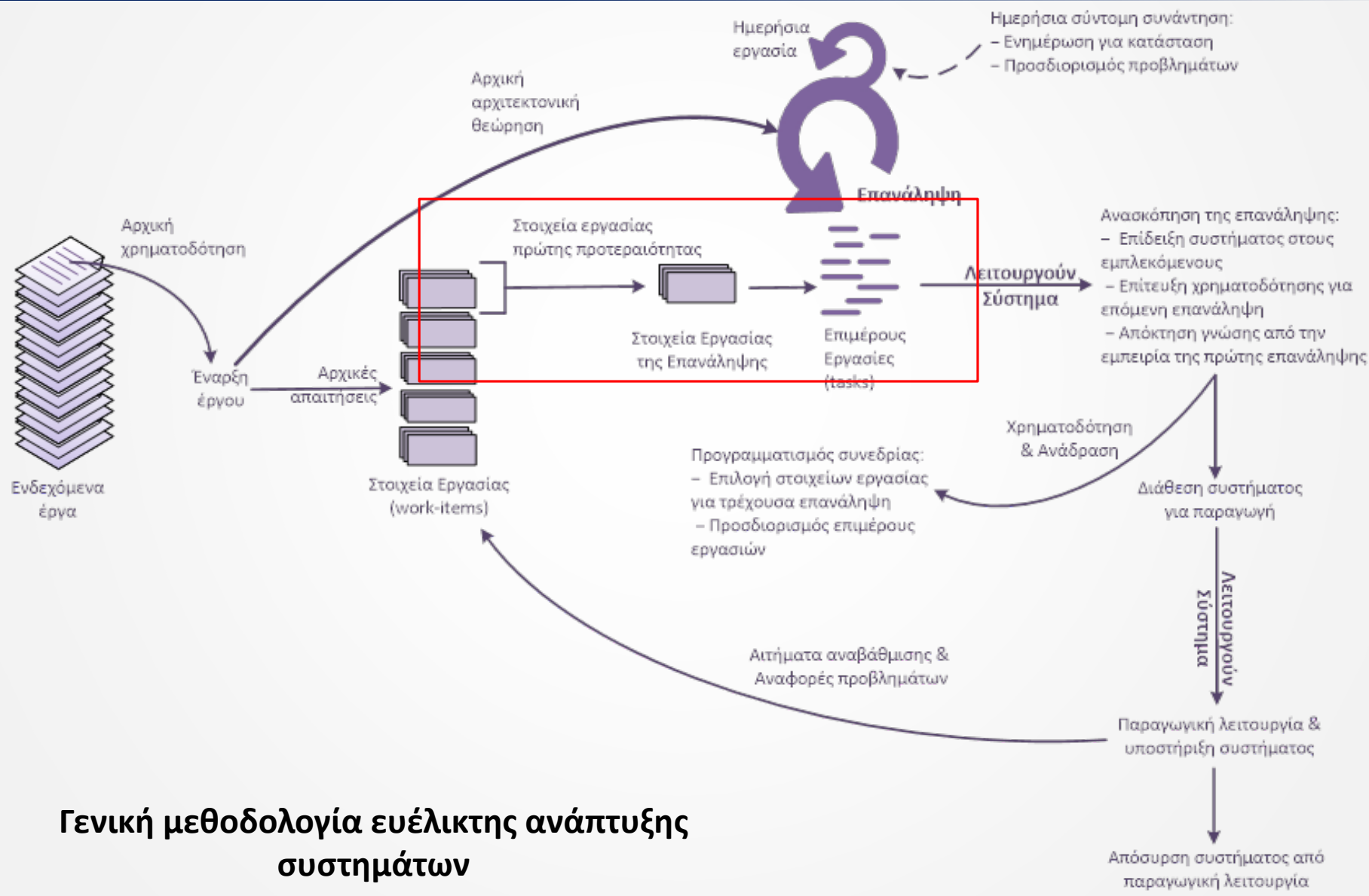
- Οι ευέλικτες μεθοδολογίες αποτελούν μια απόκριση στις μεγάλες και γρήγορες αλλαγές που συμβαίνουν στα σύγχρονα επιχειρηματικά και πληροφοριακά περιβάλλοντα.
- Οι ευέλικτες μεθοδολογίες προωθούν μια πειθαρχημένη διεργασία διοίκησης έργων ανάπτυξης συστημάτων η οποία ενθαρρύνει:
 - Τη διαρκή επιθεώρηση και προσαρμογή,
 - Τον προσανατολισμό στην ομαδική εργασία,
 - Την αυτο-οργάνωση και τη λογοδοσία,
 - Την υιοθέτηση ενός συνόλου καλών πρακτικών μηχανευτικής συστημάτων που επιτρέπουν τη γρήγορη ανάπτυξη λογισμικού υψηλής ποιότητας, και
 - Την εναρμόνιση του υπό ανάπτυξη του συστήματος με τις ανάγκες των χρηστών και τους στόχους και τη στρατηγική του οργανισμού.

- Υπάρχουν διάφορες επιμέρους μεθοδολογίες ευέλικτης ανάπτυξης συστημάτων, όπως:
 - Οι κρυστάλλινες μέθοδοι (Crystal Methods),
 - Το δυναμικό μοντέλο ανάπτυξης συστημάτων (Dynamic Systems Development Model - DSDM),
 - Η μεθοδολογία Scrum και
 - Η, πιο ριζοσπαστική όλων, μεθοδολογία του ακραίου προγραμματισμού (Extreme Programming – XP).

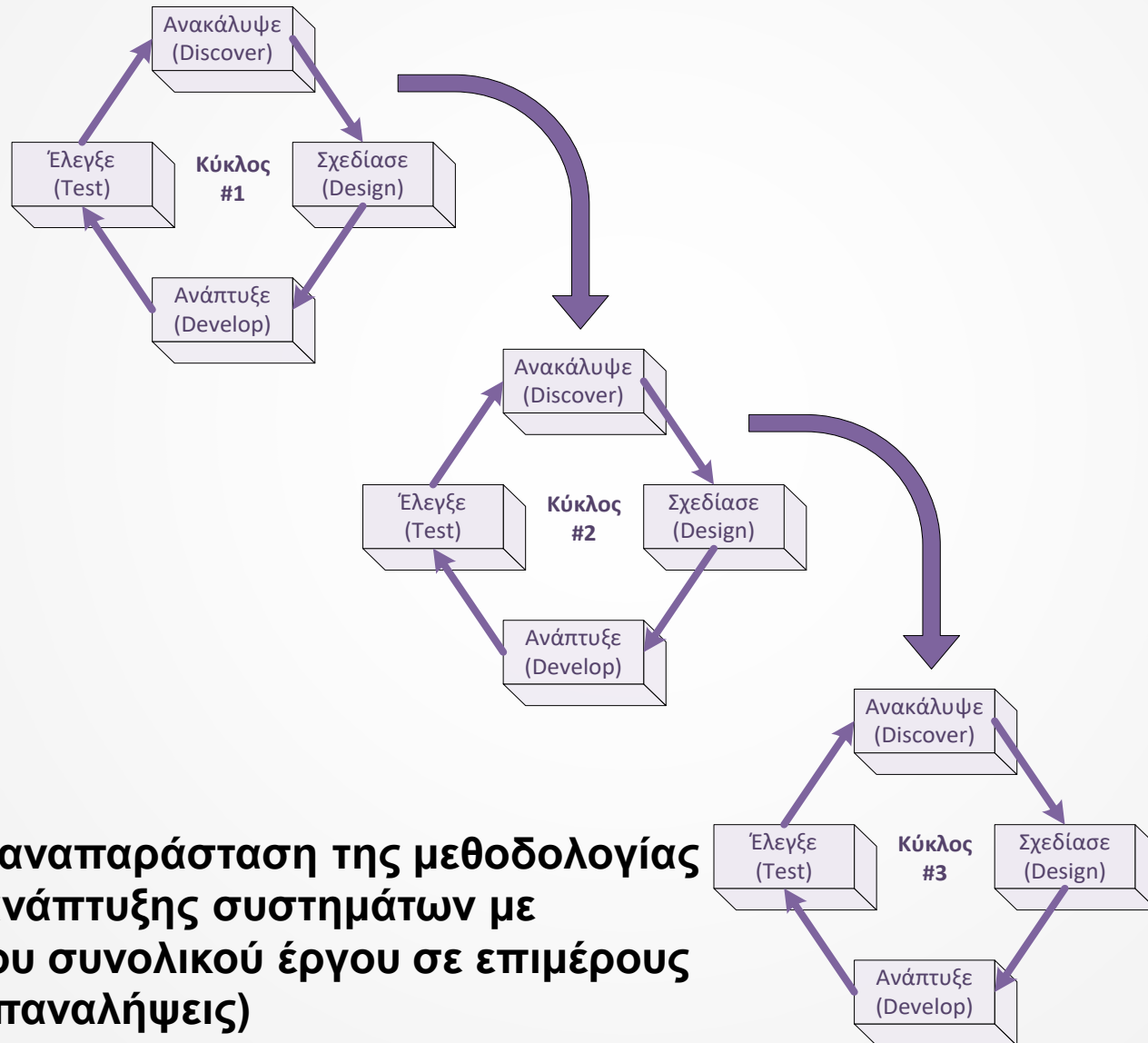
Οι περισσότερες μεθοδολογίες ευέλικτης ανάπτυξης συστημάτων αποσκοπούν στην ελαχιστοποίηση του κινδύνου αποτυχίας αναφορικά με την ανάπτυξη και αποδοχή του λογισμικού.

- Αναπτύσσουν το λογισμικό σε μικρά χρονικά διαστήματα (επαναλήψεις), διάρκειας από μία έως 4 εβδομάδες.
- Κάθε επανάληψη προσομοιάζει με ένα μικρό έργο ανάπτυξης συστήματος και περιλαμβάνει τις ακόλουθες εργασίες που είναι αναγκαίες για την κατασκευή μιας ελάχιστης επαύξησης της λειτουργίας και λειτουργικότητας του συστήματος.
 - του προγραμματισμού (*planning*),
 - της ανάλυσης απαιτήσεων (*requirements analysis*),
 - του σχεδιασμού (*design*),
 - της κωδικοποίησης (*coding*),
 - του ελέγχου ή των δοκιμών (*testing*), και
 - της τεκμηρίωσης (*documentation*).

- Ενώ μια επανάληψη, γενικώς, μπορεί να μην προσθέτει αρκετές νέες λειτουργίες και αρκετή νέα λειτουργικότητα ώστε να δικαιολογείται η αποδέσμευση ή νέα έκδοση του συστήματος (προϊόντος), η ευέλικτη ανάπτυξη συστημάτων σκοπεύει στη διάθεση νέου, βελτιωμένου αναφορικά με τα λειτουργικά χαρακτηριστικά του, λογισμικού στο τέλος κάθε επανάληψης.
- Κατόπιν, το λογισμικό αξιολογείται αναφορικά με τις προσδιορισθείσες και τις πιθανές νέες απαιτήσεις από μια ομάδα ανάπτυξης του έργου που ιδεατά αποτελείται από κατασκευαστές, ειδικούς του επιχειρησιακού περιβάλλοντος και χρήστες.
- Πρέπει να σημειωθεί εδώ ότι η μεθοδολογία ευέλικτης ανάπτυξης δίδει μεγάλη έμφαση στην συμμετοχή των χρηστών και του πελάτη στη διεργασία ανάπτυξης του συστήματος καθότι πολλά λειτουργικά χαρακτηριστικά του συστήματος δεν είναι γνωστά κατά την έναρξη του έργου αλλά προκύπτουν από τους χρήστες κατά τη διεργασία των επαναλήψεων.



Γενική μεθοδολογία ευέλικτης ανάπτυξης συστημάτων



Συνοπτική αναπαράσταση της μεθοδολογίας ευέλικτης ανάπτυξης συστημάτων με διάκριση του συνολικού έργου σε επιμέρους κύκλους (επαναλήψεις)

- Οι ευέλικτες μέθοδοι δίδουν έμφαση στην σε πραγματικό χρόνο διαπροσωπική επικοινωνία μεταξύ των μελών της ομάδας εργασίας επί του τεκμηριωτικού υλικού του συστήματος (εγγράφου και λογισμικού) το οποίο υπέχει θέση τεκμηρίωσης εργασίας.
- Κατά την επικοινωνία, πρωταρχικό μέτρο της προόδου του έργου θεωρείται το παραχθέν λογισμικό καθότι οι ευέλικτες μέθοδοι παράγουν πολύ λιγότερη έγγραφη τεκμηρίωση σε σχέση με άλλες παραδοσιακές μεθόδους.
- Σε πολλές περιπτώσεις, η ομάδα ανάπτυξης του συστήματος εγκαθίσταται σε έναν συγκεκριμένο χώρο καθόλη τη διάρκεια του έργου. Αποτελείται δε από εκπροσώπους όλων των εμπλεκόμενων ρόλων στην ανάπτυξη του συστήματος.
- Επίσης, μπορεί να συμμετέχουν τα άτομα που ορίζουν τα επιθυμητά συστήματα προς ανάπτυξη όπως είναι οι διοικητές συστημάτων, οι επιχειρησιακοί αναλυτές ή οι πραγματικοί πελάτες.

Το μανιφέστο ευελιξίας περιγράφει τις ακόλουθες τέσσερις σημαντικές αξίες που σήμερα έχουν τόσο νόημα όσο είχαν τότε που περιγράφησαν:

- Έμφαση στα άτομα και τις αλληλεπιδράσεις τους αντί στις διαδικασίες και τα εργαλεία ανάπτυξη συστημάτων.
- Έμφαση στην **ανάπτυξη λογισμικού εργασίας** αντί στην ανάπτυξη εμπεριστατωμένου τεκμηριωτικού υλικού.
- Έμφαση στη εγκαθίδρυση κλίματος συνεργασίας με τους πελάτες αντί στις στιβαρές διαπραγματεύσεις συμβολαίων με τους πελάτες.
- Έμφαση στην **αντιμετώπιση της αντίστασης στην αλλαγή (resistance to change)** αντί στον **αυστηρό προγραμματισμό (planning)** των ενεργειών.

- Ενώ αναγνωρίζεται ότι ενυπάρχει αξία στα στοιχεία που βρίσκονται στο δεξιό μέρος των άνω προτάσεων (μετά τη λέξη “αντί”), δίδεται περισσότερη βαρύτητα, ή μεγαλύτερη έμφαση, στην αξία που ενυπάρχει στα στοιχεία που βρίσκονται στο αριστερό μέρος των προτάσεων (πριν τη λέξη “αντί”).
- Από τότε που περιγράφηκε το μανιφέστο, η χρήση ευέλικτων μεθόδων παρουσιάζει διεθνώς αυξητική τάση δημοφιλίας.
- Συνεπώς, με βάση τις παραπάνω τέσσερις αξίες του μανιφέστου ευελιξίας μπορούν να καταγραφούν διάφορα σύνολα αρχών ευέλικτης ανάπτυξης συστημάτων στο πλαίσιο των τεσσάρων αυτών αξιών.

- Τα σύνολα αρχών προσδιορίζονται αυθαίρετα, στο πλαίσιο των βασικών επιταγών της ευέλικτης ανάπτυξης, ανάλογα με τις εμπειρίες και τις ικανότητες (δεξιότητες) αυτού που τα προτείνει.
- Μια προσέγγιση επέκτασης των τεσσάρων αξιών της ευέλικτης ανάπτυξης σε ένα σύνολο αρχών για εφαρμογή στην ανάπτυξη συστημάτων είναι η ακόλουθη:
 - Η ενεργή εμπλοκή των χρηστών στην διαδικασία ανάπτυξης του συστήματος είναι επιβεβλημένη.
 - Η ομάδα ανάπτυξης του συστήματος πρέπει να διαθέτει την δικαιοδοσία να λαμβάνει αποφάσεις.
 - Οι απαιτήσεις εξελίσσονται και διαφοροποιούνται, πιθανώς, αλλά το χρονοδιάγραμμα παραμένει σταθερό.
 - 1. Planned (change) requests
 - 2. Add team members
 - Οι απαιτήσεις συλλέγονται σε υψηλό επίπεδο αφαίρεσης και είναι επιφανειακές και οπτικά αντιληπτές.

- Η ανάπτυξη του συστήματος γίνεται επαναληπτικά σε μικρές και επαυξητικές εκδόσεις.
- Δίδεται έμφαση δίδεται στη συχνή παράδοση των προϊόντων προκειμένου να συλλέγονται παρατηρήσεις και σχόλια από τους εμπλεκόμενους χρήστες.
- Περαιτώνεται κάθε ενότητα (χαρακτηριστικό γνώρισμα) του συστήματος πριν τη μεταπήδηση σε ένα άλλο.
- Εφαρμογή του κανόνα 80/20, που, εν προκειμένω, υποδηλώνει ότι 80% των αποτελεσμάτων αναφορικά με την ανάπτυξη του συστήματος μπορούν να αποδοθούν στο (ή να προκύψουν από το) 20% των δραστηριοτήτων που αναλαμβάνονται.
- Οι έλεγχοι (δοκιμές) του συστήματος ενσωματώνονται στον κύκλο ζωής του έργου – η διενέργεια ελέγχων (δοκιμών) πρέπει να πραγματοποιείται νωρίς και συχνά.

Περιεχόμενα

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

Οι εμπνευστές των ευέλικτων μεθόδων και πρακτικών δημοσίευσαν τις κοινές τους αξίες και αρχές για την ανάπτυξη συστημάτων λογισμικού στο Μανιφέστο για την Ευέλικτη Ανάπτυξη Λογισμικού (*Agile Software Development*). Τα δώδεκα βασικά σημεία που περιλήφθηκαν στο μανιφέστο ευελιξίας είναι:

- i. Πρέπει να υπάρχει ικανοποίηση των πελατών μέσω της κατασκευής γρήγορων και συνεχών επαυξήσεων (*increments*).
- ii. Πρέπει να εγκατασταθεί η πρώτη επαύξηση εντός μερικών εβδομάδων και ολόκληρο το σύστημα εντός μερικών μηνών.
- iii. Πρέπει οι ομάδες πελατών και κατασκευαστών να εργάζονται από κοινού καθημερινά και καθ' όλη τη διάρκεια του έργου.
- iv. Πρέπει οι ομάδες πελατών και κατασκευαστών να πραγματοποιούν συναντήσεις πρόσωπο-με-πρόσωπο.

- v. Πρέπει να είναι καλοδεχούμενες τυχόν απαιτήσεις που να προτείνονται από τους χρήστες ακόμη και στα τελευταία στάδια του έργου.
- vi. Πρέπει να υπάρχει εμπιστοσύνη και σεβασμός μεταξύ των μελών της ομάδας κατασκευαστών.
- vii. Πρέπει να μετράται η ταχύτητα εκτέλεσης του έργου μετά την παράδοση κάθε επαύξησης.
- viii. Πρέπει να δίδεται έμφαση στον καλό σχεδιασμό για να αυξηθεί η ευελιξία.
- ix. Πρέπει να υπάρχει αυτό-οργάνωση για την καλύτερη αρχιτεκτονική και τον καλύτερο σχεδιασμό του συστήματος.
- x. Πρέπει να υπάρχει προσαρμογή και συντονισμός σύμφωνα με την κατάσταση και τις συνθήκες.

- xi.** Πρέπει να ακολουθείται η αρχή της απλότητας (*keep it simple - KIS*) καθ' όλη τη διεργασία ανάπτυξης του συστήματος.
- xii.** Πρέπει να υπάρχει συνεπής εργασία από την αρχή μέχρι το τέλος των έργων ευέλικτης ανάπτυξης.

Περιεχόμενα

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

- Η μεθοδολογία των κρυστάλλινων μεθόδων βασίζεται στη φιλοσοφική θεώρηση η οποία υπαγορεύει ότι τα πρωταρχικής σημασίας στοιχεία που επηρεάζουν την αποτελεσματικότητα και την αποδοτικότητα της εκτέλεσης των έργων ανάπτυξης συστημάτων είναι:
 - Οι άνθρωποι που εμπλέκονται στα έργα και οι αλληλεπιδράσεις μεταξύ τους,
 - Οι δεξιότητες κι τα ταλέντα των εμπλεκομένων στα έργα ανθρώπων καθώς και η τυπική και άτυπη επικοινωνία μεταξύ τους, και
 - Η κοινωνία με τα χαρακτηριστικά της αναφορικά με την επιδεκτικότητά της στην αλλαγή και στην καινοτομία και με τον ψηφιακό της πολιτισμό.
- Οι διεργασίες θεωρούνται σημαντικά μεν αλλά δευτερεύουσας σημασίας στοιχεία για την επιτυχή ανάπτυξη συστημάτων.

- Η φιλοσοφική αυτή θεώρηση σημαίνει ότι:
 - κάθε ομάδα ανάπτυξης λογισμικού διαθέτει διαφορετικά ταλέντα και διαφορετικές δεξιότητες, οπότε πρέπει να χρησιμοποιεί διαδικασίες που είναι μοναδικά προσαρμοσμένες σ' αυτήν.
 - πρέπει να ελαχιστοποιηθεί η πολυπλοκότητα των διαδικασιών αφού οι διαδικασίες δεν είναι ιδιαίτερα σημαντικές για την επιτυχία του έργου.

- Η μεθοδολογία δυναμικής ανάπτυξης συστημάτων (*Dynamic Systems Development - DSD*) αποτελεί μια εξέλιξη των πρακτικών της ταχείας ανάπτυξης συστημάτων (*Rapid Application Development - RAD*).
- Η μεθοδολογία DSD:
 - Βασίζεται στη φιλοσοφική θεώρηση ότι **“τίποτα δεν μπορεί να κατασκευάζεται τέλεια από την πρώτη φορά”** και
 - Θεωρεί την ανάπτυξη συστημάτων ως μια επαναληπτική και διερευνητική προσπάθεια.
 - Θεωρείται ως η μεθοδολογία που προβλέπει εκπαίδευση και τεκμηρίωση περισσότερο από κάθε άλλη μεθοδολογία ευέλικτης ανάπτυξης συστημάτων.

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

- Η μεθοδολογία DSD είναι πιθανώς η πρωταρχική μεθοδολογία ευέλικτης ανάπτυξης συστημάτων και, ενώ υπήρχε πριν την επινόηση του όρου “*agile* - ευέλικτη”, βασίζεται αποκλειστικά στις λεγόμενες ευέλικτες αρχές που ανάγονται στις ακόλουθες εννέα:
 - Ενεργή συμμετοχή των χρηστών
 - Παροχή στις ομάδες ανάπτυξης του συστήματος της δικαιοδοσίας για τη λήψη αποφάσεων
 - Εστίαση στη συχνή παράδοση νέων εκδόσεων του υπό ανάπτυξη συστήματος (των προϊόντων λογισμικού)
 - Χρησιμότητα του συστήματος ως βασικό κριτήριο για την αποδοχή του

- Επαναληπτική και επαυξητική διαδικασία ανάπτυξης του συστήματος προκειμένου να διασφαλιστεί ότι θα υπάρξει σύγκλιση προς τη βέλτιστη επιχειρησιακή λύση
- Αναστρέψιμες αλλαγές που επιβάλλονται κατά τη διάρκεια της ανάπτυξης του συστήματος
- Προσδιορισμός απαιτήσεων υψηλού επιπέδου
- Διενέργεια πλήρων και λεπτομερών ελέγχων (δοκιμών) κατά τη διάρκεια εκτέλεσης του κύκλου ζωής
- Προώθηση της συνεργασίας και της συνέργειας μεταξύ των εμπλεκόμενων στην ανάπτυξη του συστήματος.
- ...

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

Η Scrum είναι μια μεθοδολογία ευέλικτης ανάπτυξης συστημάτων η οποία:

- Βασίζεται στη φιλοσοφική θεώρηση ότι **“τα προβλήματα ανάπτυξης συστημάτων είναι εν πολλοίς αδόμητα και είναι δύσκολο να κατανοηθούν η να οριστούν στην απαιτούμενη έκταση και στο απαιτούμενο βάθος”**
- Μια βασική αρχή της μεθοδολογίας Scrum είναι η αναγνώριση ότι **θεμελιακά εμπειρικές προκλήσεις δεν μπορούν να αντιμετωπιστούν επιτυχώς με παραδοσιακούς διεργασιοστρεφείς μεθόδους (π.χ. τον καταρρακτοειδή κύκλο ζωής).**
- **MICRO-SERVICES integration** μπορεί να οδηγήσει σε διαφορετικό ολικό αποτέλεσμα

Η μεθοδολογία Scrum

- Είναι η πλέον δημοφιλής, ίσως, και ευρέως χρησιμοποιούμενη ευέλικτη μεθοδολογία, πιθανώς λόγω του γεγονότος ότι εφαρμόζεται σχετικά εύκολα και αντιμετωπίζει πολλά από τα διοικητικά θέματα που μάστιζαν την ανάπτυξη συστημάτων για πολλές δεκαετίες.
- Χαρακτηρίζεται από:
 - Ένα σύνολο εργασιών που ταξινομούνται κατά σειρά προτεραιότητας με βάση συγκεκριμένα κριτήρια.
 - Την περάτωση ενός συγκεκριμένου συνόλου εργασιών σε σύντομες επαναλήψεις.
 - Τη διενέργεια μιας σύντομης καθημερινής συνάντησης κατά την οποία αναλύεται η πρόοδος του έργου, περιγράφονται οι επικείμενες εργασίες και καταγράφονται τα ενδεχόμενα εμπόδια.

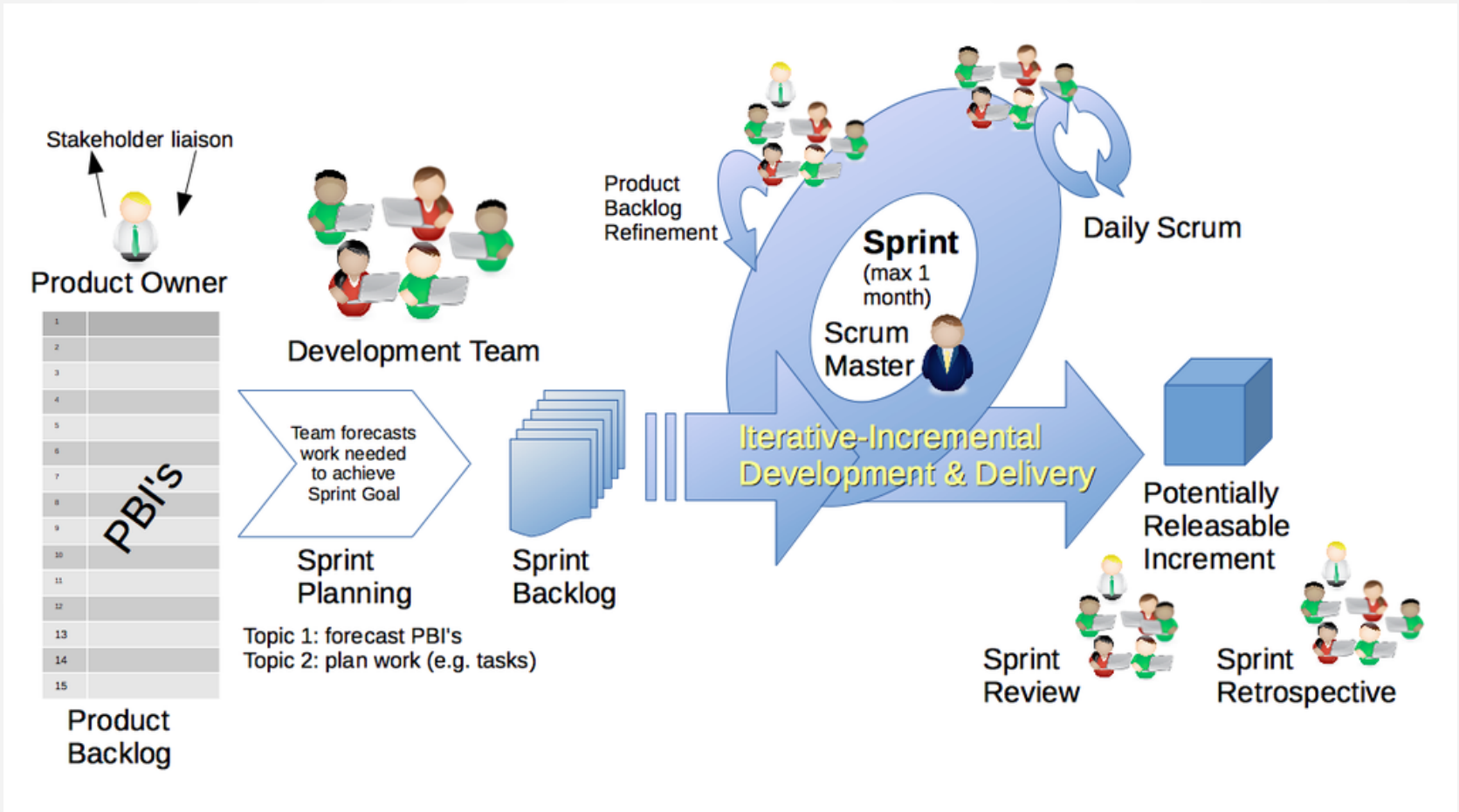
- Μια σύντομη ανασκόπηση της προηγούμενης επανάληψης με σχόλια από τα μέλη της ομάδας.
- Η μεθοδολογία Scrum προτρέπει τη δημιουργία αυτο-οργανωμένων ομάδων ενθαρρύνοντας την προφορική, πρόσωπο-με-πρόσωπο επικοινωνία μεταξύ των μελών τους και μεταξύ όλων των ειδικοτήτων (χρηστών, κατασκευαστών, κτλ) που εμπλέκονται στο έργο.
- Η εφαρμογή της μεθοδολογίας Scrum επιβλέπεται από ένα άτομο το οποίο:
 - Συντονίζει τις επιμέρους ομάδες
 - Έχει ως βασική αρμοδιότητα την εξάλειψη των ενδεχόμενων εμποδίων προκειμένου η ομάδα να παράξει και παραδώσει τα αποτελέσματα της τρέχουσας επανάληψης,

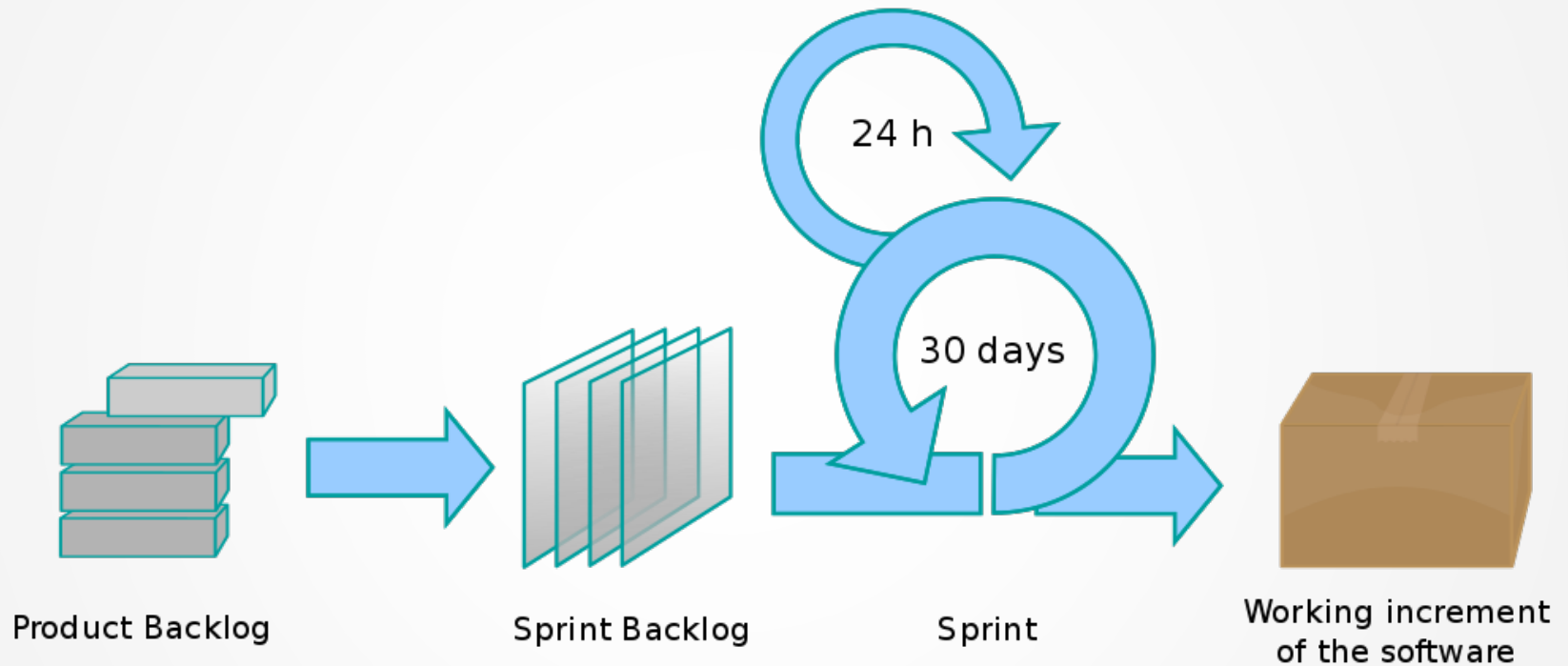
- Οι ομάδες Scrum αποτελούνται συνήθως από 7 +/- 2 μέλη και δεν έχουν επικεφαλής ομάδας να αναθέτουν καθήκοντα ή να αποφασίζουν πώς λύνεται ένα πρόβλημα.
- Η ομάδα ως μονάδα αποφασίζει πώς να αντιμετωπίσει θέματα και να λύσει προβλήματα.
- Υπάρχουν τρεις βασικοί ρόλοι σε μια ομάδα Scrum:
 - ▶ Ο **(product owner) ιδιοκτήτης προϊόντος** είναι ο βασικός παράγοντας του έργου - συνήθως ένας εσωτερικός ή εξωτερικός πελάτης ή ένας εκπρόσωπος του πελάτη. Υπάρχει μόνο ένας ιδιοκτήτης προϊόντων που μεταφέρει τη συνολική αποστολή και το όραμα του προϊόντος που κατασκευάζει η ομάδα.
 - ▶ Ο **Scrum Master** είναι ο ηγέτης των υπηρεσιών του ιδιοκτήτη προϊόντων, της ομάδας ανάπτυξης και της οργάνωσης. Χωρίς ιεραρχική εξουσία πάνω στην ομάδα, αλλά περισσότερο ως διευκολυντής, ο Scrum Master εξασφαλίζει ότι η ομάδα τηρεί τη θεωρία, τις πρακτικές και τους κανόνες του Scrum. Αυτό μπορεί να περιλαμβάνει την αφαίρεση των εμποδίων, τη διευκόλυνση των συναντήσεων και τη βοήθεια προς τον ιδιοκτήτη του προϊόντος, για την περιποίηση.
 - ▶ Η **Ομάδα Ανάπτυξης** είναι μια αυτο-οργανωτική, διαλειτουργική ομάδα εξοπλισμένη με όλες τις δεξιότητες για να παραδώσει φορτωτικές προσαυξήσεις κατά την ολοκλήρωση κάθε Sprint.

- **Backlog:** Είναι το πιο σημαντικό έγγραφο της διαδικασίας, όπου περιγράφεται κάθε απαίτηση για ένα σύστημα, ένα έργο ή ένα προϊόν. Το backlog μπορεί να θεωρηθεί ως λίστα υποχρεώσεων που αποτελείται από στοιχεία εργασίας, καθένα από τα οποία παράγει ένα παραδοτέο με επιχειρηματική αξία. Τα στοιχεία του backlog ταξινομούνται με βάση την σημαντικότητά τους από τον ιδιοκτήτη του προϊόντος.
- **Sprint Backlog:** Είναι το τμήμα με τα στοιχεία του backlog που αναμένεται να ολοκληρωθούν σε ένα συγκεκριμένο sprint.
- **Increment:** Είναι το σύνολο όλων των στοιχείων του backlog που έχουν ολοκληρωθεί από στην τελευταία έκδοση του λογισμικού.

- **Sprint:** Ένα sprint είναι μια χρονική περίοδος κατά την οποία η συγκεκριμένη εργασία ολοκληρώνεται και είναι έτοιμη για επανεξέταση. Η συνήθης διάρκεια είναι 2-4 εβδομάδες.
- **Sprint Planning:** Οι συναντήσεις ομάδων προγραμματισμού που καθορίζουν ποιες λειτουργίες θα παραδοθούν στο συγκεκριμένο sprint και πώς θα επιτευχθεί η εργασία.
- **Daily Scrum ή Stand-up:** Είναι μια σύντομη συνάντηση επικοινωνίας (όχι περισσότερο από 15 λεπτά) στην οποία κάθε μέλος της ομάδας καλύπτει γρήγορα και διαφανώς την πρόοδο από την τελευταία στάση, την προγραμματισμένη εργασία πριν από την επόμενη συνάντηση και τυχόν εμπόδια που ενδέχεται να εμποδίζουν την πρόοδό της.
- **Sprint Review:** Είναι η εκδήλωση επίδειξης για να παρουσιάσει το έργο που ολοκληρώθηκε κατά τη διάρκεια του sprint. Ο ιδιοκτήτης προϊόντος ελέγχει το έργο σύμφωνα με προκαθορισμένα κριτήρια αποδοχής και είτε δέχεται είτε απορρίπτει το έργο.
- **Sprint Retrospective:** είναι η τελική συνάντηση της ομάδας στο sprint για να καθορίσει τι πήγε καλά, τι δεν πήγε καλά και πώς μπορεί να βελτιωθεί η ομάδα στο επόμενο.

- Η Scrum αντιμετωπίζει την πολυπλοκότητα της εργασίας, καθιστώντας τις **πληροφορίες διαφανείς**, έτσι ώστε οι άνθρωποι να μπορούν να επιθεωρούν και να προσαρμόζονται με βάση τις τρέχουσες συνθήκες, παρά τις προβλεπόμενες συνθήκες.
- Η **διαφάνεια** των κοινών όρων και προτύπων απαιτείται στη μεθοδολογία Scrum, προκειμένου να διασφαλιστεί ότι αυτό που παραδίδεται είναι αυτό που αναμενόταν.
- Η συχνή επιθεώρηση εξασφαλίζει την πρόοδο και εντοπίζει νωρίς τις διαφοροποιήσεις από το αρχικό πλάνο, ώστε να γίνουν ρυθμίσεις γρήγορα.





- Η Scrum χρησιμοποιείται ευρέως από ομάδες ανάπτυξης λογισμικού.
- Στην πραγματικότητα είναι η πιο δημοφιλής, ευέλικτη μεθοδολογία.
- Σύμφωνα με την 12η ετήσια έκθεση του State of Agile, το 70% των ομάδων λογισμικού χρησιμοποιεί Scrum ή ένα υβριδικό Scrum.
- Ωστόσο, η Scrum έχει επεκταθεί σε άλλες επιχειρηματικές λειτουργίες, συμπεριλαμβανομένης της πληροφορικής και της εμπορίας, όπου υπάρχουν έργα που πρέπει να προχωρήσουν με την παρουσία πολυπλοκότητας και ασάφειας.

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

- Η μεθοδολογία ακραίου προγραμματισμού (μεθοδολογία XP):
 - Αποσκοπεί στην παροχή της δυνατότητας κατασκευής λογισμικού ακόμη και εντός ενός ασταθούς περιβάλλοντος επιτρέποντας σημαντική ευελιξία στη διεργασία μοντελοποίησης του συστήματος.
 - Ανήκει στην κατηγορία των ευέλικτων μεθοδολογιών ανάπτυξης συστημάτων και μπορεί να θεωρηθεί είτε ως μια νέα προσέγγιση είτε ως μια επέκταση της μεθόδου των εξελικτικών πρωτοτύπων ή των μεθοδολογιών ταχείας ανάπτυξης εφαρμογών.
- Αντί της τυπικής παράδοσης των απαιτήσεων από τους χρήστες στους κατασκευαστές, **οι χρήστες ενθαρρύνονται να παρέχουν παραδείγματα αναφορικά με τη χρησιμοποίηση του συστήματος σε συγκεκριμένες επιχειρησιακές καταστάσεις.**
- Η επικοινωνία μεταξύ κατασκευαστών και χρηστών είναι άτυπη και πραγματοποιείται σε καθημερινή βάση.

Η μεθοδολογία ΧΡ ακολουθεί τη γενική προσέγγιση της ευέλικτης ανάπτυξης συστημάτων που συνοψίζεται στο εξής:

“Αντί να κληθούν οι χρήστες να υπογράψουν τις απαιτήσεις και να αναμένουν μέχρι να τους παραδοθεί το τελικό σύστημα χωρίς να παρεμβαίνουν σε ενδιάμεσα στάδια, ενθαρρύνεται η συμμετοχή των χρηστών καθ’ όλη τη διάρκεια της διεργασίας ανάπτυξης του συστήματος και ενθαρρύνονται οι κατασκευαστές να ενσωματώσουν, εφόσον κριθεί αναγκαίο, νέες ή τροποποιημένες απαιτήσεις των χρηστών στο υπό ανάπτυξη σύστημα”.

- Κύριος στόχος της μεθοδολογίας XP είναι η μείωση του κόστους από αλλαγές των απαιτήσεων των χρηστών.
- Με παραδοσιακές μεθοδολογίες ανάπτυξης συστημάτων, όπως η καταρρακτοειδής μεθοδολογία, οι απαιτήσεις για το σύστημα προσδιορίζονται και συχνά “παγώνουν” στην αρχή του έργου της ανάπτυξης του συστήματος. Αυτό σημαίνει ότι το κόστος της αλλαγής των απαιτήσεων σε μεταγενέστερη φάση του έργου μπορεί να είναι πολύ υψηλό.
- Η μεθοδολογία XP βασίζεται σε τέσσερις βασικούς πυλώνες:
 - στην επικοινωνία (*communication*),
 - στην απλότητα (*simplicity*),
 - στην ανατροφοδότηση (*feedback*), και
 - στο θάρρος (*courage*).

- Βασικό θεμέλιο της όλης διεργασίας είναι ***“The KISS principle (Keep It Simple, Stupid)”***.
- Σ’ αυτό το πλαίσιο, η μεθοδολογία XP περιλαμβάνει τις ακόλουθες αρχές:
 - Παραγωγή της αρχικής έκδοσης του συστήματος εντός εβδομάδων προκειμένου να αξιολογηθεί γρήγορα από τους χρήστες που καλούνται να υποβάλλουν τις παρατηρήσεις τους,
 - Επινόηση απλών λύσεων έτσι ώστε να πραγματοποιούνται οι αναγκαίες αλλαγές σχετικά εύκολα,
 - Διαρκή βελτίωση της ποιότητας του σχεδιασμού έτσι ώστε η πραγμάτωση του επόμενου συνόλου απαιτήσεων να είναι λιγότερο δαπανηρή, και
 - Συνεχή έλεγχο προκειμένου να ανιχνεύονται τυχόν σφάλματα όσο το δυνατόν συντομότερα και, συνεπώς, να επιδιορθώνονται με το μικρότερο κόστος.

Για τη μεθοδολογία XP έχει εκδοθεί ερμηνευτικός οδηγός που στην πρώτη έκδοσή του περιλαμβάνονται δώδεκα βασικές πρακτικές οι οποίες ομαδοποιούνται στις ακόλουθες τέσσερις περιοχές:

- Ανατροφοδότηση μικρής κλίμακας (δοκιμαστική ανάπτυξη συστήματος, προγραμματιστικό παίγνιο, συμμετοχή ολόκληρης της ομάδας, προγραμματισμός κατά ζεύγη),
- Διαρκής διεργασία ανάπτυξης του συστήματος κατά ομάδες (διαρκής ολοκλήρωση ενοτήτων συστήματος, βελτίωση σχεδιασμού συστήματος, μικρές εκδόσεις συστήματος),
- Κοινή κατανόηση συστήματος (απλός σχεδιασμός συστήματος, προσομοίωση συστήματος, συλλογική ιδιοκτησία κώδικα συστήματος, προγραμματιστικά πρότυπα ή συμβατικότητες), και
- Flat: Ποιότητα ζωής προγραμματιστών (σταθερός και αποδεκτός χρόνος εργασίας, όπως η εβδομάδα 40 ωρών)

Στη δεύτερη έκδοση του ερμηνευτικού οδηγού της μεθοδολογίας XP καταγράφεται ένα σύνολο από πορισματικές πρακτικές, επιπλέον των βασικών πρακτικών, που προέρχονται από γενικά αποδεκτές βέλτιστες πρακτικές και ασκούνται στην ολότητά τους. Οι πρακτικές αυτές είναι οι ακόλουθες:

- Η αλληλεπίδραση μεταξύ των κατασκευαστών και των πελατών θεωρείται καλή πρακτική. Έτσι, μια ομάδα XP πρέπει να περιλαμβάνει έναν εκπρόσωπο του πελάτη ο οποίος θα είναι επιφορτισμένος με τον προσδιορισμό και την ιεράρχηση των εργασιών της ομάδας και ο οποίος θα μπορεί να απαντά σε ερωτήσεις άμεσα.
- Η διαρκής επιδίωξη της μάθησης θεωρείται καλή πρακτική. Αυτό σημαίνει, εν προκειμένω, ότι πρέπει να μειώνεται ο χρόνος της ανάπτυξης και της ανατροφοδότησης (μάθησης) με συχνούς και σύντομους ελέγχους.

- Η κατασκευή μικρών προγραμμάτων (μικρής έκτασης κώδικας) θεωρείται καλή πρακτική για αύξηση της πιθανότητας απρόσκοπτης λειτουργίας τους. Έτσι, οι προγραμματιστές XP συγγράφουν προγράμματα μόνο για την ικανοποίηση πραγματικών τρεχόντων αναγκών του έργου, και προσπαθούν πραγματικά να μειώσουν την πολυπλοκότητα και την επανάληψη στα προγράμματά τους.
- Η διαρκής ανασκόπηση των προγραμμάτων θεωρείται καλή πρακτική. Έτσι, οι προγραμματιστές XP εργάζονται κατά ζεύγη έτσι ώστε τα προγράμματα να ανασκοπούνται, και πιθανώς να διορθώνονται, καθώς κατασκευάζονται.

- Η διενέργεια συνεχών ελέγχων των προγραμμάτων θεωρείται καλή πρακτική. Έτσι, στη μεθοδολογία XP οι δοκιμές καταγράφονται πριν συγγραφούν τα προγράμματα. Τα προγράμματα θεωρούνται πλήρη όταν υποστούν επιτυχώς τη δοκιμασία των δοκιμών (αλλά τότε χρειάζεται ανακατασκευή τους προκειμένου να αφαιρεθεί η ενδεχόμενη πολυπλοκότητα). Το σύστημα ελέγχεται περιοδικά ή αμέσως μετά την κατασκευή του, χρησιμοποιώντας όλους τους προ-καθορισμένους αυτοματοποιημένους ελέγχους, προκειμένου να διασφαλιστεί ότι λειτουργεί αποτελεσματικά και αποδοτικά κατά τα αναμενόμενα.
- Συχνά αναφέρεται ότι η μεθοδολογία XP εφαρμόζεται μόνο σε μικρές ομάδες και, συνεπώς, μικρά έργα.

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

- Οι αντικειμενοστρεφείς προσεγγίσεις ανάπτυξης συστημάτων (*object-oriented systems development*):
 - Συμφωνούν με παλαιότερες μεθοδολογίες στο ότι οι επεξεργασίες (*processes*) και τα δεδομένα (*data*) είναι θεμελιώδη χαρακτηριστικά ενός συστήματος.
 - Διαφέρουν από παλαιότερες μεθοδολογίες στο ότι δεν μεταχειρίζονται (μοντελοποιούν) διαφορετικά τις επεξεργασίες και τα δεδομένα αλλά τα συνδυάζουν (*combine*) ή τα ενθυλακώνουν (*encapsulate*) εντός αντικειμένων (*objects*).
- Οι αντικειμενοστρεφείς προσεγγίσεις θεωρούν ότι ο υπό μελέτη κόσμος αποτελείται από αντικείμενα (αποτελούμενα από δεδομένα και επεξεργασίες αυτών) τα οποία διασυνδέονται και ομαδοποιούνται προκειμένου να αναπαραστήσουν σχηματικά διάφορες όψεις του.

- Η αναπαράσταση του πραγματικού κόσμου ως συνόλου αλληλοσυνδεδεμένων αντικειμένων προϋποθέτει την επιλογή ενός συγκεκριμένου επιπέδου αφάιρεσης (*level of abstraction*) ή επιπέδου ανάλυσης (*level of resolution*) προκειμένου η αναπαράσταση να είναι αφαιρετικά συνεπής.
- Τα αντικείμενα ενός επιπέδου αφάιρεσης δύνανται να:
 - Αναλύονται σε περισσότερα αντικείμενα κατωτέρου επιπέδου αφάιρεσης (ανωτέρου επιπέδου ανάλυσης) μέσω μιας διαδικασίας **από-το-γενικό-στο-ειδικό** (*top-down*) ή
 - Ομαδοποιούνται σε λιγότερα αντικείμενα ανωτέρου επιπέδου αφάιρεσης (κατωτέρου επιπέδου ανάλυσης) μέσω μιας διαδικασίας **από-το-ειδικό-στο-γενικό** (*bottom-up*).

Ειδικότερα, όλες οι αντικειμενοστρεφείς προσεγγίσεις για την ανάπτυξη ΠΣ χαρακτηρίζονται από τα ακόλουθα:

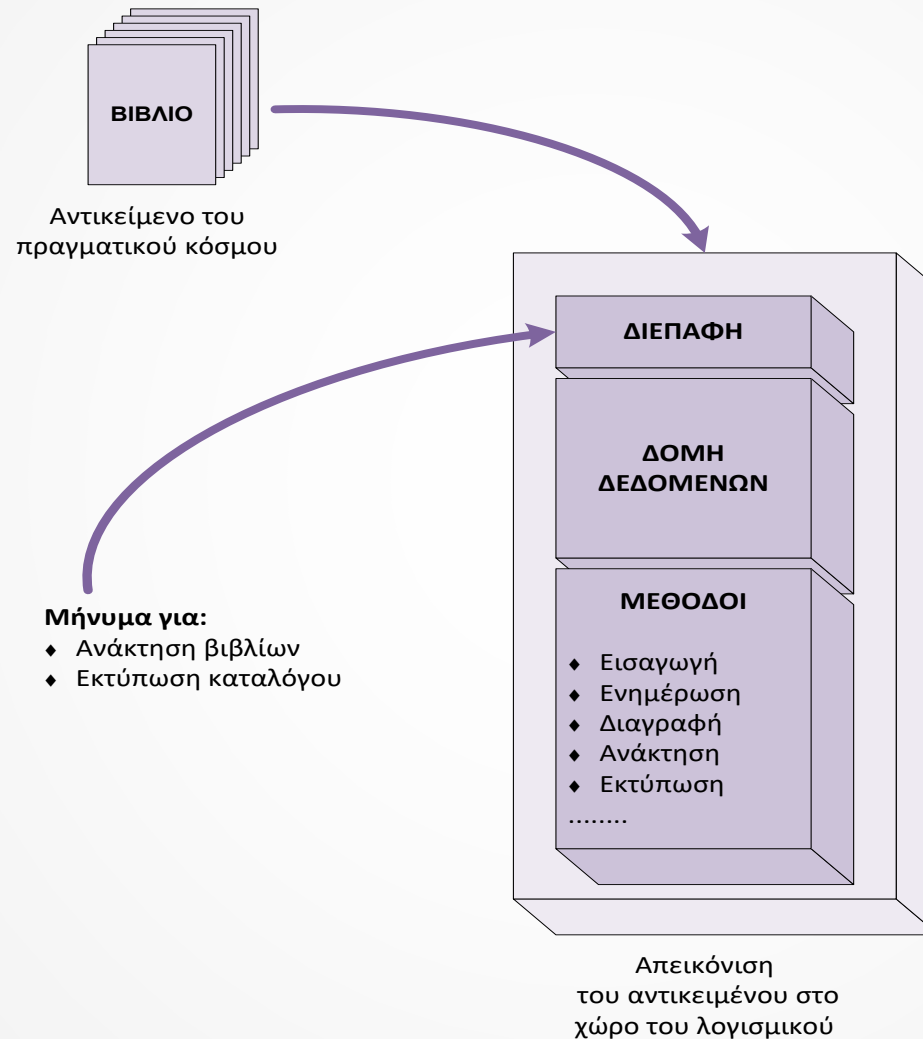
- Την τμηματοποίηση, που αναφέρεται στην διάσπαση ενός συστήματος σε επιμέρους εφαρμογές.
- Την απόκρυψη πληροφοριών που αναφέρεται στην εκτέλεση πράξεων εντός μιας εφαρμογής αποκρύπτοντας τις από άλλες εφαρμογές αλλά διαθέτοντας σε άλλες εφαρμογές συγκεκριμένες μόνο πληροφορίες,
- Τη ζεύξη, που αναφέρεται στην ολοκλήρωση των διαφόρων εφαρμογών, διατηρώντας την αυτονομία τους, προκειμένου να συστήσουν το όλο σύστημα, και
- Την ανθεκτικότητα, που αναφέρεται στην συντηρησιμότητα και την επεκτασιμότητα των εφαρμογών του υπό ανάπτυξη συστήματος.

Οι **βασικές έννοιες** που χρησιμοποιούνται στη αντικειμενοστρεφή προσέγγιση είναι οι ακόλουθες:

- **Αντικείμενο (object)** ονομάζεται κάτι επί του οποίου μπορεί να γίνουν πράξεις (μέθοδοι) που σχετίζονται με την υπό μελέτη προβληματική κατάσταση. Κάθε αντικείμενο μπορεί να είναι μια έννοια που αντιστοιχεί σε φυσικό, σε ιδεατό ή σε εικονικό συστατικό που αναφέρεται στην υπό μελέτη προβληματική κατάσταση. Διαθέτει μια μόνιμη ταυτότητα, βρίσκεται σε μια κατάσταση ανά πάσα στιγμή και επιδεικνύει μια συμπεριφορά. Η ταυτότητα το διακρίνει από άλλα αντικείμενα, η κατάσταση είναι η τρέχουσα τιμή των δυναμικών (όχι στατικών, μεταβαλλόμενων) χαρακτηριστικών (ιδιοτήτων) του αντικειμένου και η συμπεριφορά είναι οι πράξεις που μπορούν να γίνουν επί του αντικειμένου.

- **Κλάση αντικειμένων (object class)** ονομάζεται ένας τύπος (*type*) αντικειμένων έτσι ώστε κάθε αντικείμενο να συνιστά ένα στιγμιότυπο (*instance*) ή μια πραγματοποίηση (*occurrence*) μιας κλάσης αντικειμένων. Συνεπώς, όλα τα αντικείμενα μιας κλάσης αντικειμένων διαθέτουν ένα κοινό σύνολο ιδιοτήτων (*properties*) ή χαρακτηριστικών (*attributes*) που αναφέρονται στη δομή τους (*structure*) και στη συμπεριφορά τους (*behaviour*).
- Κάθε κλάση δεν περιέχει μόνο τα δεδομένα αλλά και τις πράξεις ή μεθόδους που μπορεί να εκτελέσει επί των δεδομένων αυτών. Οι μέθοδοι περιέχουν κωδικοποιημένους (*coded*) ελέγχους και λογικά τμήματα (προγράμματα) που μπορεί να καλούνται μέσω μηνυμάτων. Δηλαδή, μήνυμα (*message*) είναι μια επίκληση μιας κλάσης αντικειμένων για την εκτέλεση μιας μεθόδου της.

- Ο προσδιορισμός των **κλάσεων** που έχουν αναφορά στο πεδίο του προβλήματος καθώς και των **σχέσεων μεταξύ των διαφόρων κλάσεων** είναι ένα βασικό μέλημα κατά τον αντικειμενοστρεφή σχεδιασμό ενός συστήματος.
- Κάθε κλάση αντικειμένων αποτελείται από δύο τμήματα:
 - **Ιδιωτικό:** Συνίσταται από μια δομή δεδομένων και από το σύνολο των μεθόδων που εφαρμόζονται επί της δομής αυτής. Το ιδιωτικό τμήμα μιας κλάσης αντικειμένων ονομάζεται και μη ορατό τμήμα διότι τα συστατικά του αποκρύπτονται από κάθε άλλο αντικείμενο.
 - **Κοινόχρηστο:** Συνιστά το επίπεδο επικοινωνίας μεταξύ διαφορετικών αντικειμένων. Το κοινόχρηστο τμήμα ενός αντικειμένου ονομάζεται και τμήμα επικοινωνίας διότι είναι ορατό από κάθε άλλο αντικείμενο που επικοινωνεί μ' αυτό.

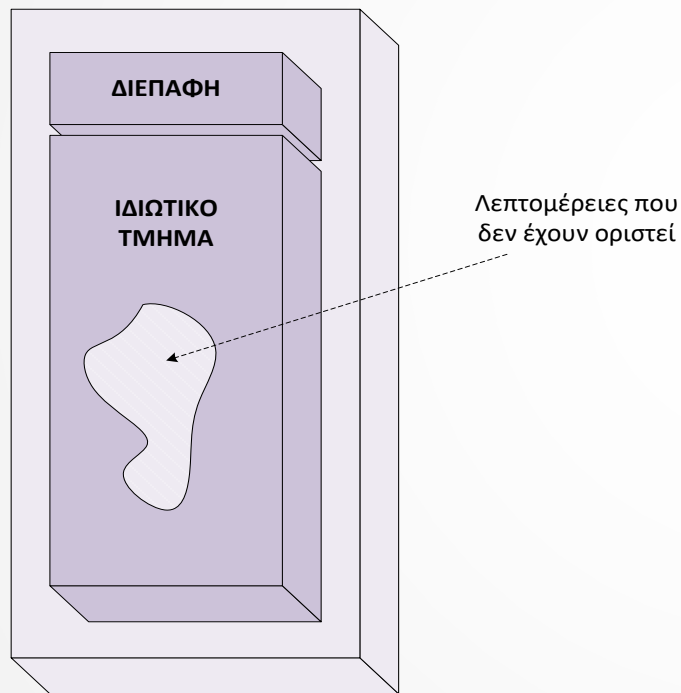


Ένα αντικείμενο του πραγματικού κόσμου (δηλαδή, η κλάση αντικειμένων “ΒΙΒΛΙΟ”) και η απεικόνισή του στο χώρο του λογισμικού

- Με το διαχωρισμό των κλάσεων αντικειμένων σε ιδιόχρηστο και κοινόχρηστο τμήμα επιτυγχάνεται η απόκρυψη πληροφοριών. Δηλαδή, οι λεπτομέρειες εκτέλεσης των μεθόδων μιας κλάσης αντικειμένων αποκρύπτονται από όλα τα λογικά τμήματα (*modules*) της εφαρμογής που δεν αφορούν σ' αυτήν την κλάση αντικειμένων.
- Ο προσδιορισμός των κλάσεων αντικειμένων (με τις δομές δεδομένων και τις μεθόδους τους) εκφράζει την ενυπάρχουσα τμηματικότητα των εφαρμογών. Δηλαδή, τα επιμέρους λογικά τμήματα μιας εφαρμογής (δεδομένα και μέθοδοι) ομαδοποιούνται ενώ προσδιορίζεται ένας τυπικός μηχανισμός επικοινωνίας μεταξύ τους (που είναι τα μηνύματα).

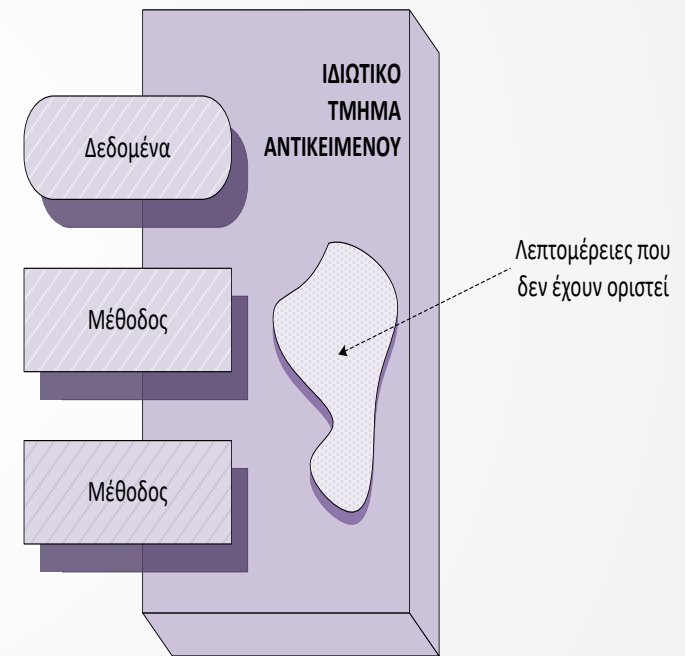
Κατά τη διεργασία του αντικειμενοστρεφούς σχεδιασμού του συστήματος, η περιγραφή μιας κλάσης αντικειμένων μπορεί να γίνει με τους ακόλουθους δύο τρόπους:

- Με την περιγραφή ενός πρωτοκόλλου, που προσδιορίζει τη **διεπαφή της κλάσης αντικειμένων** καθορίζοντας τα είδη των μηνυμάτων που μπορεί να δεχθεί και τις μεθόδους της που εκτελούνται όταν παραληφθεί το μήνυμα.
- Με την περιγραφή της **πραγμάτωσης, που δείχνει τις λεπτομέρειες πραγμάτωσης κάθε μεθόδου** της κλάσης αντικειμένων για την ικανοποίηση των απαιτήσεων κάθε μηνύματος που διαβιβάζεται σ' αυτήν.



(α)

Αναπαράσταση ενός λογικού τμήματος (module) εφαρμογής

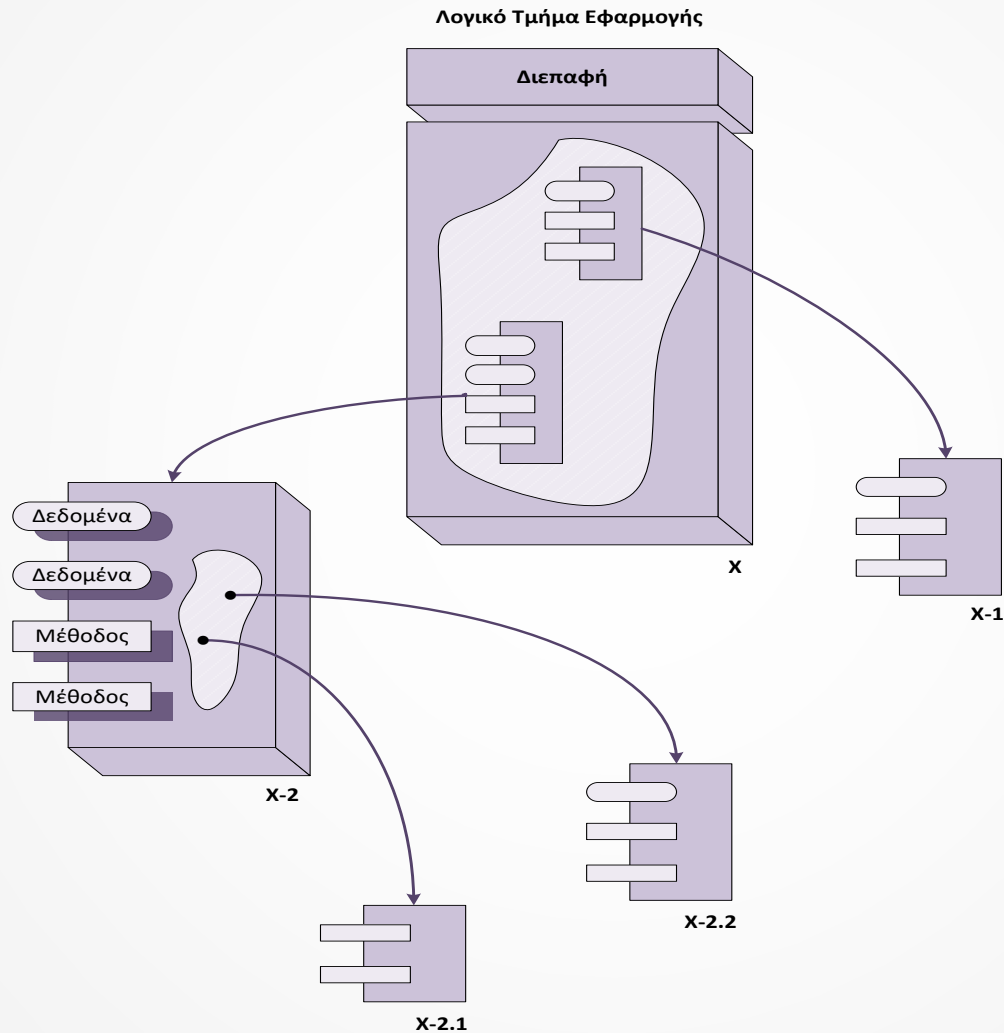


(β)

Συμβολισμός ιδιωτικού τμήματος ενός αντικειμένου.

- Ένα λογικό τμήμα (αντικείμενο) μιας αντικειμενοστρεφώς σχεδιασμένης εφαρμογής (επίσης αντικείμενο) μπορεί να αναπαρασταθεί με ένα παραλληλεπίπεδο.
- Το παραλληλεπίπεδο αποτελείται από το **κοινόχρηστο τμήμα**, που είναι ορατό από άλλα λογικά τμήματα, και από το **ιδιωτικό τμήμα** που αποκρύπτεται από άλλα λογικά τμήματα.
- Το μη μορφοποιημένο σχήμα εντός του ιδιωτικού τμήματος του αντικειμένου αναπαριστά τα **χαρακτηριστικά** ή τις **λογισμικές λεπτομέρειες** ή **απαιτήσεις του συστήματος που δεν έχουν προσδιοριστεί ακόμη** ή **δεν έχουν γίνει πλήρως κατανοητές**.
- Μετά τον προσδιορισμό ορισμένων χαρακτηριστικών ή λεπτομερειών ή απαιτήσεων του συστήματος, το ιδιωτικό τμήμα αναπαρίσταται όπως φαίνεται στο σχήμα (β) της προηγούμενης διαφάνειας (δομή δεδομένων και ένα σύνολο μεθόδων ή πράξεων επί των δεδομένων αυτών).

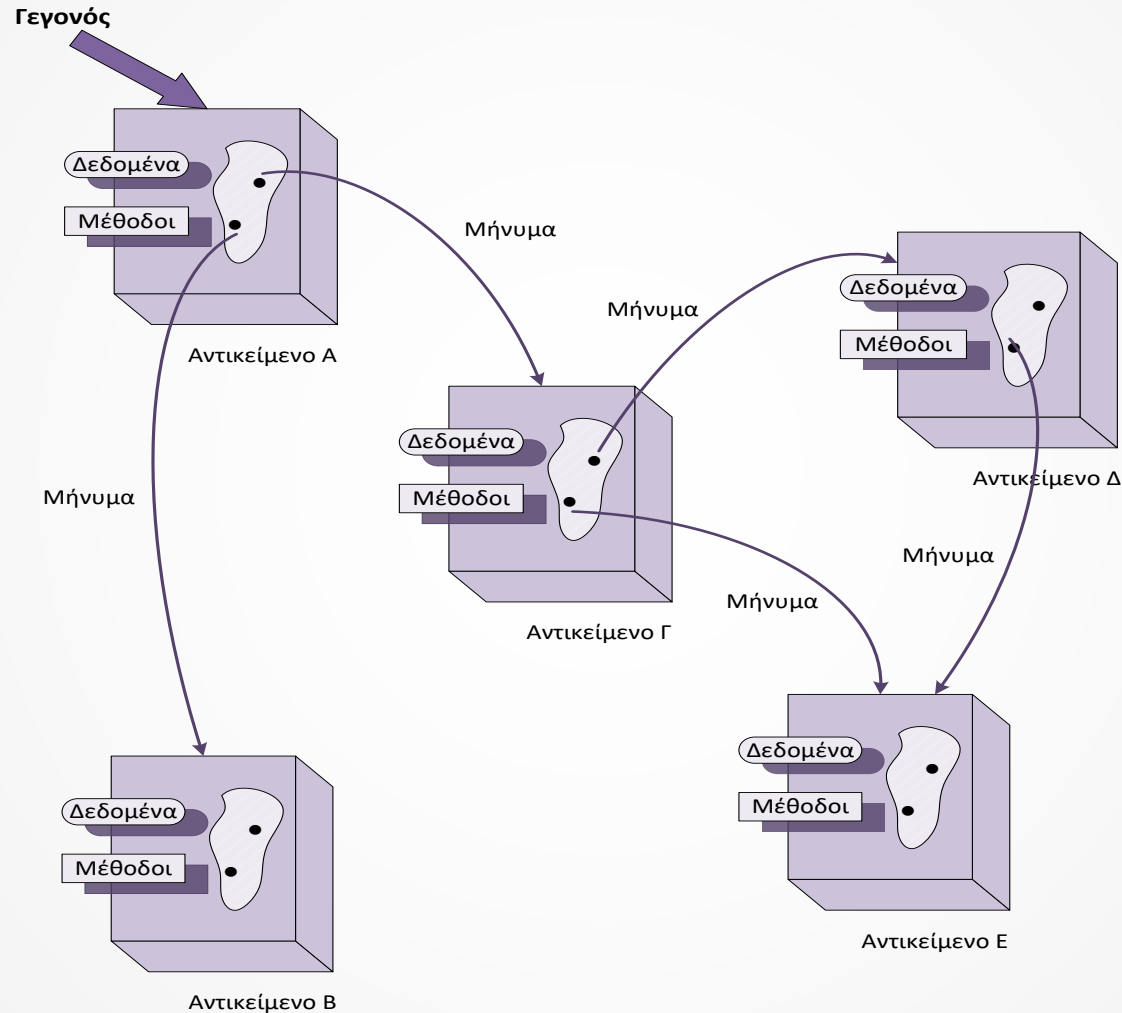
- Τα δεδομένα (πιο σωστά, η δομή δεδομένων) συμβολίζονται με κυρτά παραλληλεπίπεδα.
- Οι μέθοδοι που εκτελούνται επί των δεδομένων συμβολίζονται με γωνιαία παραλληλεπίπεδα.
- Το μη μορφοποιημένο σχήμα εντός του ιδιωτικού τμήματος του αντικειμένου αναπαριστά τα ακατάληπτα ή μη προσδιορισθέντα (ακόμη) χαρακτηριστικά ή λεπτομέρειες ή απαιτήσεις του συστήματος.



Λογικά τμήματα (modules) αντικειμενοστρεφών εφαρμογών.

- Οι συνδέσεις μεταξύ των επιμέρους λογικών τμημάτων (*modules*) μιας αντικειμενοστρεφώς σχεδιασμένης εφαρμογής (ιεραρχικά απεικονισμένες) φαίνονται στο σχήμα της προηγούμενης διαφάνειας.
- Σε υψηλό επίπεδο αφαίρεσης φαίνεται ότι το λογικό τμήμα (αντικείμενο) X μιας εφαρμογής εξαρτάται από τα δεδομένα και τις μεθόδους που περιέχονται στα λογικά τμήματα (αντικείμενα) X-1 και X-2.
- Ομοίως, το λογικό τμήμα X-2 εξαρτάται από τα δεδομένα και τις μεθόδους των λογικών τμημάτων (αντικειμένων) X-2.1 και X-2.2.
- Η διεργασία αυτή συνεχίζεται μέχρι το επιθυμητό επίπεδο λεπτομέρειας, δηλαδή μέχρι να εξαλειφθούν οι νησίδες ακατάληπτων ή μη προσδιορισθέντων χαρακτηριστικών ή λεπτομερειών ή απαιτήσεων του συστήματος.

- Συνεπώς, μια αντικειμενοστρεφής εφαρμογή (ένα αντικειμενοστρεφές πρόγραμμα) είναι απλώς μια συλλογή από αλληλο-συσχετιζόμενα αντικείμενα (ή κλάσεις αντικειμένων που είναι επίσης αντικείμενα) όπου οι συνδέσεις αναπαριστώνται από μονο-κατευθυνόμενους διαδρόμους αποστολής μηνυμάτων.
- **Κάθε εφαρμογή εκκινείται εξωτερικά, συχνά από ένα συμβάν ή γεγονός (*event*)** το οποίο πυροδοτεί ένα αντικείμενο το οποίο, με τη σειρά του, εκκινεί άλλα, κοκ.
- Οι ενεργοποιήσεις αυτές σχηματίζουν ένα δίκτυο από αντικείμενα που ομού συνιστούν την αντικειμενοστρεφή εφαρμογή.
- Μια διαγραμματική αναπαράσταση των αντικειμενοστρεφών εφαρμογών φαίνεται στο σχήμα της επόμενης διαφάνειας.



**Αντικείμενα που συνιστούν αντικειμενοστρεφείς εφαρμογές
(αντικειμενοστρεφή προγράμματα)**

- Η ανάπτυξη μιας εφαρμογής βασίζεται στην κατασκευή των αντικειμένων και στον προσδιορισμό του δικτύου (πλέγματος) ή της συνεργασίας μεταξύ των αντικειμένων.
- Ο διάδρομος (*path*) που ακολουθεί το πρόγραμμα της εφαρμογής διαμέσου του δικτύου ονομάζεται νήμα (*thread*) ή νήμα ελέγχου και μπορεί να είναι δύσκολα αναγνωρίσιμο μόλις κληθεί το πρόγραμμα καθότι τα αντικείμενα αλληλεπιδρούν κατά τρόπους που εξαρτώνται από τα γεγονότα εκκίνησής τους και τις αποκρίσεις των χρηστών.

- Η έννοια του αντικειμένου είναι αυτή του στιγμιότυπου (*instance*) ή της πραγματοποίησης (*occurrence*) μιας κλάσης, που σημαίνει ότι:
 - Τα αντικείμενα είναι μοναδικά και ότι διαφέρουν μόνον τα δεδομένα που περιέχονται σ' αυτά (δηλαδή, η ταυτότητα και η κατάσταση τους).
 - Η γενική συμπεριφορά των αντικειμένων της ίδιας κλάσης είναι ίδια.
- Όμως, επειδή δεν είναι δυνατόν να γραφεί ο κώδικάς για κάθε επιμέρους αντικείμενο αυτό που γίνεται στην πράξη είναι ότι ο κώδικας γράφεται μια φορά και ισχύει για όλα τα αντικείμενα της ίδιας κλάσης.

- Σύνοψη προηγούμενης διάλεξης
- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων
 - ▶ Ταχεία ανάπτυξη συστημάτων
 - ▶ Ευέλικτη ανάπτυξη συστημάτων
 - Εισαγωγή
 - Agile Software Development
 - Η μεθοδολογία των κρυστάλλινων μεθόδων
 - Η μεθοδολογία της δυναμικής ανάπτυξης συστημάτων
 - Η μεθοδολογία Scrum
 - Η μεθοδολογία ακραίου προγραμματισμού
- Η γενική αντικειμενοστρεφής προσέγγιση
 - ▶ Αντικειμενοστρεφής προγραμματισμός
 - ▶ Αντικειμενοστρεφής ανάλυση και σχεδιασμός

- Η γενική εικόνα για την Αντικειμενοστρεφή Ανάλυση και Σχεδιασμό – ΑΑΣ (*Object-oriented Analysis and Design – OOAD*) συνδυάζει στοιχεία από τρεις κατηγορίες σχεδιασμού:
 - τον σχεδιασμό των **δεδομένων του συστήματος**,
 - τον σχεδιασμό της **αρχιτεκτονικής δομής του συστήματος** (ή των εφαρμογών του), και
 - τον **σχεδιασμό των διεργασιών ή λειτουργιών** που χρησιμοποιούν τις εφαρμογές του συστήματος.

- Με τον ορισμό των αντικειμένων (ή κλάσεων αντικειμένων) προσδιορίζονται οι θεωρήσεις και αλληλο-συσχετίσεις των δεδομένων,
- Με τον ορισμό των μεθόδων καθορίζονται τα λογικά τμήματα και οι δομές των εφαρμογών, και
- Με την ανάπτυξη ενός μηχανισμού για τη χρησιμοποίηση των αντικειμένων (π.χ. παραγωγή μηνυμάτων) περιγράφονται οι διεπαφές για τη χρήση των εφαρμογών του συστήματος.

Η γενική διεργασία της αντικειμενοστρεφούς ανάλυσης και σχεδιασμού συστημάτων αποτελείται από τις ακόλουθες φάσεις:

- **Ορισμός του προβλήματος ή ανάλυση των απαιτήσεων**, όπου μπορεί να χρησιμοποιηθούν διάφορες τεχνικές για την διαγραμματική περιγραφή της προβληματικής κατάστασης και τη σκιαγράφηση του υπό ανάπτυξη συστήματος.
- **Ανάπτυξη μιας άτυπης στρατηγικής** για την αντιμετώπιση του προβλήματος που καθορίστηκε στην προηγούμενη φάση. Τα κύρια χαρακτηριστικά της στρατηγικής αυτής είναι ότι:
 - περιγράφεται σύντομα χρησιμοποιώντας απλούς όρους,
 - όλες οι έννοιες που περιγράφονται πρέπει να είναι του ίδιου επιπέδου λεπτομέρειας (αφαίρεσης),

- εστιάζεται στο “τι” πρέπει να γίνει για να επιλυθεί το πρόβλημα και όχι στις λεπτομέρειες για το “πως” θα επιτευχθεί η λύση του
- δεν περιλαμβάνει απαραίτητα όλες τις απαιτήσεις που καθορίσθηκαν στην ανάλυση των απαιτήσεων αλλά τις πιο σημαντικές εξ αυτών κατά την άποψη των κατασκευαστών.
- *Οριστικοποίηση της στρατηγικής ακολουθώντας τα εξής βήματα:*
 - προσδιορισμός των (τύπων) δεδομένων των αντικειμένων και των χαρακτηριστικών τους,
 - προσδιορισμός των μεθόδων που εκτελούνται επί των δεδομένων των αντικειμένων,
 - προσδιορισμός των διεπαφών επικοινωνίας κάθε αντικειμένου με άλλα αντικείμενα μέσω μηνυμάτων αναφορικά με την εκτέλεση των μεθόδων, και

- αντιμετώπιση ζητημάτων σχεδιασμού σε υψηλότερο επίπεδο λεπτομέρειας με στόχο μια περιγραφή της πραγμάτωσης των αντικειμένων.
- **Επανάληψη** των φάσεων (i), (ii), και (iii) μέχρι να επιτευχθεί ένας ολοκληρωμένος αντικειμενοστρεφής σχεδιασμός του συστήματος.

Η αντικειμενοστρεφής προσέγγιση:

- Αναπτύχθηκε προκειμένου να καλύψει το κενό της έλλειψης συνέργειας μεταξύ των διεργασιο-κεντρικών και δεδομένο-κεντρικών προσεγγίσεων ανάπτυξης συστημάτων.
 - Αποβλέπει, στην ολοκληρωμένη άποψη του οργανισμού, από τις οπτικές και των διεργασιών και των δεδομένων, και
 - Διακρίνει τον οργανισμό σε αλληλοεπιδρώντα και συλλειτουργούντα συστατικά.
- Η βασική ιδέα είναι ότι δεν μπορεί να επιτευχθεί εύκολα η διάσπαση ενός συστήματος σε ένα σύνολο διεργασιών (διεργασιο-κεντρική προσέγγιση) και σε ένα ανεξάρτητο σύνολο δεδομένων (δεδομένο-κεντρική προσέγγιση), καθώς και οι δύο αυτές όψεις του είναι πολύ στενά συνδεδεμένες μεταξύ τους.

- Η αντικειμενοστρεφής μεθοδολογία **επιτάσσει τη διάσπαση των προβλημάτων σε αντικείμενα.**
- Τα αντικείμενα θεωρούνται μέρη του συστήματος που περιέχουν και μεθόδους και δεδομένα.
- Ένα αντικείμενο μπορεί να ενσωματώνει μερικές δραστηριότητες διεργασιών (που αναφέρονται ως μέθοδοι του αντικειμένου) και μπορεί να έχει διάφορες καταστάσεις (που αναφέρονται ως χαρακτηριστικά του αντικειμένου).
- Έτσι, οι κατασκευαστές θα εστιάσουν στην ολιστική οντότητα (*holistic entity*) του συστήματος που εκτελεί μεθόδους και μεταφέρει δεδομένα, παρά σε μια μόνο όψη του, των δεδομένων (δεδομένο–κεντρική προσέγγιση) ή των μεθόδων (διεργασιο–κεντρική προσέγγιση).

- Σήμερα, η αντικειμενοστρεφής ανάπτυξη συστημάτων πραγματοποιείται συνήθως με τη χρήση ενός βασικού εργαλείου που ονομάζεται UML (*Unified Modeling Language*).
- Η UML είναι ένα σύνολο πρότυπων τεχνικών κατασκευής διαγραμμάτων και μοντέλων για την αναπαράσταση διαφόρων όψεων του υπό ανάπτυξη συστήματος.
- Η UML έχει γίνει αποδεκτή από την επιστημονική ένωση *Object Management Group (OMG)* ως πρότυπη διαγραμματική συμβολή στην αντικειμενοστρεφή ανάπτυξη συστημάτων (το πρώτο πρότυπο κατέστη διαθέσιμο το έτος 1997).

- Μια αντικειμενοστρεφής προσέγγιση για την ανάπτυξη συστημάτων πρέπει να έχει τα ακόλουθα χαρακτηριστικά:
 - Να καθοδηγείται από περιπτώσεις χρήσης (*use-case driven*),
 - Να είναι αρχιτεκτονικο-κεντρική (*architecture centric*),
 - Να είναι επαναληπτική και επαυξητική (*iterative and incremental*).

Καθοδήγηση από περιπτώσεις χρήσης

- Η αρχική τεχνική κατασκευής μοντέλων για τον ορισμό της συμπεριφοράς του συστήματος είναι τα μοντέλα περίπτωσης χρήσης.
- Οι περιπτώσεις χρήσης περιγράφουν πώς αλληλεπιδρούν οι χρήστες του συστήματος για την εκτέλεση μιας (σχετικά ευρείας) δραστηριότητας.
- Δεδομένου ότι κάθε περίπτωση χρήσης εστιάζει σε μία μόνο δραστηριότητα κάθε στιγμή, είναι εξαιρετικά απλή η αναπαράσταση της δραστηριότητας με τα αντίστοιχα διαγράμματα περιπτώσεων χρήσης (*use-case diagrams*).

Αρχιτεκτονικο-κεντρικότητα (1/3)

- Ο όρος αρχιτεκτονικο-κεντρικότητα σημαίνει την παροχή μιας άποψης του υπό κατασκευή συστήματος σε υψηλό επίπεδο αφαίρεσης (μικρή λεπτομέρεια).
- Η αρχιτεκτονική του λογισμικού εφαρμογών που έχει επιλεγεί πρέπει να οδηγεί στον προσδιορισμό, στην κατασκευή και στην τεκμηρίωση του συστήματος.
- Η αρχιτεκτονική του συστήματος πρέπει να υποστηρίζει τρεις θεωρήσεις του ΠΣ:
 - τη λειτουργική θεώρηση (*functional view*),
 - τη στατική θεώρηση (*static view*) και
 - τη δυναμική θεώρηση (*dynamic view*).

Αρχιτεκτονικο-κεντρικότητα (2/3)

- *Λειτουργική θεώρηση.* Περιγράφει τη συμπεριφορά του συστήματος από τη σκοπιά των χρηστών του. Στην UML, η αναπαράσταση της λειτουργικής θεώρησης του συστήματος γίνεται με τη χρήση των διαγραμμάτων περιπτώσεων χρήσης (*use-case diagrams*).
- *Στατική θεώρηση.* Περιγράφει τη δομή του συστήματος σε όρους κλάσεων δεδομένων (συμπεριλαμβανομένων των μεθόδων και των χαρακτηριστικών) και συσχετίσεων μεταξύ τους. Στην UML, η αναπαράσταση της στατικής θεώρησης του συστήματος γίνεται με τη χρήση των διαγραμμάτων κλάσεων (*class diagrams*).

Αρχιτεκτονικο-κεντρικότητα (3/3)

- *Δυναμική θεώρηση*. Περιγράφει την εσωτερική συμπεριφορά του συστήματος σε όρους επικοινωνίας μεταξύ αντικειμένων και αλλαγών καταστάσεων. Στην UML, η αναπαράσταση της δυναμικής θεώρησης (όψης) του συστήματος γίνεται με τη χρήση διαγραμμάτων δραστηριοτήτων (*activity diagrams*), διαγραμμάτων ακολουθιών (*sequence diagrams*), διαγραμμάτων συνεργασίας (*collaboration diagrams*) και διαγραμμάτων μετάβασης καταστάσεων (*state transition diagrams*).

Επαναληπτικότητα και επαυξητικότητα

- Το επαναληπτικό και επαυξητικό υπόδειγμα σημαίνει ότι κάθε επανάληψη της διεργασίας ανάπτυξης του συστήματος πρέπει να φέρει το σύστημα πιο κοντά στις απαιτήσεις, όπως αυτές έχουν προσδιοριστεί εξ αρχής ή προσδιορίζονται εξελικτικά εντός της διεργασίας ανάπτυξης.
- Ενώ ο ΚΖΑΣ είναι μια σταδιακή (κατά φάσεις) εκτελούμενη διεργασία, η αντικειμενοστρεφής προσέγγιση χρησιμοποιεί τα διαγράμματα της UML προκειμένου να κινηθεί από το εννοιολογικό και αφηρημένο πεδίο των φάσεων της ανάλυσης και του σχεδιασμού σε αυξημένης λεπτομέρειας πεδία της φάση της πραγμάτωσης (*implementation*) του συστήματος.