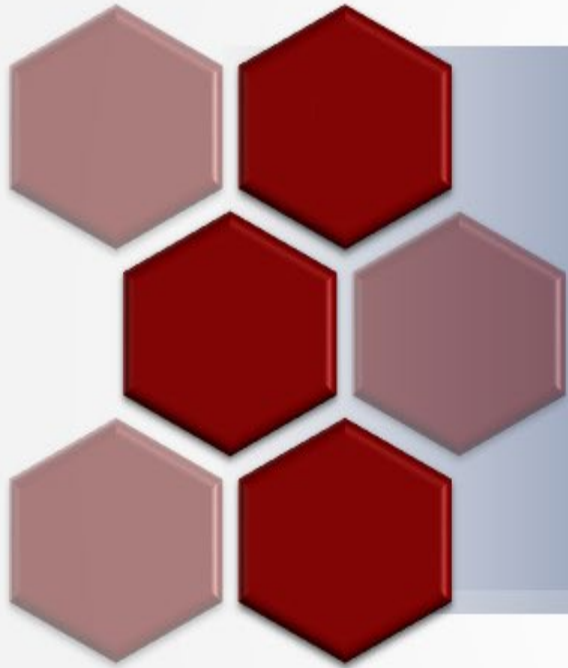




ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ



# Πληροφοριακά Συστήματα

- Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων

- Έννοιες (2-3 διαλ.)
- 1 πραγματικό σενάριο (1-2 διαλ.) με DevOps
  - ▶ Ανάλυση απαιτήσεων & σχεδιασμός έργου (φάσεων, πακέτα εργασίας)
  - ▶ Αρχιτεκτονική προσέγγιση & σχεδιασμός τεχνικής λύσης
  - ▶ Προσομοίωση υλοποίησης (change requests, etc)
  - ▶ Έλεγχος (unit testing, system and integration testing)
- Υλοποιήσεις ΠΣ (1-2 διαλ.)
  - ▶ Code versioning tools (SVN, etc)
  - ▶ Github (push code, merge, etc)
  - ▶ Gitlab (tracking issues, bugs, etc)
  - ▶ Services (micro-services), containers, orchestration of containers
  - ▶ junit, Jenkins, etc

## ■ Γραπτή εξέταση

- ▶ ???

- ▶ 7/10

## ■ (απαλλακτική) ΑΤΟΜΙΚΗ εργασία

- ▶ 4/10

- ▶ 10/10 πιθανή κατανομή σε ομαδικές εργασίες

- 1η ομάδα: απαιτήσεις & προδιαγραφές

- 2η ομάδα: σχεδιασμό έργου

- 3η ομάδα: υλοποίηση / προγραμματισμός

- 4η ομάδα: testing

# Μεθοδολογίες ανάπτυξης Πληροφοριακών Συστημάτων

Η μεθοδολογία λειτουργικής επαύξησης (*Functional Incrementation*) στοχεύει στη θεραπεία, κατά το δυνατόν, των αδυναμιών του καταρρακτοειδούς μοντέλου αναφορικά με την σειριακή, χωρίς επαναλήψεις, διεργασία ανάπτυξης τις ΠΣ.

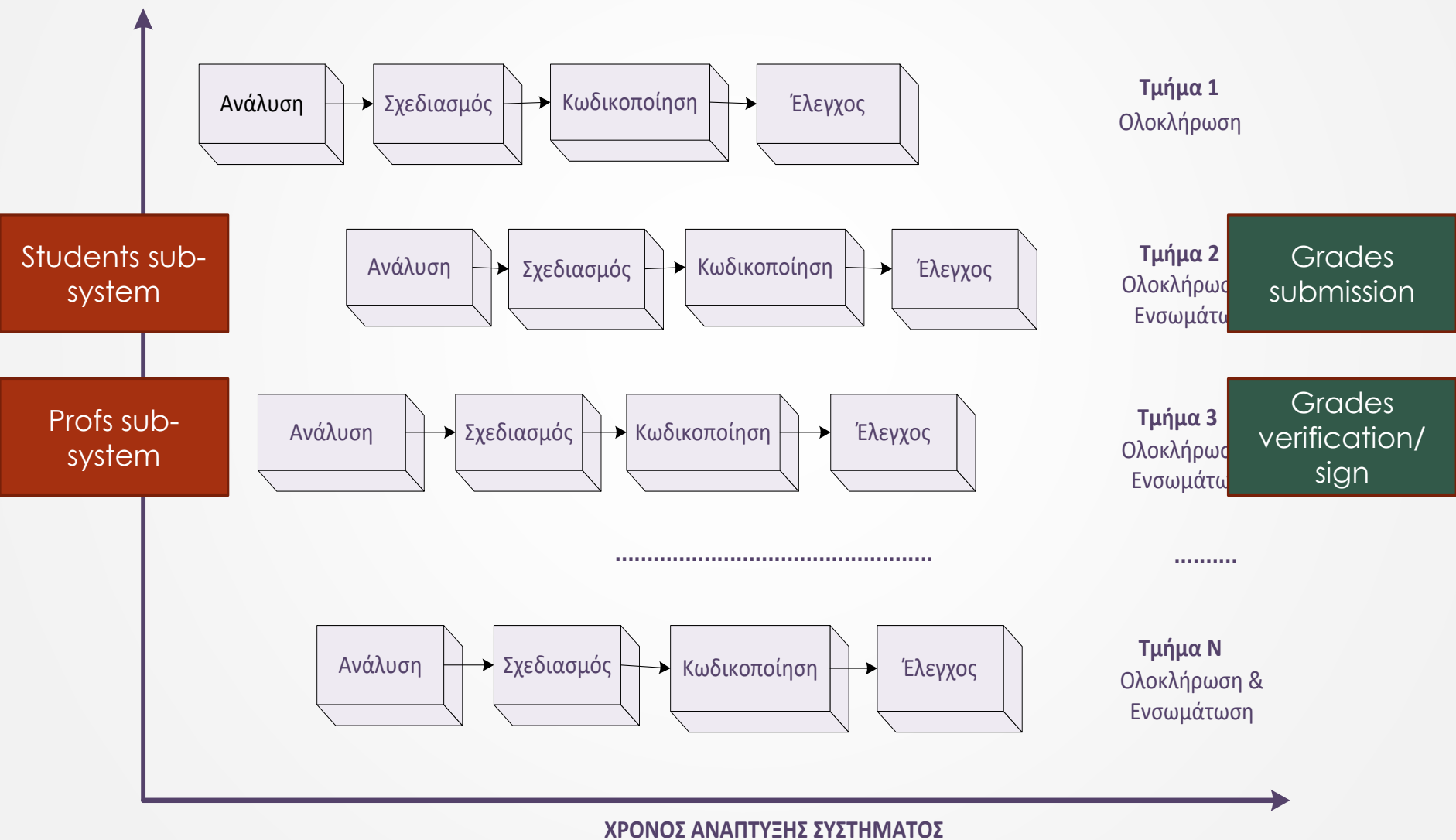
- Το υπό ανάπτυξη σύστημα διασπάται σε λειτουργικά τμήματα, ή λειτουργικές μονάδες (*functional units*), που είναι ανεξάρτητα μεταξύ τους.
- Το κάθε τμήμα αναπτύσσεται κατά τον **σειριακό τρόπο** που επιβάλλει το καταρρακτοειδές μοντέλο (τα λειτουργικά τμήματα μπορεί να ταυτίζονται ή όχι με οργανωτικά τμήματα και μπορεί να είναι επιχειρησιακές διεργασίες – *business processes* – που διαπερνούν οργανωτικές μονάδες του οργανισμού).

Στο πλαίσιο αυτής της μεθοδολογίας:

- Διενεργείται μια αρχική βάση ανάλυσης και σχεδιασμού του συστήματος με σκοπό τον προσδιορισμό των υποσυστημάτων στα οποία θα κατατμηθεί το όλο σύστημα και των λειτουργικών και τεχνικών απαιτήσεών τους (συμπεριλαμβανομένων των απαιτήσεων ασφάλειας).
- Οι εφαρμογές που αντιστοιχούν στα επιμέρους τμήματα αναπτύσσονται ανεξάρτητα και παράλληλα.
- Όταν αναπτυχθεί η εφαρμογή που αντιστοιχεί σε ένα λειτουργικό τμήμα, συνενώνεται με προηγουμένως αναπτυχθείσες εφαρμογές και ενσωματώνεται στο όλο σύστημα.

Έτσι, **σταδιακά, εντάσσονται οι προβλεπόμενες εφαρμογές**, που αντιστοιχούν σε λειτουργικές μονάδες, στο υπό ανάπτυξη ΠΣ και το σύστημα αποκτά το προβλεπόμενο από τις προδιαγραφές εύρος ώστε να καλύπτει το προδιαγραφέν λειτουργικό πεδίο (που μπορεί να εντάσσεται στον οργανισμό ή να διαπερνά τον οργανισμό).





- Η διάσπαση του όλου συστήματος σε επιμέρους εφαρμογές μπορεί να πραγματοποιηθεί σε κάποιο χρόνο = πριν την έναρξη της ανάλυσης των υποσυστημάτων.
- Η ανάπτυξη των επιμέρους εφαρμογών μπορεί να πραγματοποιείται σταδιακά σε μεταγενέστερο χρόνο και, πιθανώς, από διαφορετικό φορέα.
- Σ' αυτές τις περιπτώσεις πρέπει να δίδεται ιδιαίτερη σημασία αφενός στην τήρηση ανοικτών προτύπων διαλειτουργικότητας συστημάτων και την υιοθέτηση αρχιτεκτονικών ανεξάρτητων κατασκευαστών ώστε να είναι δυνατή η ολοκλήρωση και διαλειτουργικότητα των επιμέρους εφαρμογών (που έχουν αναπτυχθεί πιθανώς με διαφορετικές τεχνολογίες) και αφετέρου στην τεκμηρίωση των επιμέρους εφαρμογών του συστήματος αναφορικά με σκοπό τη βελτίωση της συντηρησιμότητάς τους.

- Η διαλειτουργικότητα συστημάτων και η διασυνδεσιμότητα δεδομένων πρέπει να αποτελεί βασικό μέλημα της διοίκησης του έργου και του ενδιαφερόμενου οργανισμού.

Τα κύρια **πλεονεκτήματα** τις μεθοδολογίας λειτουργικής επαύξησης είναι:

1. Παρέχεται η δυνατότητα **παράλληλης ανάπτυξης των επιμέρους εφαρμογών του συστήματος** με αποτέλεσμα τη συντόμευση της διαδικασίας ανάπτυξης του συστήματος. Η παράλληλη ανάπτυξη προϋποθέτει ότι η διοίκηση του έργου συντονίζει την όλη προσπάθεια αποτελεσματικά και από τεχνικής άποψης και από άποψης κατανομής πόρων. Για παράδειγμα, πρέπει να τίθενται, και να τηρούνται, συγκεκριμένα πρότυπα σε όλα τα επίπεδα, ακόμη και στο επίπεδο των διεπαφών του χρήστη, γεγονός που, συχνά, επιβάλλει την ανάλυση περισσότερων από τους προϋπολογισθέντες ανθρωπομήνες για την ανάπτυξη του συστήματος.

2. Το προκαθορισμένο σύστημα “χτίζεται” (κατασκευάζεται) σταδιακά, εμπλουτιζόμενο διαρκώς με περισσότερες εφαρμογές οι οποίες εντάσσονται στον κορμό του, κάθε μία από τις οποίες αντιστοιχεί σε ένα λειτουργικό τμήμα του οργανισμού. Παράλληλα, το γεγονός αυτό παρέχει τη δυνατότητα στις επιμέρους ομάδες ανάπτυξης των εφαρμογών του συστήματος να κατανοούν πληρέστερα τις απαιτήσεις που αντιστοιχούν στις επιμέρους εφαρμογές (αφού κάθε ομάδα ασχολείται με μία μόνο εφαρμογή) οπότε μειώνεται σημαντικά η πιθανότητα να παραβλεφθούν ή να μην γίνουν κατανοητές απαιτήσεις που θα έπρεπε να έχουν ικανοποιηθεί.

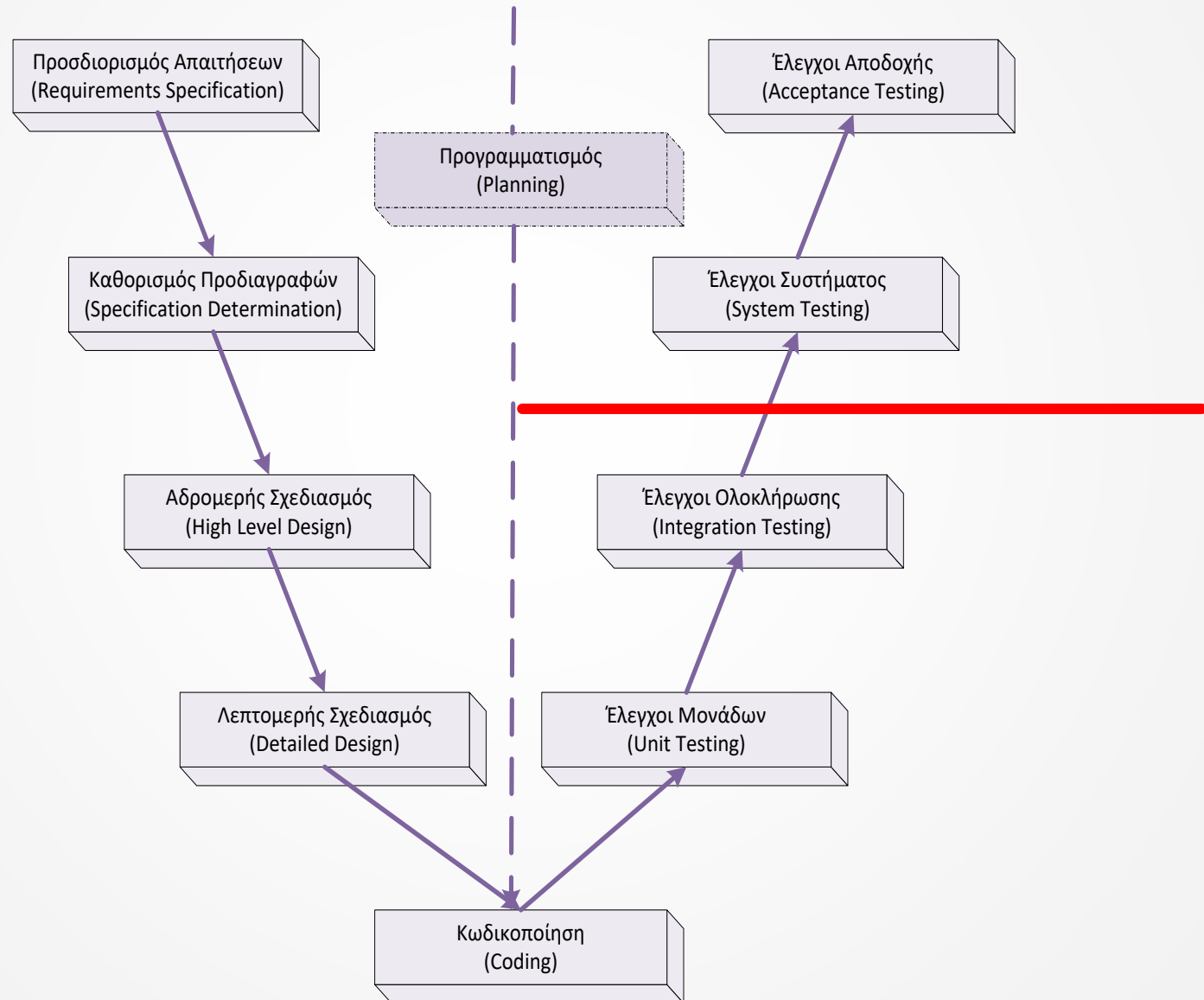
Τα κύρια **μειονεκτήματα** τις μεθοδολογίας τις λειτουργικής επαύξησης είναι:

1. Βασίζεται σε μεγάλο βαθμό **στην αρχική κατάτμηση του οργανισμού σε λειτουργικά τμήματα και στον αρχικό γενικό σχεδιασμό του συστήματος**. Αυτό μπορεί να δημιουργήσει προβλήματα δεδομένου ότι οι τεχνικοί που, κατά κόρον, δεν είναι εξοικειωμένοι με τη λειτουργία του οργανισμού καλούνται να διασπάσουν τον οργανισμό σε επιμέρους λειτουργικά τμήματα (πιθανώς ανεξάρτητα της οργανωτικής δομής) ή/και επιχειρησιακές διεργασίες. Αποτέλεσμα αυτού μπορεί να είναι η μη κατάλληλη κατάτμηση του οργανισμού σε επιμέρους λειτουργικά τμήματα και, κατ' ακολουθία, η μη κατάλληλη κατάτμηση του ΠΣ σε επιμέρους εφαρμογές.

2. Χρησιμοποιείται συνήθως για την ανάπτυξη συστημάτων μεγάλης κλίμακας όπου έχει νόημα η κατάτμηση του όλου συστήματος σε μικρότερες ενότητες και η παράλληλα ανάπτυξη αυτών. Επίσης, πρέπει να είναι γνωστές οι συνολικές απαιτήσεις του συστήματος και να μην μεταβάλλονται σημαντικά κατά τη διάρκεια της ανάπτυξής του δεδομένου ότι ακολουθείται η καταρρακτοειδής προσέγγιση στο επίπεδο των επιμέρους εφαρμογών του.

- Η μεθοδολογία V συνιστά μια παραλλαγή της καταρρακτοειδούς μεθοδολογίας δίνοντας ιδιαίτερη έμφαση στην **διασύνδεση των φάσεων του κύκλου ζωής ανάπτυξης συστημάτων** – ΚΖΑΣ (System Development Life Cycle **SDLC** με αντίστοιχες **φάσεις δοκιμών (ελέγχων)**).
- Η μεθοδολογία V προβλέπει την **πραγμάτωση κάθε φάσης με βάση την τεκμηρίωση της προηγούμενης φάσης** προκειμένου να ελέγχονται και να εγκρίνονται τα παραδοτέα κάθε φάσης της διεργασίας πριν αυτή δρομολογηθεί προς την επόμενη φάση.
- Με τον διαρκή έλεγχο των διεργασιών και της αντίστοιχης τεκμηρίωσης βελτιώνεται η αποδοτικότητα της διεργασίας ανάπτυξης του συστήματος επειδή τα ενδεχόμενα προβλήματα μπορούν να εντοπιστούν και να επιλυθούν έγκαιρα.





- Η μεθοδολογία V περιγράφει τρία διαδοχικά επίπεδα της ανάπτυξης συστημάτων που αναφέρονται ως:
  - απαιτήσεις (ολικού συστήματος),
  - σχεδιασμός υψηλού επιπέδου αφαίρεσης–μικρής λεπτομέρειας–(αρχιτεκτονική συστήματος), και
  - σχεδιασμός χαμηλού επιπέδου αφαίρεσης–μεγάλης λεπτομέρειας–(συνιστώσες λογισμικού).
- Για κάθε ένα απ' αυτά τα επίπεδα υπάρχει ένα αντίστοιχο επίπεδο προγραμματισμού (*planning*) και ελέγχου (*testing*).
- Ο προγραμματισμός είναι ο άξονας που υπάρχει μεταξύ της αριστερής και της δεξιάς πορείας του παραπάνω σχήματος που συνιστούν το σύμβολο-γράμμα V, και συνιστά το ενδιάμεσο σύνολο ενεργειών μεταξύ του σχεδιασμού και του ελέγχου.

- Επειδή η μεθοδολογία V προσβλέπει στη διόρθωση των σφαλμάτων του συστήματος αμέσως μόλις εντοπιστούν (πριν την εγκατάσταση και λειτουργία του συστήματος), μειώνεται το κόστος συντήρησης.
- Επειδή ο έλεγχος είναι διασπασμένος σε όλη τη διεργασία ανάπτυξης του συστήματος, συμμετέχουν, με την αναλογούσα ευθύνη, όλα τα εμπλεκόμενα μέρη στην ανάπτυξη του συστήματος. Αυτό σημαίνει ότι οι εφαρμοζόμενες μέθοδοι δοκιμών (ελέγχων) είναι επαρκείς για κάθε φάση του ΚΖΑΣ.
- Το γεγονός ότι πραγματοποιούνται έλεγχοι από την έναρξη της διεργασίας ανάπτυξης του συστήματος αυξάνει την αποτελεσματικότητά της μεθοδολογίας.

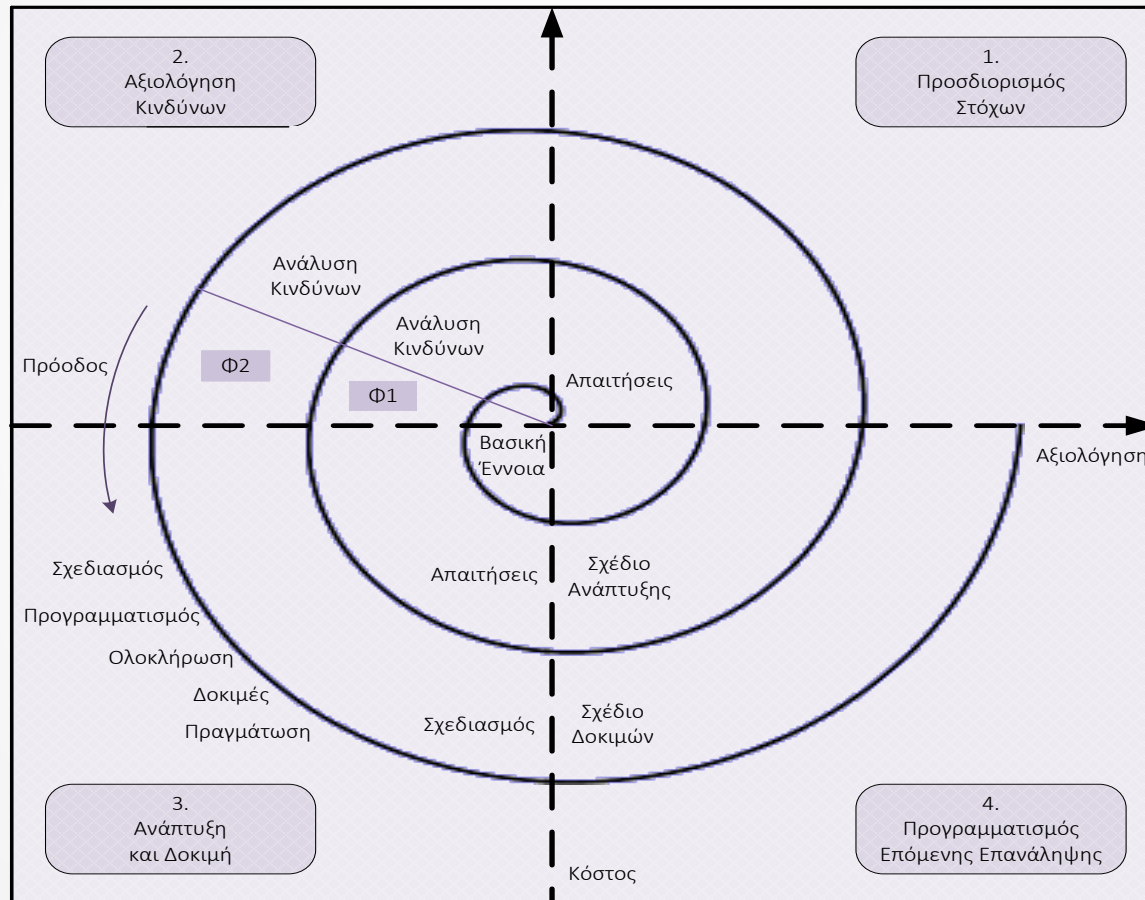
Επειδή η πραγμάτωση της μεθοδολογίας V **απαιτεί συχνά σημαντικούς πόρους** (μετρούμενους σε ανθρωπο-προσπάθεια και κόστος), συνήθως **προτείνεται για μεγάλα και πολύπλοκα έργα ανάπτυξης ΠΣ που αφορούν σε μεγάλους οργανισμούς.**

- Η μεθοδολογία σπειροειδούς ανάπτυξης συστημάτων – ΣΑΣ (*Spiral System Development – SSD*) επιβάλλει:
  - τον διαχωρισμό του υπό ανάπτυξη ΠΣ σε κύκλους εργασίας ή επαναλήψεις και
  - την σταδιακή αναβάθμιση των λειτουργικών χαρακτηριστικών του συστήματος από τον ένα κύκλο στον άλλον (μία επανάληψη στην άλλη).
- Η βασική διαφορά του σπειροειδούς μοντέλου από το επαυξητικό μοντέλο είναι ότι τώρα ορίζεται “ο κύκλος (η επανάληψη)” ως η βάση για την προσθήκη αναβαθμισμένων λειτουργικών χαρακτηριστικών του συστήματος.
- Η βασική καινοτομία του σπειροειδούς μοντέλου είναι ότι πριν από κάθε κύκλο (επανάληψη) διενεργείται ανάλυση κινδύνων αναφορικά με την επικείμενη εκτέλεση του κύκλου αυτού.

- Η διεργασία ανάπτυξης του συστήματος αποτελείται από έναν ΚΖΑΣ ο οποίος εκτελείται ως μια σειρά κύκλων ή επαναλήψεων.
- Κάθε κύκλος αρχίζει με τον προσδιορισμό των στόχων και των απαιτήσεων του τρέχοντος σταδίου ανάπτυξης του συστήματος και με μια ανάλυση των εναλλακτικών λύσεων και των περιορισμών.
- Η διεργασία αυτή θα **φωτίσει περιοχές αβεβαιότητας (κινδύνων)** οι οποίες θα ληφθούν υπόψη κατά το επόμενο βήμα της σκιαγράφησης της στρατηγικής (*strategy*) ή του προγράμματος (*plan*) μέσω μιας διαδικασίας προτυποποίησης ή άλλων τεχνικών προσομοίωσης.

- Η μεθοδολογία ΣΑΣ επιφέρει τη **διαρκή βελτίωση του πρωτοτύπου καθώς μειώνονται οι κίνδυνοι.**
- Καθώς οι κίνδυνοι μειώνονται σε αποδεκτά επίπεδα, **εκτελείται το επόμενο βήμα** σύμφωνα με το βασικό καταρρακτοειδές μοντέλο με αλληλουχία των φάσεων:
  - βασική έννοια (concept),
  - απαιτήσεις (requirements),
  - σχεδιασμός (design) και
  - πραγμάτωση (implementation).

Μόλις περατωθεί αυτός ο κύκλος, αρχίζει ο επόμενος καθώς δημιουργείται μια νέα, επαυξημένη έκδοση του συστήματος.





Κάθε κύκλος ή επανάληψη της μεθοδολογίας ΣΑΣ επιδεικνύει έξι σημαντικά χαρακτηριστικά που ονομάζονται **κανόνες (invariants)**:

- **Κανόνας 1** – είναι ο **ταυτόχρονος ορισμός των βασικών στοιχείων**, όπως είναι η βασική έννοια του συστήματος (concept), οι απαιτήσεις (requirements), το πρόγραμμα (plan), ο σχεδιασμός (design) και η κωδικοποίηση (coding).
- **Κανόνας 2** – είναι ότι **κάθε κύκλος ακολουθεί τέσσερις στρατηγικές αρχές** που αντιστοιχούν στα τέσσερα τεταρτημόρια του μοντέλου και είναι: καθορισμός στόχων, αξιολόγηση κινδύνων, ανάπτυξη και δοκιμή, και πρόγραμμα της επόμενης επανάληψης. Αν δεν ακολουθηθούν αυτές οι αρχές μπορεί να επηρεαστεί αρνητικά η όλη διεργασία.

- **Κανόνας 3** – είναι ότι το **επίπεδο της προσπάθειας που καταβάλλεται καθορίζεται από τη αξιολόγηση των κινδύνων**. Πρέπει να καθοριστούν **λογικά (ρεαλιστικά) χρονικά πλαίσια** για την εκτέλεση κάθε έργου σύμφωνα με τις εκτιμήσεις των κινδύνων και τον καθορισμό του “πόσο είναι αρκετό” για κάθε δραστηριότητα κάθε κύκλου της διεργασίας.
- **Κανόνας 4** – είναι ότι ο **βαθμός λεπτομέρειας καθοδηγείται από τις αξιολογήσεις των κινδύνων**. Όπως και ο κανόνας 3, είναι σημαντικό εδώ να καθοριστεί “πόση λεπτομέρεια είναι αρκετή” για κάθε κύκλο τις διεργασίας.
- **Κανόνας 5** – αναφέρεται στη χρήση βασικών οροσήμων, που ονομάζονται Στόχοι του Κύκλου Ζωής, Αρχιτεκτονική του Κύκλου Ζωής, και Αρχική Λειτουργική Ικανότητα. Σε κάθε ορόσημο θα υπάρχει αξιολόγηση των βασικών στοιχείων από τις εμπλεκόμενους.

- **Κανόνας 6** – δηλώνει ότι **επιπλέον του κατασκευαστικού μέρους, η διεργασία ανάπτυξης του συστήματος χρειάζεται να εστιάσει και στον συνολικό κύκλο ζωής.** Αυτό σημαίνει ότι οι μακροπρόθεσμοι προβληματισμοί αναφορικά με την ανάπτυξη και λειτουργία του συστήματος πρέπει επίσης να λαμβάνονται υπόψη.

- Η μεθοδολογία ΣΑΣ έχει σημαντικά πλεονεκτήματα έναντι προηγούμενων μεθοδολογιών.
- Η έμφαση στην ανάλυση κινδύνων παρέχει μια σημαντική βελτίωση και την καθιστά ιδανικό μοντέλο για μεγάλα και κρίσιμης αποστολής έργα.
- Δεν είναι πολύ αποτελεσματική για μικρότερα έργα δεδομένου ότι η εκτίμηση των κινδύνων μπορεί να αυξήσει τις δαπάνες του συστήματος σε βαθμό που η κατασκευή του συστήματος μπορεί να αποδειχθεί οικονομικά αναποτελεσματική ανεξάρτητα των κινδύνων. Η διαδικασία εκτίμησης των κινδύνων απαιτεί σημαντική εξειδίκευση και χρειάζεται να προσαρμοστεί σε κάθε σύστημα γεγονός το οποίο θα αυξήσει περαιτέρω το κόστος του έργου.

- Γενικά, το σπειροειδές μοντέλο κύκλου ζωής θεωρείται ένα **προηγμένο μοντέλο** κύκλου ζωής που **εστιάζει στον πρώιμο προσδιορισμό και μείωση των κινδύνων του έργου**.
- Σε ένα “σπειροειδές” έργο:
  - Πραγματοποιείται μια κατασκευή του υπό ανάπτυξη συστήματος σε μικροκλίμακα,
  - Ανιχνεύονται οι κίνδυνοι του έργου με βάση αυτήν,
  - Κατασκευάζεται ένα πρόγραμμα για τη διαχείριση των κινδύνων και,
  - Αποφασίζεται αν θα προβεί στο επόμενο βήμα της εκτέλεσης του έργου στις πραγματικές του διαστάσεις, δηλαδή αν θα πραγματοποιηθεί η επόμενη επανάληψη της σπείρας.

- Η σπειροειδής μεθοδολογία επεκτείνει το καταρρακτοειδές μοντέλο εισάγοντας στη διαδικασία ανάπτυξης την **προτυποποίηση (*prototyping*)** και συχνά προτιμάται από το καταρρακτοειδές μοντέλο για **μεγάλα, δαπανηρά και περίπλοκα έργα.**

- Ο κύκλος ζωής της μηχανευτικής της ευχρηστίας (*Usability Engineering Life Cycle*):
  - Συνιστά έναν ΚΖΑΣ ο οποίος θέτει τα **χαρακτηριστικά χρηστικότητας του συστήματος στο επίκεντρο** της διεργασίας ανάπτυξης και, ως εκ τούτου, συνεπάγεται τη διεξαγωγή συνεχών μετρήσεων και την διενέργεια ανάλυσης χρηστικότητας με την πρόοδο του έργου.
  - Αναφέρεται **πρωτίστως στο σχεδιασμό των διεπαφών** με τον χρήστη.
- Ο κύριος στόχος της μεθοδολογίας είναι η εφαρμογή ενός **δομημένου επαναληπτικού σχεδιασμού** και μιας διαδικασίας αξιολόγησης σε όλες τις φάσεις του ΚΖΑΣ, διασφαλίζοντας έτσι τη διαρκή συμμετοχή των χρηστών σ' αυτόν.

- Ο κύκλος ζωής ευχρηστίας διακρίνεται σε τρία μέρη:
  - Ανάλυση/Σχεδιασμός (*analysis/design*),
  - Ανάπτυξη (*development*), και
  - Αξιολόγηση (*evaluation*).
- Οι διαδοχικές δραστηριότητες των τριών αυτών μερών **εκτελούνται κατά την καταρρακτοειδή προσέγγιση, αλλά επιτρέποντας κάποιες επαναλήψεις** μέχρι την παράδοση του τελικού συστήματος.
- Κατά τη διάρκεια αυτής της διεργασίας είναι κρίσιμο να καθοριστεί **ποιοι θα αποτελέσουν την βάση χρηστών (*user base*)** ποια θα είναι η αξιοποίηση του συστήματος-προϊόντος.



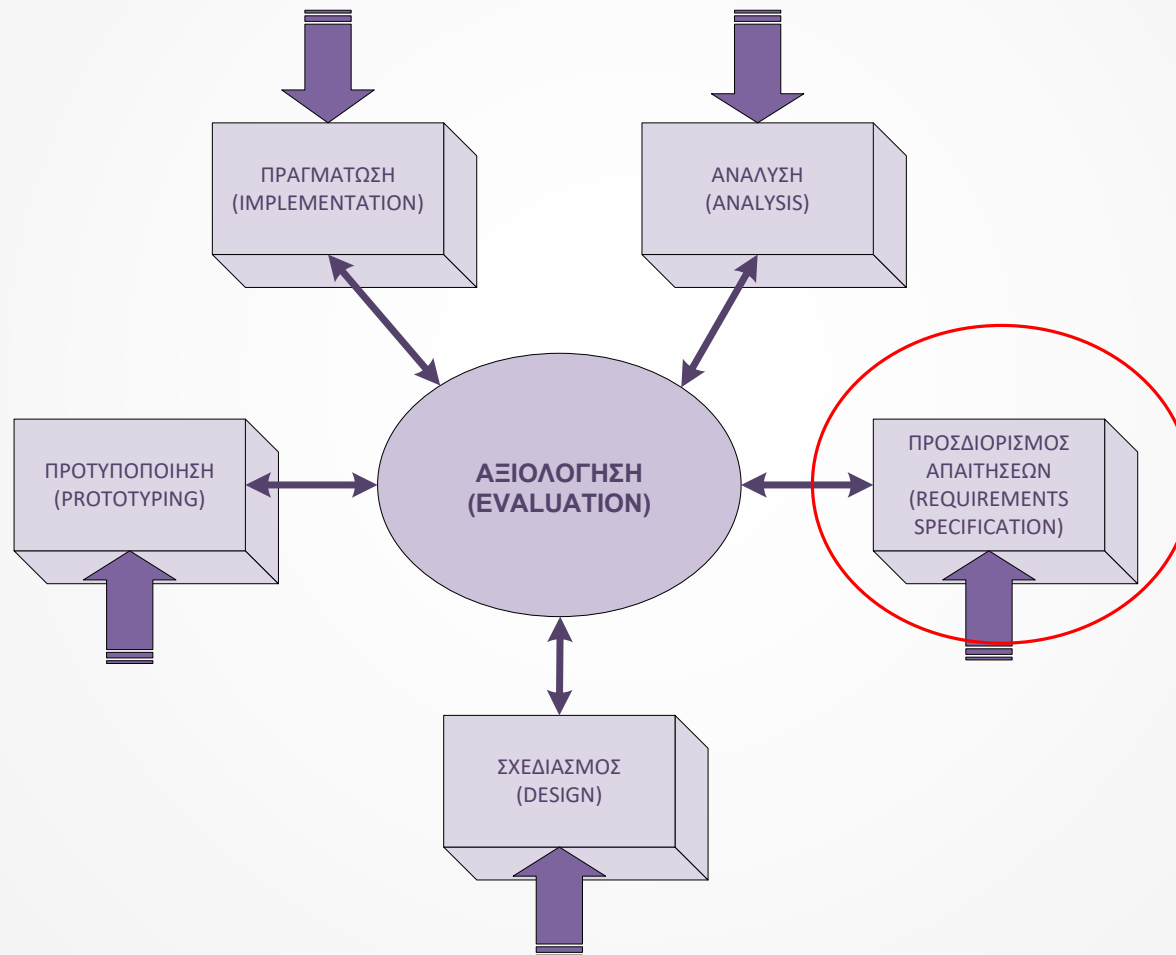
- Για το σκοπό αυτό, στην αρχή της διεργασίας ανάπτυξης του συστήματος αναλαμβάνεται μια σημαντική δραστηριότητα που αναφέρεται ως **ανάλυση εργασιών χρηστών** (*user task analysis*) χρησιμοποιώντας συνεντεύξεις, δημοσκοπήσεις, παρατηρήσεις, κτλ.
- Μ' αυτόν τρόπο παράγονται σενάρια λειτουργίας και συλλέγονται απαιτήσεις του συστήματος που θα ληφθούν υπόψη στο επόμενες φάσεις της διεργασίας ανάπτυξης του συστήματος.

- Μετά τον προσδιορισμό των απαιτήσεων και τον αυστηρό ορισμό τους μέσω **χρηστο-κεντρικών μετρικών**, σκιαγραφείται ένας αρχικός σχεδιασμός και ένα γρήγορο πρωτότυπο του συστήματος το οποίο αξιολογείται μέσω εκτεταμένης αξιολόγησης και διαφόρων ελέγχων (δοκιμών) χρηστικότητας.
- Ο ΚΖΑΣ είναι εμφανώς χρηστο-κεντρικός (*user-centric*) καθώς σκοπεύει στην ανάπτυξη ενός συστήματος που είναι οικονομικά αποδοτικό αλλά έχει και πολύ υψηλή χρηστικότητα.
- Το γεγονός ότι βοηθά στην ανάπτυξη συστημάτων που είναι εξαιρετικά φιλικά προς τον χρήστη **προλαμβάνει σφάλματα που παράγονται από ανθρώπινη κακή χρήση των διεπαφών** με το σύστημα. Συνεπώς, προάγει την παραγωγικότητα των χρηστών και του οργανισμού.

- Το αστεροειδές μοντέλο κύκλων ζωής:
  - αποτελεί παραλλαγή της μηχανευτικής ευχρηστίας
  - προτάθηκε ως ένα χρηστο-κεντρικό σύνολο από οδηγίες ανάπτυξης λογισμικού που απορρίπτει την αυστηρή βήμα-προς-βήμα προσέγγιση των καταρρακτοειδών μοντέλων ανάπτυξης συστημάτων.
- Η πιο καινοτομική βάση του μοντέλου είναι ότι **κάθε βήμα προς την κατεύθυνση της ανάπτυξης του συστήματος δεν έχει κατ' ανάγκην μια ειδική θέση σε μια συγκεκριμένη διεργασία ανάπτυξης.**

- Το μοντέλο υποθέτει ότι υπάρχουν **πολλά βασικά στάδια της ανάπτυξης** αλλά **μπορούν να εκτελεστούν σε διαφορετικές σειρές και σε διαφορετικά χρονικά πλαίσια** σύμφωνα με τις συγκεκριμένες ανάγκες του έργου, με **δυνατότητα** είτε **πισωγυρίσματος από μια φάση πολλές φορές**, αν απαιτηθεί, είτε **πλήρους υπερπήδησης άλλης φάσης** αν δεν χρειάζεται στο έργο.
- Ο βασικός κανόνας πίσω απ' αυτήν την βάση είναι ότι η **εκτέλεση κάθε φάσης πρέπει να συνοδεύεται από εκτεταμένη αξιολόγηση.**
- Όλες οι φάσεις είναι αλληλο-συσχετισμένες και, κατά την εκτέλεση της διεργασίας ανάπτυξης του συστήματος, είναι δυνατό να πραγματοποιηθεί στροφή προς μία εξ αυτών εφόσον η φάση αυτή αξιολογείται παράλληλα.

- Παρομοίως, **κάθε δραστηριότητα που έχει περατωθεί, ανεξάρτητα της σειράς περάτωσής της, πρέπει να αναλυθεί πλήρως** (π.χ. Με εκτεταμένες δοκιμές και συλλογή δεδομένων γι' αυτήν την δραστηριότητα, μέσω διαφόρων μεθόδων όπως συνεντεύξεις με τους χρήστες ή παρατηρήσεις της χρήσης του συστήματος απ' τους χρήστες εντός του πραγματικού περιβάλλοντος εργασίας τους).



## Κύκλος ζωής κινητών εφαρμογών

- Οι κύκλοι ζωής κινητών εφαρμογών (*mobile apps*) δεν διαφέρουν σημαντικά από τους παραδοσιακούς κύκλους ζωής για κοινές εφαρμογές ή εφαρμογές ιστού.

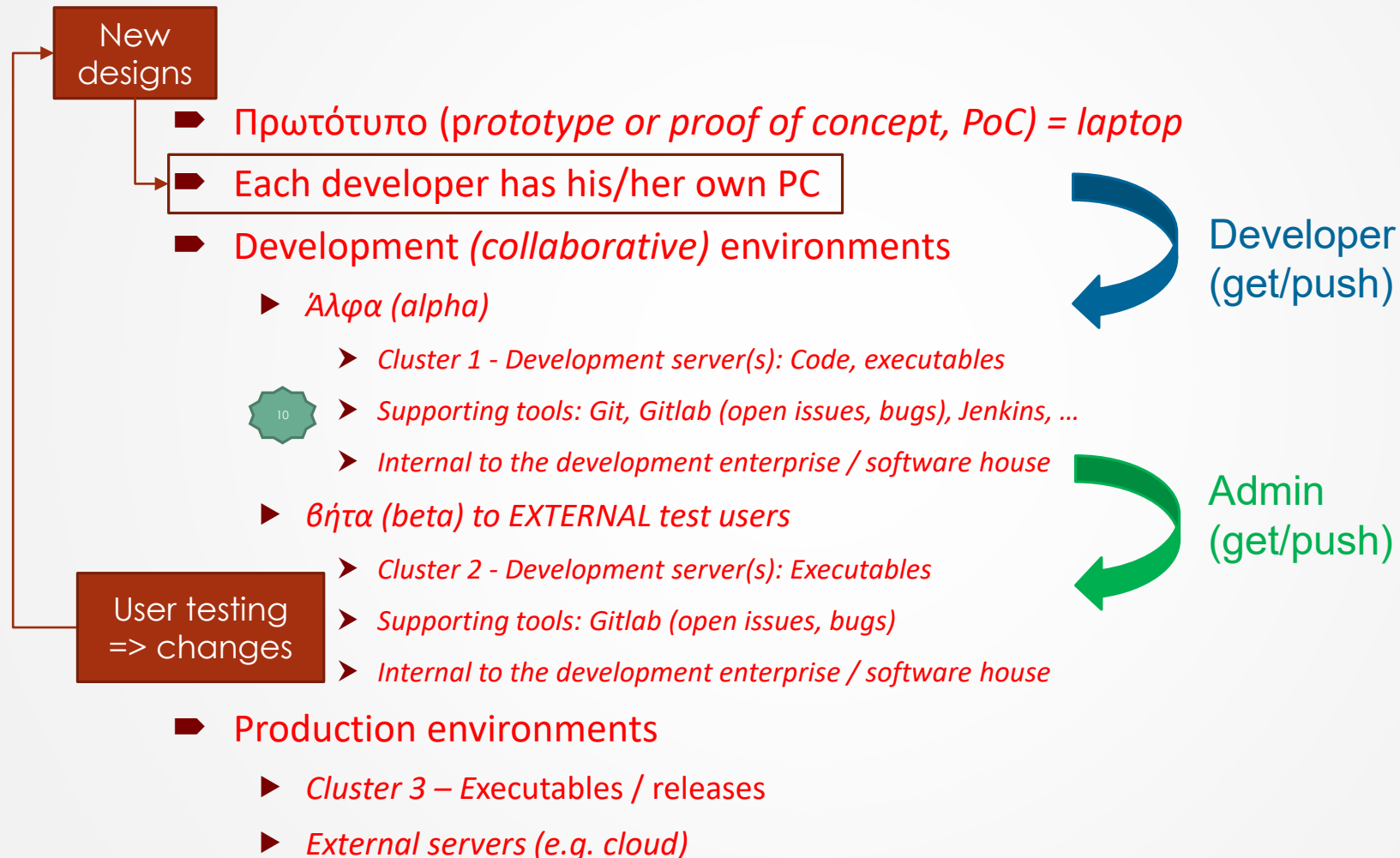


Μια μεθοδολογία κύκλου ζωής για κινητές εφαρμογές αποτελείται από πέντε κύριες φάσεις:

- i. **Σύλληψη (Inception)** – Όλες οι εφαρμογές αρχίζουν με μια ιδέα που συνήθως εκλεπτύνεται προκειμένου να αποτελέσει μια σταθερή βάση για μια εφαρμογή. Συνεπώς, η φάση της σύλληψης αναφέρεται στον ορισμό και στην εκλέπτυνση της ιδέας μιας κινητής εφαρμογής.
- ii. **Σχεδιασμός (Design)** – Με το πέρας της φάσης της “σύλληψης” το βασικό ερώτημα μετατίθεται από το “τι εφαρμογή θα σχεδιαστεί” στο “πως θα σχεδιαστεί η εφαρμογή”. Σ’ αυτό ακριβώς αναφέρεται η φάση του σχεδιασμού. Βασικά, η φάση του σχεδιασμού αποτελείται από τον ορισμό της εμπειρίας χρήστη - ΕΧ της εφαρμογής και από την μετατροπή αυτής της ΕΧ σε κατάλληλο σχεδιασμό διεπαφών χρήστη, συνήθως με τη βοήθεια ενός σχεδιαστή γραφικών.

- iii. **Ανάπτυξη (Development)** – Πρόκειται για τη φάση με την μεγαλύτερη κατανάλωση πόρων που, βασικά, αφορά στην κατασκευή της εφαρμογής. Η φάση της ανάπτυξης αρχίζει συνήθως πολύ νωρίς. Πράγματι, μόλις ωριμάσει η ιδέα κατά την εννοιολογική φάση της σύλληψης/έμπνευσης αναπτύσσεται συχνά ένα πρωτότυπο εργασίας προς επαλήθευση/επικύρωση της λειτουργικότητας και των παραδοχών, ενώ βοηθά στην κατανόηση του εύρους της εργασίας που θα ακολουθήσει.

**iv. Σταθεροποίηση (Stabilization)** – Όταν η ανάπτυξη βρίσκεται σε προχωρημένο στάδιο η ομάδα διασφάλισης ποιότητας αρχίζει συνήθως τις δοκιμές (ελέγχους) της εφαρμογής και τις διορθώσεις των σφαλμάτων. Σε μερικές περιπτώσεις, η εφαρμογή τίθεται σε κατάσταση “*beta*” κατά την οποία παρέχεται η δυνατότητα σε ένα ευρύτερο κοινό να τη χρησιμοποιήσει και να παράσχει σχόλια και παρατηρήσεις για τυχόν αλλαγές. Ουσιαστικά, η σταθεροποίηση είναι η διαδικασία της αποσφαλμάτωσης της εφαρμογής, όχι μόνο από τη λειτουργική άποψη αλλά και από την άποψη της χρηστικότητας και της αποδοτικότητας. Είναι, συχνά καλύτερο να αρχίσει η φάση της σταθεροποίησης πολύ νωρίς κατά τη φάση της ανάπτυξης έτσι ώστε οι διορθώσεις πορείας να συμβούν πριν να γίνουν πολύ ακριβές από άποψης κόστους. Στην πράξη, 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 – έκδοση (*releaseSSSSSS*).**



- v. Deployment (Πραγμάτωση)** – Όταν η εφαρμογή έχει σταθεροποιηθεί είναι έτοιμη για διάθεση προκειμένου να χρησιμοποιηθεί από όσους το επιθυμούν. Εάν η εφαρμογή έχει τεθεί σε κατάσταση “*beta*” και έχει διανεμηθεί ως τέτοια σε κοινότητες χρηστών, η εφαρμογή πραγματώνεται μόνο μετά την ενσωμάτωση των εποικοδομητικών σχολίων και παρατηρήσεων των κοινοτήτων των χρηστών. Υπάρχουν διαφορετικές προσεγγίσεις πραγμάτωσης και διανομής της εφαρμογής που εξαρτώνται, εν πολλοίς, από την πλατφόρμα χρήσης.

- Συχνά, οι παραπάνω φάσεις επικαλύπτονται.
- Μια εφαρμογή μπορεί να σταθεροποιείται ταυτόχρονα με την εισαγωγή νέων χαρακτηριστικών σε μια νέα έκδοσή της.
- Οι παραπάνω φάσεις μπορούν να χρησιμοποιηθούν σε κάθε μεθοδολογία ΚΖΑΣ, όπως οι ευέλικτες (*Agile*), οι σπειροειδείς (*Spiral*) και οι καταρρακτοειδείς (*Waterfall*) μεθοδολογίες.

# Προτυποποίηση

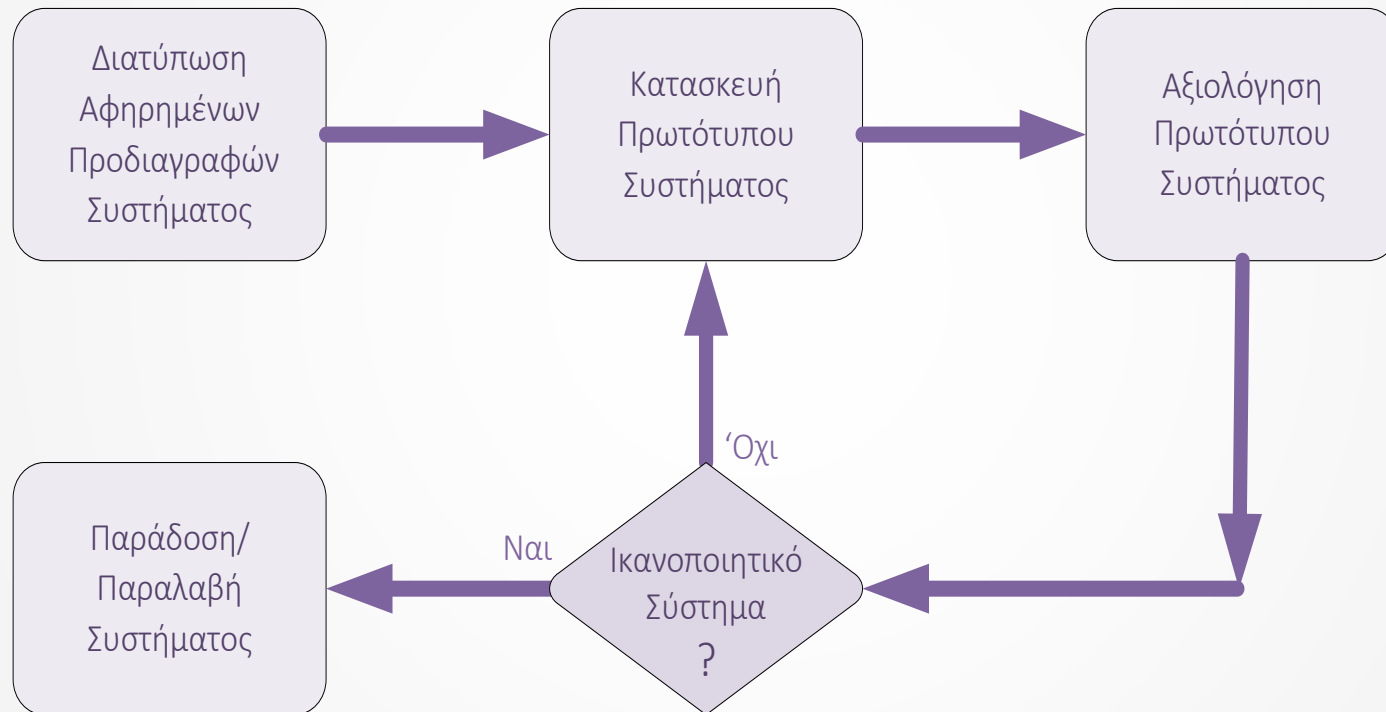
- Το μοντέλο προτυποποίησης (*prototyping model*) είναι ένα επαναληπτικό πλαίσιο που βρίσκεται στο επίκεντρο των πλέον ευέλικτων προσεγγίσεων για την ανάπτυξη συστημάτων ήδη από την δεκαετία του 1980.
- Προς το τέλος της δεκαετίας του 1990 παρατηρήθηκε ότι τα μοντέλα ΚΖΑΣ τα οποία είχαν ενσωματώσει και την προτυποποίηση ως μέσο προσδιορισμού των απαιτήσεων αποδείχθηκαν πιο σφριγηλά, πιο δυναμικά και πιο προσαρμοστικά στις ανάγκες των χρηστών.
- Επειδή αποδείχθηκαν στην πράξη ως μοντέλα μειωμένου κινδύνου και πιο αποτελεσματικά, καταβλήθηκε προσπάθεια να ενταχθεί η προτυποποίηση ως ξεχωριστή μεθοδολογία σε ένα νέο διακριτό και συνεπές πλαίσιο.



- Το μοντέλο προτυποποίησης βασίζεται στην ιδέα της δημιουργίας ολόκληρου ή μέρους ενός συστήματος σε πιλοτική έκδοση που ονομάζεται **πρωτότυπο (prototype)**.
- Η προτυποποίηση μπορεί να θεωρηθεί ως μια διεργασία που είναι:
  - μέρος ενός μεγαλύτερου ΚΖΑΣ ή
  - η κεντρική προσέγγιση που καθορίζει τον ΚΖΑΣ.
- Ο τελικός στόχος είναι να κατασκευαστούν διάφορες εκδόσεις του συστήματος και να εκλεπτυνθούν με συνέπεια (κάτι που διορθώνεται σε μία έκδοση του συστήματος να μην επαναφέρεται στην παλαιά μορφή σε επόμενη έκδοση ή, τουλάχιστον, να μην συμβαίνει αυτό συχνά) **μέχρι να αναπτυχθεί η τελική έκδοση του συστήματος.**

Ουσιαστικά, ένα μοντέλο προτυποποίησης περιλαμβάνει τέσσερα στάδια:

- Αναλύονται και προσδιορίζονται οι απαιτήσεις και ανάγκες των χρηστών.
- Η ομάδα αναπτύσσει ένα πρωτότυπο εργασίας του συστήματος το οποίο πραγματώνεται έτσι ώστε οι χρήστες να μπορούν να το δοκιμάζουν και να παρέχουν ανατροφοδότηση και εμπειρία σε πραγματικό χρόνο.
- Αν κριθεί αναγκαίο να πραγματοποιηθούν βελτιώσεις και αλλαγές, το πρωτότυπο αναθεωρείται και εκλεπτύνεται και ένα νέο πρωτότυπο διατίθεται και πραγματώνεται για έλεγχο. Αυτός ο υπο-κύκλος συνεχίζεται μέχρις ότου το σύστημα καταστεί γενικά αποδεκτό από τους χρήστες και δεν χρειάζεται περαιτέρω σημαντικές αλλαγές και ενημερώσεις.
- Διατίθεται η τελική έκδοση.



- Γενικά, πρόκειται για μια **χρηστο-κεντρική προσέγγιση** δεδομένου ότι η ανατροφοδότηση των χρηστών είναι καθοριστική για την ανάπτυξη διαδοχικών πρωτοτύπων και, τελικά, της τελικής έκδοσης του συστήματος. Το γεγονός αυτό προσδίδει στο έργο πολύ μεγαλύτερη πιθανότητα επιτυχίας μειώνοντας τον κίνδυνο αποτυχίας του
- Στο πλαίσιο του μοντέλου αυτού, δίδεται **ιδιαίτερη έμφαση στην κατασκευή του συστήματος** (το οποίο εξελίσσεται καθώς αναπτύσσεται) και λιγότερη στη δημιουργία τεκμηριωτικού υλικού. Έτσι, το χρονικό πλαίσιο για το έργο της ανάπτυξης του συστήματος είναι πολύ στενότερο από τα αυστηρά μοντέλα, όπως το καταρρακτοειδές.

- Τα μοντέλα προτυποποίησης είναι ασθενή αναφορικά με τον προγραμματισμό της ανάλυσης και του σχεδιασμού του συστήματος.
- Ενώ συλλέγονται και εκτιμώνται οι απαιτήσεις καθώς αναπτύσσεται το σύστημα σε διαδοχικά στάδια, υπάρχει μικρός έλεγχος του κόστους και των δαπανώμενων πόρων που σημαίνει ότι μπορεί να αυξηθεί δραματικά το οικονομικό κόστος του έργου.
- Συνεπώς, η προτυποποίηση είναι κατάλληλη για σχετικά μεγαλύτερα έργα και, ιδιαίτερα, για έργα που είναι χρηστο-κεντρικά από επιλογή ή λόγω της μη καλά ορισμένης φύσης τους.

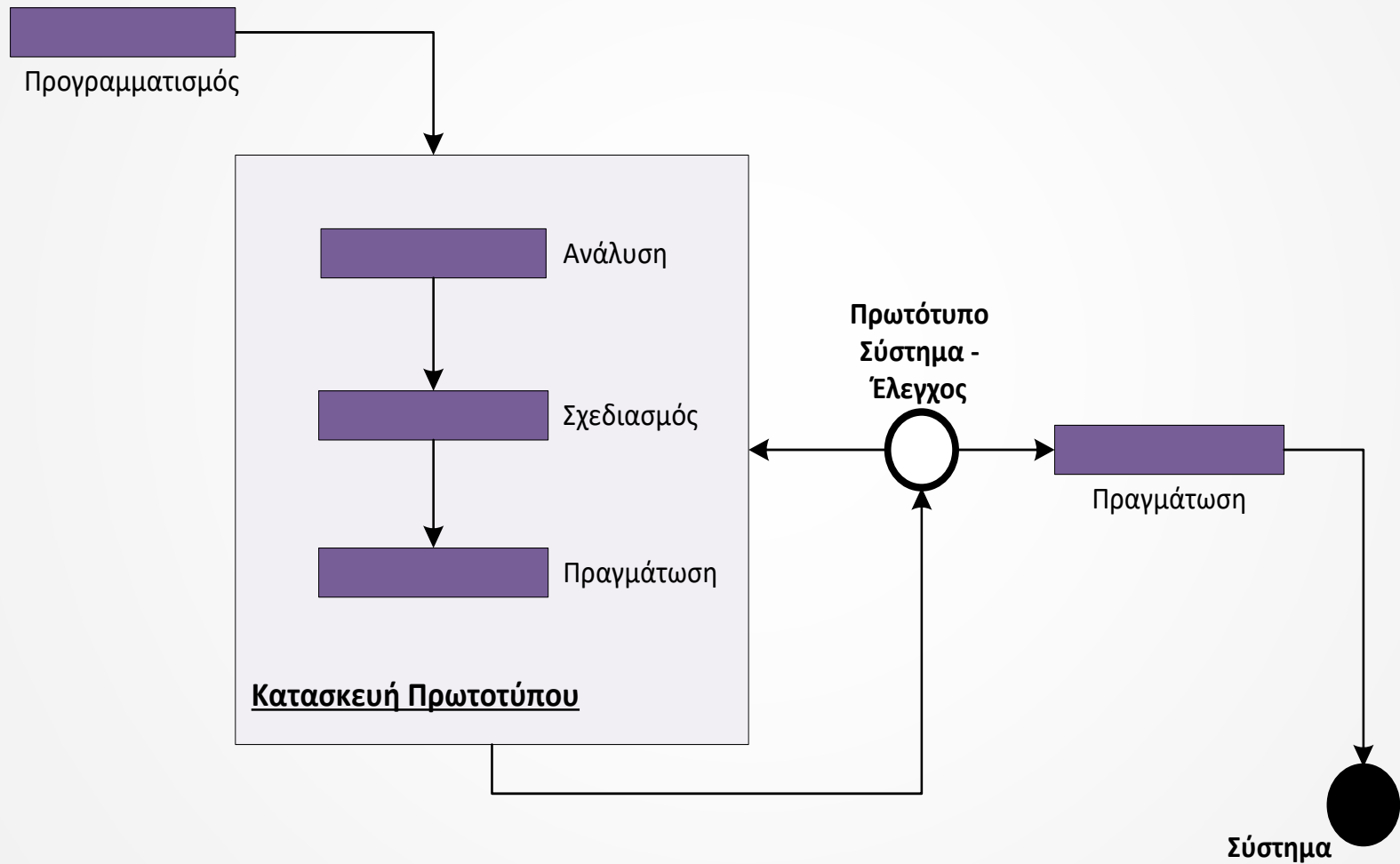
- Υπάρχουν διάφορα είδη προτυποποίησης, σύμφωνα με τις συγκεκριμένες ανάγκες του έργου. Τα είδη αυτά μπορούν να ενταχθούν σε τρεις κατηγορίες:
  - εξέλιξη (*evolution*),
  - διερεύνηση (*exploration*) και
  - πειραματισμός (*experimentation*).
- Οι σχετικές μεθοδολογίες περιγράφονται στις ακόλουθες διαφάνειες.

- Η μεθοδολογία των εξελικτικών πρωτοτύπων (*evolutionary prototyping*) είναι ένα μοντέλο κύκλου ζωής με βάση το οποίο **το ΠΣ αναπτύσσεται εξελικτικά** έτσι ώστε να μπορεί να τροποποιείται εύκολα και γρήγορα, σε ανταπόκριση της ανατροφοδότησης που δέχεται από τους χρήστες και τους εν γένει πελάτες.
- Η εξελικτική προσέγγιση:
  - Περιγράφει την ανάπτυξη ενός συστήματος σε **διαδοχικές εκδόσεις**.
  - Βρίσκεται πολύ κοντά στα επαυξητικά και επαναληπτικά μοντέλα ΚΖΑΣ, αναφορικά με το ότι ο κύριος σκοπός τους είναι να ενσωματωθούν στο σύστημα οι αναπόφευκτες αλλαγές στις απαιτήσεις και τις ανάγκες στο χρονικό διάστημα από την έναρξη μέχρι την ολοκλήρωση του έργου.

- Η μεθοδολογία των εξελικτικών πρωτοτύπων χρησιμοποιείται συνήθως όταν:
  - Οι επιχειρησιακές διεργασίες είναι πιθανό να αλλάξουν καθώς προοδεύει το έργο ή
  - Ο αναθέσας το έργο δεν γνωρίζει πολλά πράγματα περί του συστήματος που θα κατασκευαστεί.
- Οι φάσεις της **ανάλυσης**, του **σχεδιασμού** και της **πραγμάτωσης** εκτελούνται ταυτόχρονα και κάθε εξελικτικός κύκλος καταλήγει σε ένα **πρωτότυπο του συστήματος** που τίθεται στην κρίση του αναθέσαντος το έργο (δια των χρηστών).
- Ο εξελικτικός κύκλος επαναλαμβάνεται διαρκώς με βάση τις παρατηρήσεις των χρηστών μέχρι να ικανοποιηθούν όλες οι απαιτήσεις τους. Το τελευταίο πρωτότυπο με κάποιες μικρές διορθώσεις είναι η τελική εκδοχή του συστήματος.



- Με τη μεθοδολογία των εξελικτικών πρωτοτύπων, μια πρώτη εκδοχή του συστήματος παραδίδεται στους χρήστες πολύ σύντομα προκειμένου να προβούν σε παρατηρήσεις, αν και συνειδητά δεν ικανοποιεί πλήρως τις απαιτήσεις.
- Έτσι, απαιτείται μόνο μια αρχική διαδικασία ανάλυσης και σχεδιασμού του υπό ανάπτυξη συστήματος.
  - ▶ Ο εντοπισμός σημαντικών λειτουργιών του συστήματος που δεν έχουν προσδιοριστεί μπορεί να γίνει περίπου στο μέσον του χρόνου εκτέλεσης του έργου.
- Επομένως είναι πιθανό να **μεταβληθεί η αρχική απόφαση** αναφορικά με το σχεδιασμό του συστήματος και να αρχίσει η όλη διεργασία πάλι από την αρχή.



Κατά τη μεθοδολογία των εξελικτικών πρωτοτύπων, ένα σύστημα αναπτύσσεται σταδιακά από τα πιο ορατά στους χρήστες σημεία του στα πιο αλγοριθμικά και, συνεπώς, πιο αόρατα στους χρήστες.

- Κατ' αρχήν αναπτύσσονται οι πιο ορατές όψεις του συστήματος, παρουσιάζεται αυτή η έκδοση του συστήματος στους εμπλεκόμενους.
- Κατόπιν συνεχίζει να αναπτύσσεται το πρωτότυπο επί τη βάσει τις ανατροφοδότησης που δέχεται από τους χρήστες.
- Σε κάποιο σημείο όλοι οι εμπλεκόμενοι (κατασκευαστές και πελάτες) συμφωνούν ότι το πρωτότυπο είναι ικανοποιητικό.
- Στο σημείο αυτό το πρωτότυπο αυτό θεωρείται ως το τελικό προϊόν.

Κύρια πλεονεκτήματα:

- Δεν διαγράφεται η καταβληθείσα προσπάθεια για την κατασκευή των πρωτοτύπων
- Η κατασκευή του συστήματος επιταχύνεται σε σχέση με το καταρρακτοειδές μοντέλο
- Η συμμετοχή των χρηστών είναι πολύ σημαντική από την αρχή του έργου
- Τα τεχνικά και άλλα προβλήματα ανακαλύπτονται σε πρώιμο στάδιο όταν είναι ευκολότερη η επίλυσή τις, μειώνοντας τον κίνδυνο αποτυχίας του τελικού συστήματος
- Είναι πολύ κατάλληλη για μη καλά δομημένες καταστάσεις όπου υπάρχουν ασαφείς και ασταθείς απαιτήσεις

Κύρια μειονεκτήματα:

- Τα πρωτότυπα εξελίσσονται τόσο γρήγορα και συχνά που δεν είναι αποτελεσματικό να παράγεται σημαντικό τεκμηριωτικό υλικό
- Οι συνεχείς αλλαγές τείνουν να διαβρώνουν τη δομή του πρωτοτύπου συστήματος δυσχεραίνοντας και επιβαρύνοντας την συντήρηση του συστήματος.
- Δεν είναι σαφές πώς θα χρησιμοποιηθούν αποτελεσματικά οι διάφορες δεξιότητες που συνήθως χρησιμοποιούνται σε ομάδες μηχανικής λογισμικού με αυτήν τη μεθοδολογία ανάπτυξης του συστήματος.

## Βασικοί κίνδυνοι:

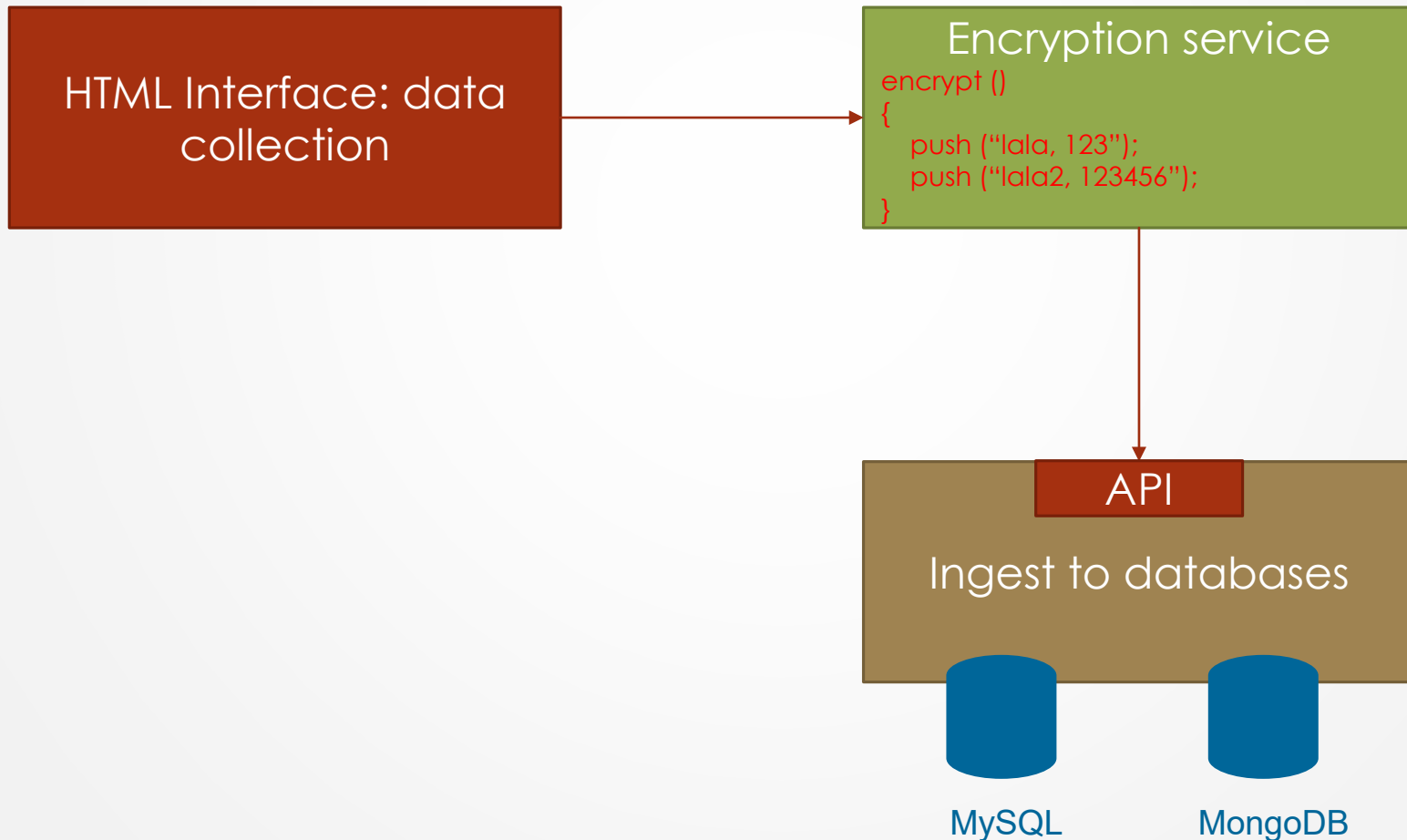
- Υπερβάσεις του χρονοδιαγράμματος και του προϋπολογισμού του έργου,
- Αναποτελεσματική χρήση των πρωτοτύπων,
- Μη ρεαλιστικές προσδοκίες αποδοτικότητας του συστήματος και
- Μη αποτελεσματικός και σύγχρονος σχεδιασμός του.

## Βασικοί παράγοντες επιτυχίας:

- Χρήση κατασκευαστών που διαθέτουν γνώση και εμπειρία,
- Καλή διαχείριση του χρονοδιαγράμματος και του προϋπολογισμού
- Καλή διαχείριση της δραστηριότητας της προτυποποίησης

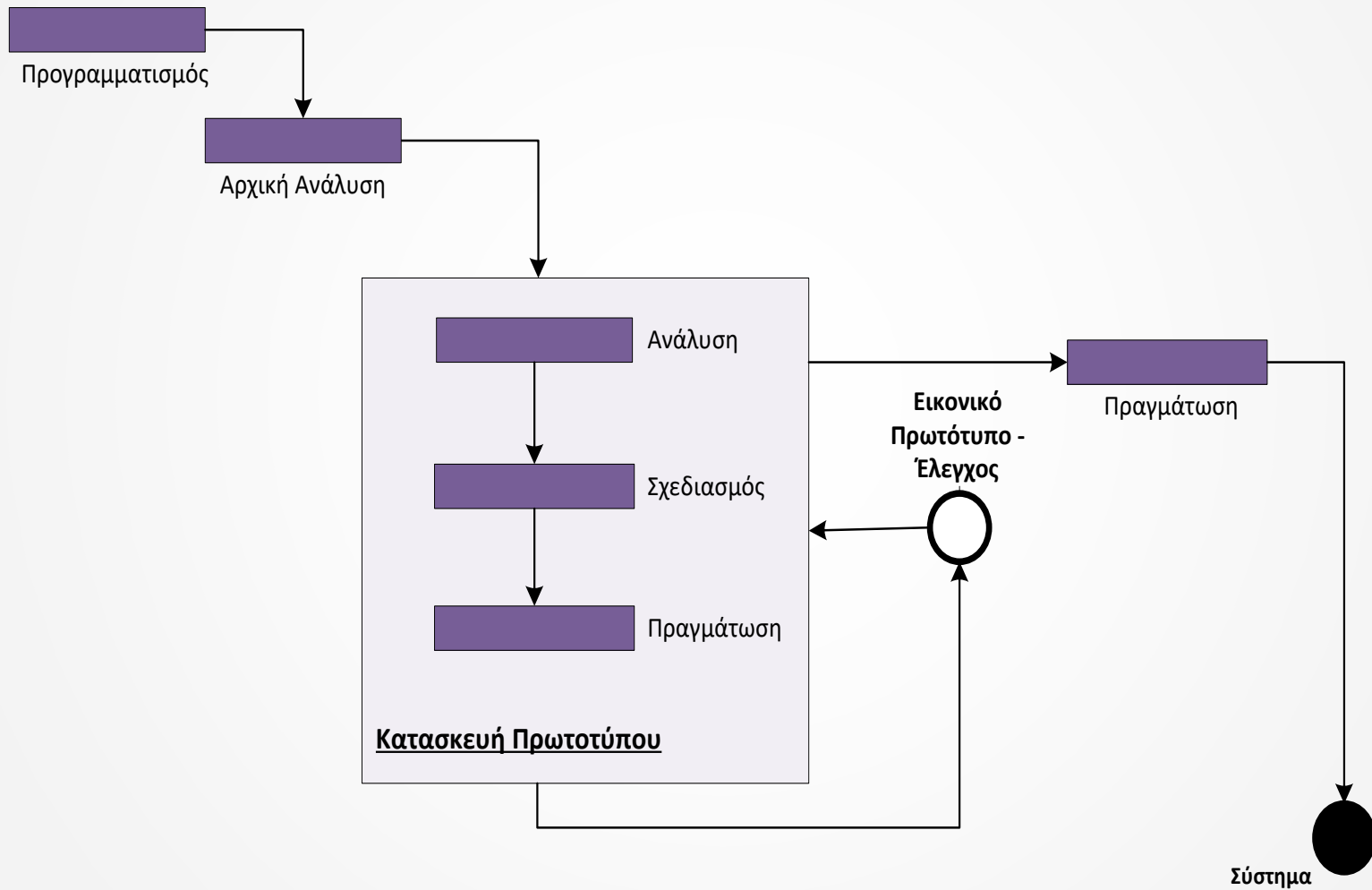
Η μεθοδολογία απορριπτών πρωτοτύπων (*throwaway prototyping*) θεωρείται ως **διερευνητική προτυποποίηση**.

- Προβλέπει την κατασκευή πρωτοτύπων (όπως η μέθοδος εξελικτικών πρωτοτύπων ).
- Χρησιμοποιεί το πρωτότυπο μόνο ως μέσο παρουσίασης του συστήματος στους χρήστες προς υποβολή παρατηρήσεων και τίποτε άλλο.
- Βασίζεται στη λογική ότι οι απαιτήσεις πρέπει να ανιχνεύονται πλήρως σε κάθε επανάληψη. Έτσι, εκδόσεις του συστήματος παραδίδονται γρήγορα, ανιχνεύονται οι ανάγκες και οι απαιτήσεις κάθε έκδοσης και χρησιμοποιούνται προκειμένου να τελειοποιηθεί καταλλήλως η επόμενη έκδοση.

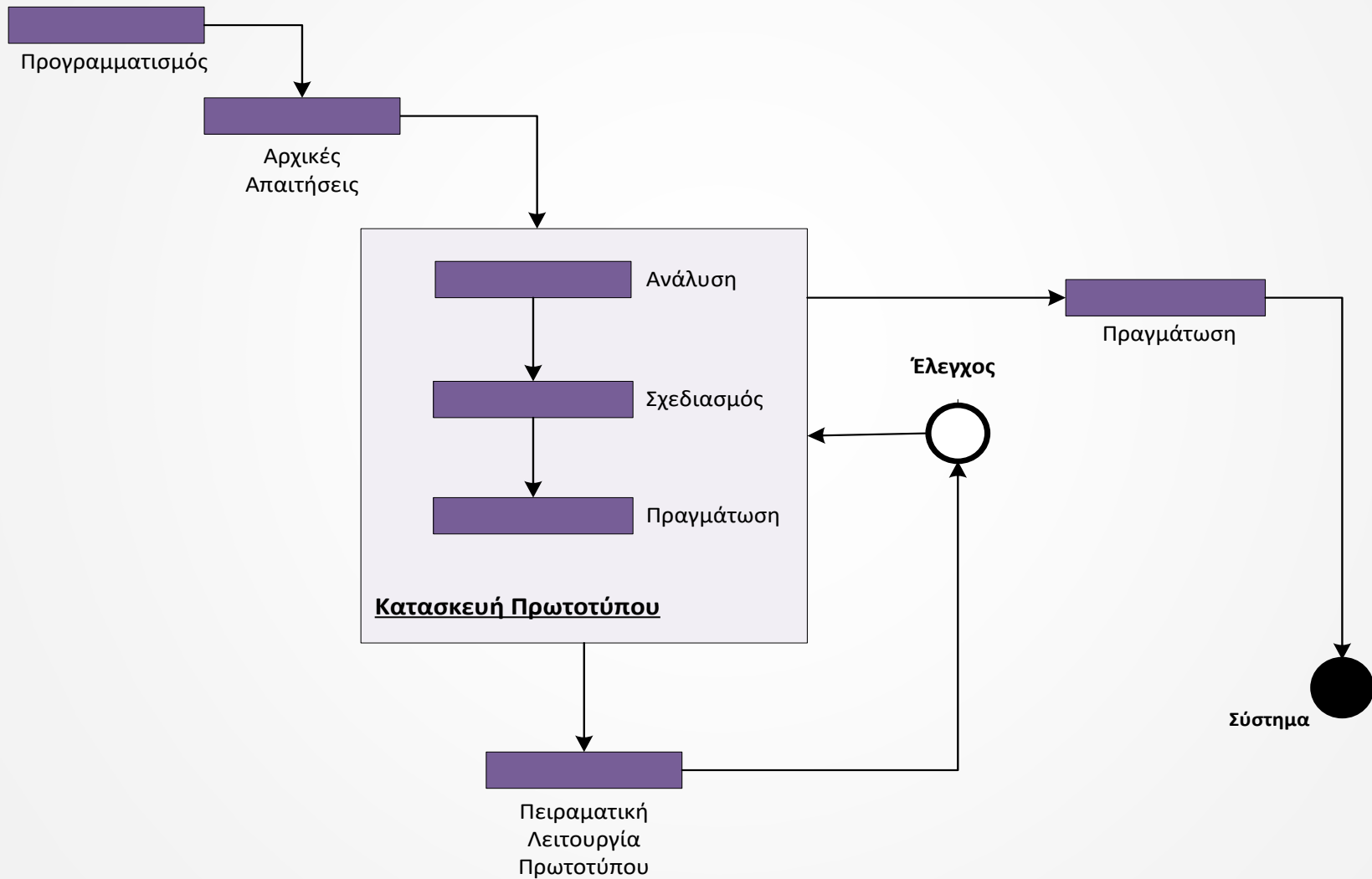




- Οι ανάγκες εκμαιεύονται καθώς το σύστημα χρησιμοποιείται και ελέγχεται από τους χρήστες.
- Στόχος του πρωτοτύπου είναι να βοηθηθούν οι χρήστες στην απόκτηση μιας οπτικής αναπαράστασης του υπό κατασκευή συστήματος.
- Με βάση τις παρατηρήσεις των χρηστών, κατασκευάζονται τα επόμενα πρωτότυπα μέχρι να οπτικοποιηθεί το πλήρες πραγματικό σύστημα.
- Το επόμενο βήμα είναι η **πραγμάτωση** (πilotική ή/και παραγωγική) του πραγματικού συστήματος.
- Το **απορριπτέο πρωτότυπο** ονομάζεται επίσης **εικονικό πρωτότυπο** ή **μακέτα του συστήματος** (*dummy (mock-up) prototype*).



- Η μεθοδολογία **πειραματικών πρωτοτύπων** (experimental prototyping):
  - Προβλέπει την κατασκευή πρωτοτύπων (όπως στη μεθοδολογία των εξελικτικών ή των απορριπτέων πρωτοτύπων)
  - Προσομοιάζει περισσότερο προς τη μεθοδολογία των απορριπτέων πρωτοτύπων στο ότι προβλέπει την απόρριψη του πειραματικού συστήματος.
  - Σε αντίθεση με τις άλλες μεθοδολογίες προτυποποίησης, πρεσβεύει ότι η λύση που ικανοποιεί τις ανάγκες των χρηστών πρέπει πρώτα να προτείνεται (αναπτύσσεται) και κατόπιν να αξιολογείται μέσω μιας πειραματικής χρήσης.



- Στόχος του πειραματικού πρωτοτύπου είναι
  - να βοηθηθούν οι χρήστες στην απόκτηση μιας οπτικής αναπαράστασης του υπό κατασκευή συστήματος
  - να πειραματιστούν οι χρήστες με το υπό κατασκευή σύστημα σε προσομοιώσεις των επιχειρησιακών διεργασιών του πραγματικού κόσμου.
- Με βάση τις παρατηρήσεις των χρηστών, κατασκευάζονται τα επόμενα πρωτότυπα μέχρι να οπτικοποιηθεί το πλήρες πραγματικό σύστημα.

# Ταχεία ανάπτυξη συστημάτων

- 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*).



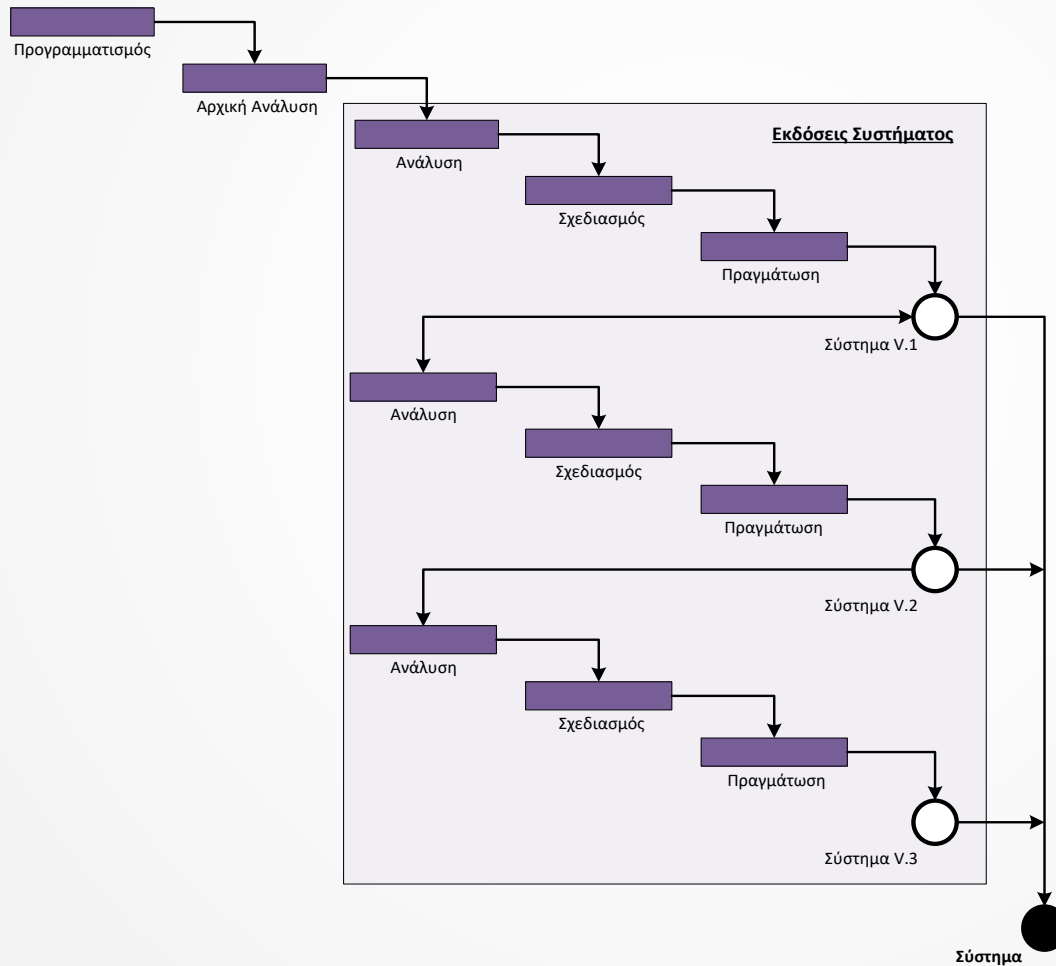
- Η γενική μεθοδολογία ΤΑΣ εισήγαγε τη χρήση προηγμένων εργαλείων ανάπτυξης συστημάτων, όπως:
  - οι γεννήτριες κώδικα και
  - οι οπτικές γλώσσες προγραμματισμού.
- Με τη χρήση αυτών των εργαλείων:
  - επιταχύνεται η διαδικασία ανάπτυξης του συστήματος και,
  - σε κάποιο βαθμό, παράγεται υψηλότερη ποιότητα 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 συστήματος προς όφελος της ποιότητάς του και της αποδοχής του από τους χρήστες.

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.
- 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 απόψεων για το υπό ανάπτυξη σύστημα.

Συγκριτικά με την καταρρακτοειδή προσέγγιση,

- Η ΚΑΕ φαίνεται ότι επιφέρει την ανάπτυξη του συστήματος σε πιο σύντομο χρόνο και επιτυγχάνει μεγαλύτερο βαθμό ικανοποίησης των χρηστών, με τα δύο αυτά χαρακτηριστικά να προέρχονται από την διαρκή συμμετοχή των πελατών στην ανάπτυξη του συστήματος.
- Οι πελάτες συμμετέχουν ελάχιστα στη διεργασία ανάπτυξης του συστήματος και μόνο μέσω μιας πιθανής διαδικασίας συνεντεύξεων στην αρχή του έργου προκειμένου να προσδιοριστούν οι απαιτήσεις και του ελέγχου των παραδοτέων στο τέλος κάθε φάσης.



- Η ανάπτυξη συστημάτων από τελικούς χρήστες απλώς σημαίνει ότι οι χρήστες προβαίνουν μόνοι τους στην εργασία ανάπτυξης όλου ή μέρους του συστήματος.
- Πρόκειται για μια προσέγγιση που αποβλέπει και επιτυγχάνει σε σημαντικό βαθμό την ταχεία ανάπτυξη του συστήματος (της εφαρμογής) και που προσομοιάζει σε σημαντικό βαθμό με τη μεθοδολογία πρωτοτύπων εκτός του ότι οι τελικοί χρήστες (αντί των τεχνικών πληροφορικής) είναι αυτοί που δημιουργούν και τροποποιούν τα πρωτότυπα.

## 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 ωρών ή 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.

- 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).

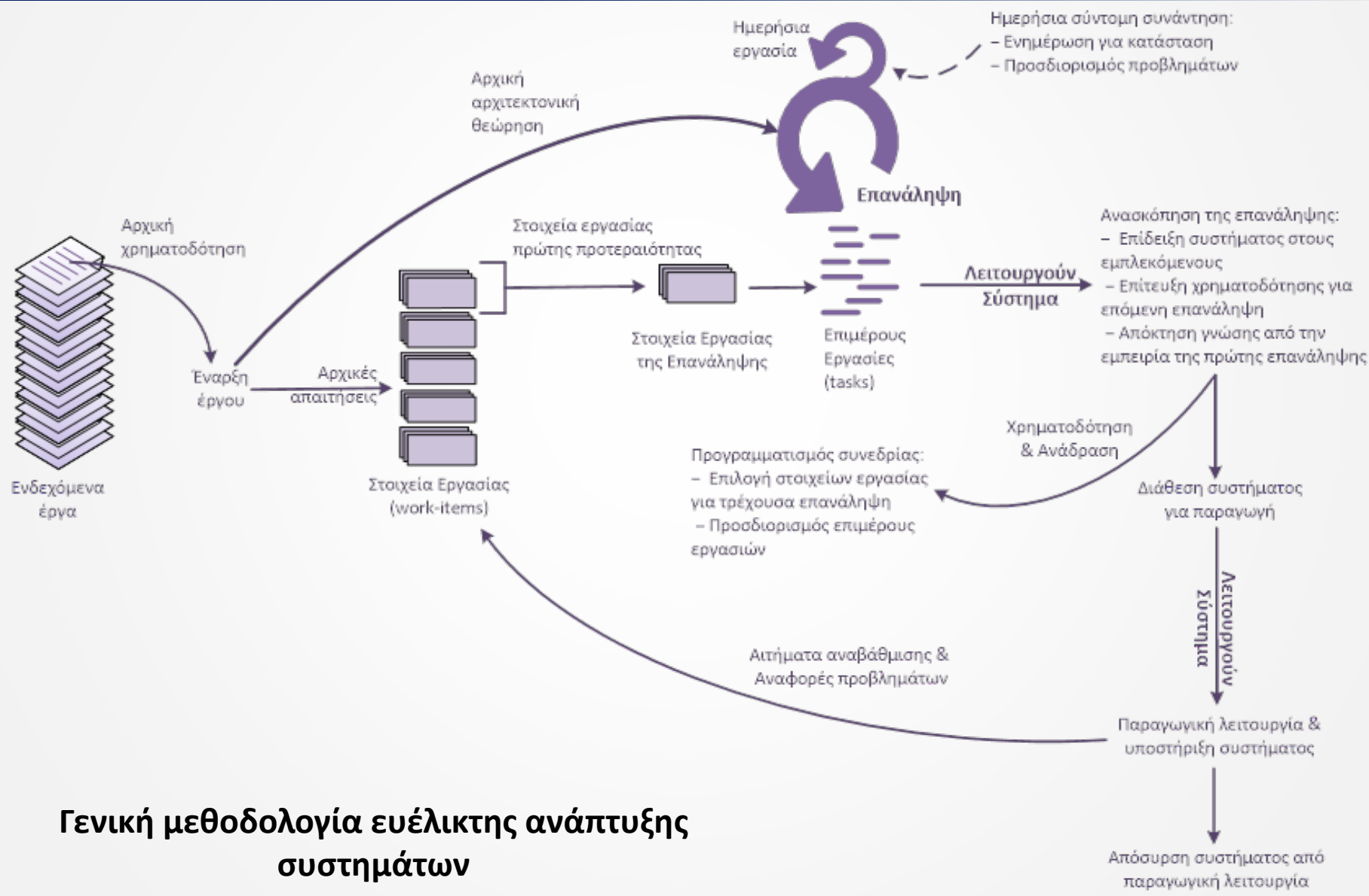
- Οι ευέλικτες μεθοδολογίες αποτελούν μια απόκριση στις μεγάλες και γρήγορες αλλαγές που συμβαίνουν στα σύγχρονα επιχειρηματικά και πληροφοριακά περιβάλλοντα.
- Οι ευέλικτες μεθοδολογίες προωθούν μια πειθαρχημένη διεργασία διοίκησης έργων ανάπτυξης συστημάτων η οποία ενθαρρύνει:
  - Τη διαρκή επιθεώρηση 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).
- Από τις μεθοδολογίες αυτές, που είναι οι πιο δημοφιλείς, η DSDM είναι ίσως η πιο πλήρης.
- Οι μεθοδολογίες Scrum και XP είναι πιο εύκολα εφαρμόσιμες και είναι συμπληρωματικές καθώς αντιμετωπίζουν διαφορετικές όψεις των έργων ανάπτυξης συστημάτων ενώ εδράζονται σε παρόμοιες έννοιες.

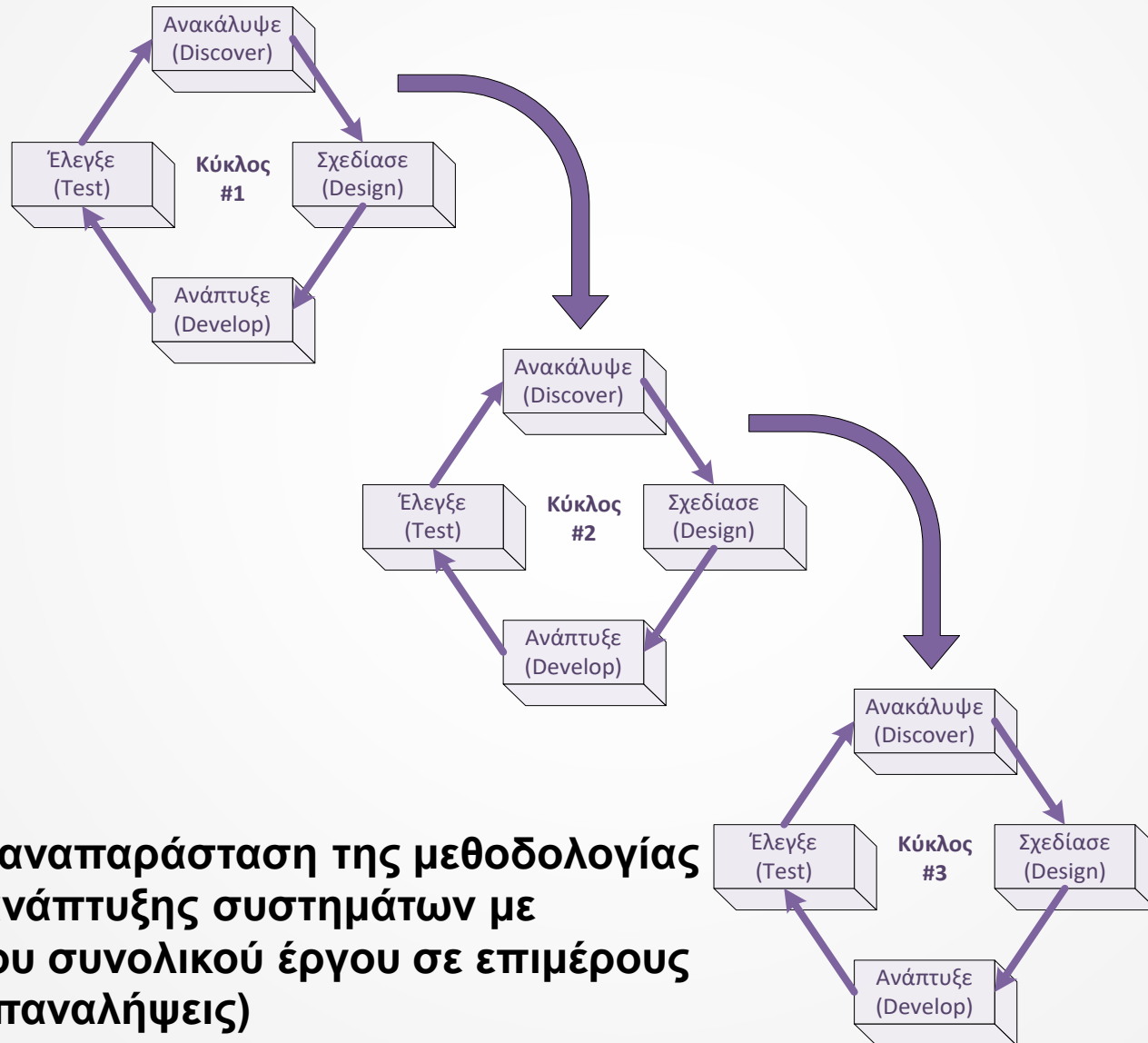
Οι περισσότερες μεθοδολογίες ευέλικτης ανάπτυξης συστημάτων αποσκοπούν στην ελαχιστοποίηση του κινδύνου αποτυχίας αναφορικά με την ανάπτυξη και αποδοχή του λογισμικού.

- 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.

- Ουσιαστικά, οι αρχές αυτές είναι χαρακτηριστικά τα οποία είναι κοινά σε όλες τις μεθοδολογίες ευέλικτης ανάπτυξης συστημάτων και τις διαφοροποιούν θεμελιακά από την παραδοσιακή καταρρακτοειδή μεθοδολογία.
- Οι αρχές αυτές χρησιμοποιούνται ως οδηγοί για την εφαρμογή της ευέλικτης ανάπτυξης συστημάτων σε περιβάλλοντα του πραγματικού κόσμου.

- 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.
  - 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.
- Περατώνεται 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.

- Είναι ουσιαστική και σημαντική η συνεργασία και συνέργεια μεταξύ των εμπλεκομένων στο έργο και αυτών που έχουν συμφέρον από την εκτέλεσή του.

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.

*Ίσως το πιο σημαντικό από τα άνω σημεία είναι το σημείο (iv), ότι: η πιο αποτελεσματική και αποδοτική μέθοδος διακίνησης πληροφοριών προς και εντός μιας ομάδας ανάπτυξης συστημάτων είναι η συνομιλία πρόσωπο-με-πρόσωπο.*

- Τα τελευταία χρόνια παρατηρείται μια αυξητική τάση επιτυχών αναπτύξεων συστημάτων κατόπιν εφαρμογής ευέλικτων μεθοδολογιών.
- Σε αντίστοιχες περιγραφές αναφέρεται ότι οι οργανισμοί-χρήστες των ευέλικτων μεθοδολογιών έχουν βελτιώσει σημαντικά την αποτελεσματικότητα και αποδοτικότητα των ομάδων εκτέλεσης έργων ανάπτυξης ΠΣ και την επιτυχία των έργων αυτών.
- Το γεγονός αυτό έχει προκαλέσει την ευρεία υιοθέτηση ευέλικτων μεθοδολογιών από οργανισμούς διαφόρων κλάδων δραστηριότητας του δημόσιου και ιδιωτικού τομέα.

- Η ευέλικτη ανάπτυξη συστημάτων δεν είναι πανάκεια για όλα τα προβλήματα ανάπτυξης συστημάτων.
- Το πραγματικό πλεονέκτημα αποκτάται από τη γνώση διαφόρων ΚΖΑΣ και ευέλικτων μεθοδολογιών και την επιλογή ενός μίγματος από τις καλύτερες προσεγγίσεις που θεωρούνται οι πλέον κατάλληλες για κάθε συγκεκριμένη προβληματική κατάσταση.
- Για να γίνει αυτό αξιόπιστα και με επιτυχία απαιτείται σημαντική γνώση, ικανότητα, δεξιότητες και εμπειρία αναφορικά με την ανάπτυξη συστημάτων και την επιτυχία των αντιστοίχων έργων.
- Στα έργα ανάπτυξης συστημάτων χρησιμοποιώντας ευέλικτες μεθόδους, η διοίκηση έργων λαμβάνει μια διαφορετική μορφή που βασίζεται πολύ περισσότερο στις ικανότητες (δεξιότητες) του διευθυντή του έργου για επικοινωνία, διευκόλυνση και συντονισμό, δίνοντας λιγότερη έμφαση στον προγραμματισμό και στον έλεγχο.

- Η ευέλικτη ανάπτυξη συστημάτων συνιστά ένα εννοιολογικό πλαίσιο για την ανάληψη έργων ανάπτυξης του λογισμικού πληροφοριακών συστημάτων.
- Στη συνέχεια δίδεται μια σύντομη περιγραφή των πιο γνωστών και δημοφιλών μεθοδολογιών ευέλικτης ανάπτυξης συστημάτων
- Κάθε μία από αυτές τις μεθοδολογίες βασίζεται σε μία φιλοσοφική θεώρηση με ιδιαίτερα χαρακτηριστικά.
- Οι φιλοσοφικές θεωρήσεις όλων των ευέλικτων μεθοδολογιών είναι εξαιρετικά παρεμφερείς.

- 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 χρήση της λέξης “*crystal*” αναφέρεται στις διάφορες όψεις ενός πολύτιμου λίθου – κάθε μια από τις οποίες αναπαριστά μια διαφορετική όψη του υποκείμενου πυρήνα του πολύτιμου λίθου.
- Ο υποκείμενος πυρήνας αναπαριστά αξίες και αρχές ενώ κάθε όψη αναπαριστά ένα συγκεκριμένο σύνολο στοιχείων όπως τεχνικές, ρόλοι, εργαλεία και πρότυπα που είναι ορατά στον έξω κόσμο.
- Κατά τα λοιπά, η μεθοδολογία των κρυστάλλινων μεθόδων εκτελείται κατά τον συγκεκριμένο τρόπο εκτέλεσης των ευέλικτων μεθόδων που αναφέρθηκε παραπάνω.

- 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.



- 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 εμπλεκομένων στην ανάπτυξη του συστήματος.
- ➡ Κατά τα λοιπά, η μεθοδολογία DSD εκτελείται κατά τον συγκεκριμένο τρόπο εκτέλεσης των ευέλικτων μεθόδων που αναφέρθηκε παραπάνω.

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”**
- Δίδει ιδιαίτερη έμφαση στη δραστική βελτίωση της παραγωγικότητας των ομάδων ανάπτυξης συστημάτων που είχαν καταστεί δυσλειτουργικές με τη χρήση ιδιαιτέρως δύσκαμπτων και γραφειοκρατικών διεργασιολογιών.

- Η μεθοδολογία Scrum υιοθετεί μια εμπειρική προσέγγιση εστιάζοντας στην ανάπτυξη συστημάτων από ομάδες και στη μεγιστοποίηση των ικανοτήτων των ομάδων να ανταποκριθούν κατά ευέλικτο τρόπο στις όποιες προκλήσεις παρουσιαστούν.
- 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).

## 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:
  - Διοικεί τον συνωστισμό,
  - Έχει ως βασική αρμοδιότητα την εξάλειψη των ενδεχόμενων εμποδίων προκειμένου η ομάδα να παράξει και παραδώσει τα αποτελέσματα της τρέχουσας επανάληψης,

- Δεν κατέχει τον ρόλο του αρχηγού της ομάδας αλλά δρα ως ανάχωμα μεταξύ της ομάδας ανάπτυξης του συστήματος και κάθε πιθανής αποσταθεροποιητικής επιρροής.
- ➡ Κατά τα λοιπά, η μεθοδολογία Scrum εκτελείται κατά τον συγκεκριμένο τρόπο εκτέλεσης των ευέλικτων μεθόδων που αναφέρθηκε παραπάνω.

- 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”.***

- Ευθυγραμμιζόμενη με τις αρχές της ευέλικτης ανάπτυξης συστημάτων, η μεθοδολογία ΧΡ προσβλέπει στην ανάπτυξη ενός λειτουργικού συστήματος που θα ικανοποιεί τις απαιτήσεις των χρηστών και όχι στην απλή τήρηση της σύμβασης μεταξύ των δύο μερών.
- Οι κατασκευαστές ενθαρρύνονται να ελέγχουν κάθε λογικό τμήμα εκάστης εφαρμογής μόλις αυτό περατωθεί και όχι στο τελικό στάδιο της ανάπτυξης όλης της εφαρμογής και/ή του όλου συστήματος.

- 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.

- Η μεθοδολογία ΧΡ είναι, ίσως, η περισσότερο ριζοσπαστική ευέλικτη μεθοδολογία ανάπτυξης συστημάτων που αντιμετωπίζει τις φάσεις της ανάλυσης, της κατασκευής και των ελέγχων (δοκιμών) με καινοτόμες προσεγγίσεις που κάνουν σημαντική διαφορά στην ποιότητα του τελικού συστήματος (προϊόντος).

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
- 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 ΧΡ 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 κατασκευή απλών προγραμμάτων (απλός κώδικας) θεωρείται καλή πρακτική για αύξηση της πιθανότητας απρόσκοπτης λειτουργίας τους. Έτσι, αποφεύγεται η εξαρχής συγγραφή πολύπλοκων προγραμμάτων ή ξαναγράφονται σε απλούστερη μορφή τα προγράμματα που αρχικά γράφηκαν απλά αλλά εξελίχθηκαν σε πολύπλοκα.



- Η διαρκής ανασκόπηση των προγραμμάτων θεωρείται καλή πρακτική. Έτσι, οι προγραμματιστές ΧΡ 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 ΧΡ 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.
- Έχει αποδειχθεί στην πράξη ότι η μεθοδολογία XP έχει λειτουργήσει επιτυχώς με μεγαλύτερες ομάδες πολλών κατασκευαστών.

Τα κύρια πλεονεκτήματα της μεθοδολογίας XP είναι:

- Η σημαντική μείωση του κινδύνου ελλιπούς κατανόησης και ελλιπούς ενσωμάτωσης στο σύστημα των πραγματικών απαιτήσεων, μέσω της επιβαλλόμενης από τη μεθοδολογία στενής συνεργασίας των κατασκευαστών με τους χρήστες.
- Η σημαντική μείωση του κόστους ανάπτυξης του συστήματος δεδομένου ότι δεν παράγεται τεκμηριωτικό υλικό αλλά όλη η διαθέσιμη ανθρωπο-προσπάθεια κατευθύνεται στην ανάπτυξη και στις δοκιμές του συστήματος.
- Η καταλληλότητά της για την ανάπτυξη μικρού εύρους συστήματα, που απαιτούν μικρή ομάδα προγραμματιστών αλλά και μεγαλύτερου εύρους συστήματα τα οποία, όμως, διασπώνται σε μικρότερα ώστε και πάλι να δημιουργούνται μικρές ομάδες προγραμματιστών.

Το κύριο μειονέκτημα της μεθοδολογίας XP είναι:

- Η παραγωγή ως μοναδικού παραδοτέου το αναπτυχθέν σύστημα, δηλαδή η μη παραγωγή συνοδευτικού τεκμηριωτικού υλικού για τις φάσεις της ανάλυσης και του σχεδιασμού του συστήματος, γεγονός το οποίο μπορεί να έχει αρνητικές επιπτώσεις στη συντήρησή του.
- Αυτό το μειονέκτημα είναι πολύ σημαντικό σε μεγάλου εύρους και μεγάλης πολυπλοκότητας συστήματα καθιστώντας τη μεθοδολογία αυτή, και όλες όσες δεν παράγουν τεκμηριωτικό υλικό, σχεδόν ανεφάρμοστες.

- Μια νέα έννοια που χρησιμοποιείται στη μεθοδολογία ακραίου προγραμματισμού είναι ο **προγραμματισμός κατά ζεύγη**, όπου δύο προγραμματιστές εργάζονται συνεχώς μαζί.
- Ο ένας παίζει το ρόλο του προϊσταμένου προγραμματιστή και ο άλλος έχει την αρμοδιότητα διενέργειας των ελέγχων (δοκιμών), αλλά οι εργασίες τους μπορεί να επικαλύπτονται.

- Ο προγραμματισμός κατά ζεύγη θεωρείται από πολλούς ως αναποτελεσματική χρήση πόρων.
- Ο δεύτερος προγραμματιστής είναι συχνά λιγότερο έμπειρος και μπορεί να επιβραδύνει την απόδοση ενός έμπειρου προγραμματιστή.
- Επιπλέον, μπορεί να είναι πιο αποδοτικό να οριστεί ένας προγραμματιστής ο οποίος θα ελέγχει μεγάλα τμήματα κώδικα αντί να ορίζονται πολλοί προγραμματιστές για το έλεγχο πολλών μικρότερων τμημάτων.