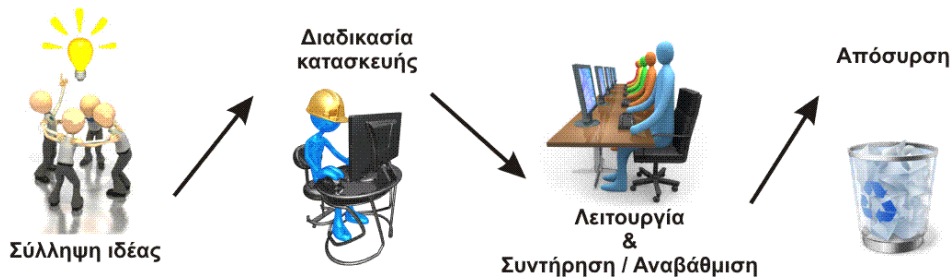


## 1. Κύκλος ζωής λογισμικού

Ο κύκλος ζωής λογισμικού είναι οι φάσεις (τα στάδια) από τις οποίες διέρχεται μία εφαρμογή λογισμικού, από την σύλληψη της ιδέας, τη διαδικασία κατασκευής / ανάπτυξης, τη λειτουργία & συντήρηση / αναβάθμιση, μέχρι την απόσυρσή της.



### 1.1. Φάσεις διαδικασίας κατασκευής (ανάπτυξης) λογισμικού

Η διαδικασία ανάπτυξης λογισμικού δεν είναι μία μονολιθική διαδικασία κατά την οποία με μόνο μία φάση δημιουργούνται ως δια μαγείας οι εφαρμογές. Είναι μία διαδικασία στην οποία διακρίνουμε ξεχωριστές φάσεις, καθεμιά από τις οποίες παίζει το δικό της σημαντικό ρόλο, εμφανίζει τις δικές της ιδιαιτερότητες και απαιτεί τη δική της μεθοδολογία υλοποίησης. Οι φάσεις αυτές είναι οι εξής:

- Ανάλυση του προβλήματος.
- Σχεδίαση της λύσης.
- Υλοποίηση των προγραμμάτων.
- Έλεγχος συμμόρφωσης στις προδιαγραφές και διόρθωση λαθών.
- Λειτουργία και Υποστήριξη.

#### Ανάλυση του προβλήματος

Πρωταρχικός στόχος της ανάλυσης είναι να διαπιστωθεί κατά πόσον το πρόβλημα, που ζητείται να λυθεί, είναι επιλύσιμο με υπολογιστή - **εφικτότητα (feasibility)** υλοποίησης. Στη φάση της ανάλυσης περιγράφονται οι απαιτήσεις της εφαρμογής, επισημαίνονται οι ιδιαιτερότητες, απομακρύνονται οι επουσιώδεις λεπτομέρειες που περιπλέκουν τη λύση, σκιαγραφείται ο τρόπος λύσης, υποδεικνύονται τα σημεία στα οποία απαιτείται ιδιαίτερη προσοχή και περιγράφονται οι συνθήκες κάτω από τις οποίες είναι δυνατή η υλοποίηση. Ακόμα αναλύονται στοιχεία που αφορούν το κόστος, τρόποι διανομής του τελικού προϊόντος στην αγορά κ.ά.

Θεωρείται η πιο σημαντική φάση μιας και θέτει τα θεμέλια για όλες τις επόμενες φάσεις. Αν το έργο είναι μεγάλο πραγματοποιείται από εξειδικευμένους **αναλυτές συστημάτων**. Αν το έργο είναι μικρό, τη δουλειά μπορεί να τη κάνει και ένας έμπειρος προγραμματιστής.

Το **προϊόν** της φάσης αυτής είναι μία κατανοητή, πλήρης και σαφής περιγραφή χαρακτηριστικών και συμπεριφοράς της εφαρμογής ή αλλιώς **προδιαγραφές** της εφαρμογής.

### Σχεδίαση της λύσης

Η φάση της σχεδίασης και ανάπτυξης της λύσης μπορεί να χαρακτηριστεί ως η κατ' εξοχήν δημιουργική φάση. Η φάση αυτή απαιτεί τεχνικές γνώσεις σε θέματα υπολογιστικών συστημάτων και γνώσεις πάνω στον τομέα του προβλήματος.

Στη φάση αυτή δουλεύουν οι **σχεδιαστές** οι οποίοι σε μεγάλα έργα πρέπει να σχεδιάσουν όχι μόνο τη λύση του προβλήματος αλλά και τον τρόπο με τον οποίο θα δουλέψουν οι ομάδες εργασίας, να κάνουν χρονοπρογραμματισμό των επιμέρους εργασιών, κατανομή των πόρων και οικονομικό προγραμματισμό.

Σ' αυτή τη φάση το πρόβλημα χωρίζεται σε επιμέρους προβλήματα, που είναι δυνατόν να επιλυθούν πιο εύκολα και καθορίζεται της μεταξύ τους συσχέτισης. Δημιουργείται ένα **πλάνο λύσης**. Στο πλάνο λύσης για κάθε πρόβλημα ή επινοείται, αν δεν υπάρχει, ή επιλέγεται ένας από τους γνωστούς **αλγόριθμους** που το επιλύει. Επίσης, οργανώνονται τα δεδομένα σε **δομές δεδομένων** και σχεδιάζονται οι **φόρμες – διεπαφές** της εφαρμογής.

Το **προϊόν** της φάσης αυτής είναι το **πλήρες σχέδιο υλοποίησης** της εφαρμογής.

### Υλοποίηση των προγραμμάτων

Στην φάση αυτή, γίνεται η δημιουργία και η σχεδίαση των φορμών επικοινωνίας χρήστη – υπολογιστή (διεπαφή), η διάταξή τους στην οθόνη και η **κωδικοποίηση** των αλγορίθμων και των δομών δεδομένων σε προγράμματα σύμφωνα με τους κανόνες μιας γλώσσας προγραμματισμού. Ο στόχος δεν είναι να δημιουργηθεί μόνο λογισμικό το οποίο να πραγματοποιεί αυτά που έχουν προδιαγραφεί, αλλά λογισμικό που να μπορεί εύκολα να επεκταθεί και να συντηρηθεί, λογισμικό που να είναι λειτουργικό και φιλικό από την πλευρά του χρήστη, λογισμικό που να μπορούν εύκολα να βρεθούν τα πιθανά σφάλματα. Τα παραπάνω περιγράφουν ένα **ποιοτικό λογισμικό**.

Στη φάση αυτή εργάζονται κυρίως οι **προγραμματιστές** οι οποίοι όμως μπορούν να συνεργαστούν και με άλλες ομάδες εργασίας όπως για παράδειγμα γραφίστες.

Το **προϊόν** της φάσης αυτής είναι η **εφαρμογή**.

### Έλεγχος προδιαγραφών και διόρθωση λαθών

Η φάση αυτή ονομάζεται και φάση **δοκιμών**. Στη φάση αυτή γίνονται οι έλεγχοι κατά πόσο το λογισμικό που έχει παραχθεί συμφωνεί με τις προδιαγραφές της ανάλυσης και ανιχνεύονται και διορθώνονται κατά το δυνατόν τα υπάρχοντα λάθη.

Αυτή η φάση μπορεί να εκτελείται παράλληλα με την κωδικοποίηση. Για την ανίχνευση των λαθών οι προγραμματιστές χρησιμοποιούν ξεχωριστό όρο, τον όρο εκσφαλμάτωση (debugging).

Το **προϊόν** της φάσης αυτής είναι η **εφαρμογή** στην τελική της μορφή.

## Λειτουργία και Υποστήριξη

Η τελευταία φάση έχει και τη μεγαλύτερη διάρκεια. Περιλαμβάνει την έναρξη της λειτουργίας του λογισμικού, την συντήρηση και αναβάθμισή του - **υποστήριξη (maintenance)**.

## Τεκμηρίωση

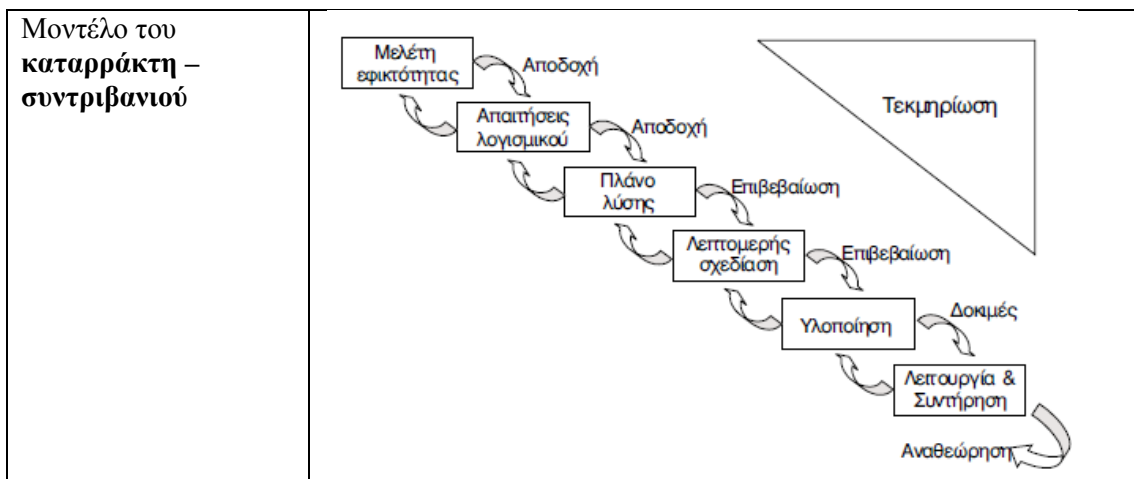
Στη διαδικασία ανάπτυξης του λογισμικού ανήκει και η **τεκμηρίωση (documentation)**. Η τεκμηρίωση δεν αποτελεί ξεχωριστή φάση ανάπτυξης αλλά διαδικασία που πραγματοποιείται σε όλο τον κύκλο ζωής του λογισμικού.

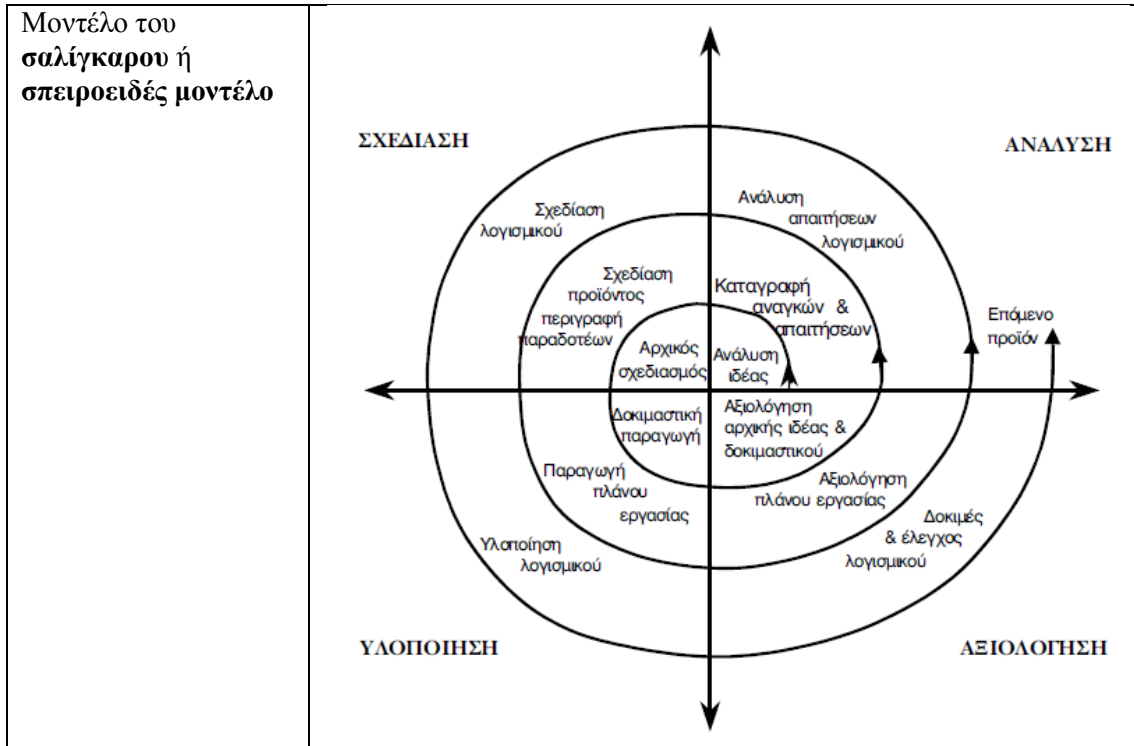
Υπάρχουν δύο είδη τεκμηρίωσης:

- η **εσωτερική τεκμηρίωση** – τα σχόλια μέσα στον κώδικα του προγράμματος
- και η **εξωτερική τεκμηρίωση** – ένας κατασκευαστικός φάκελος που περιλαμβάνει καταγεγραμμένα στοιχεία από την πρώτη φάση μέχρι και την τελευταία (π.χ. λεπτομερή περιγραφή του προβλήματος, αλγόριθμοι που επιλέχθηκαν, δομές δεδομένων κ.ά.).

### 1.2. Μοντέλα σύνδεσης των φάσεων ανάπτυξης λογισμικού

Για τη σύνδεση των φάσεων ανάπτυξης λογισμικού έχουν προταθεί κατά καιρούς πολλά μοντέλα. Δύο από τα πιο κλασικά είναι το μοντέλο του **καταρράκτη – συντριβανιού** (waterfall-fountain model) και το μοντέλο του **σαλίγκαρου** ή **σπειροειδές μοντέλο** (spiral model).





### Ποιος είναι ο ρόλος του προγραμματιστή στις φάσεις ανάπτυξης λογισμικού

Ο **προγραμματιστής** συμμετέχει σε όλες τις φάσεις ανάπτυξης λογισμικού. Αναλύει τα στοιχεία που του περιγράφει ο αναλυτής συστήματος, σχεδιάζει από κοινού με τον αναλυτή τρόπους λύσεις, υλοποιεί το λογισμικό, δοκιμάζει και αξιολογεί τα προγράμματά του, αναλαμβάνει τη συντήρησή τους.

Στη φάση της ανάλυσης του προβλήματος, ο προγραμματιστής προσπαθεί να **κατανοήσει** το πρόβλημα και την εφαρμογή.

Στη φάση της σχεδίασης ο προγραμματιστής **προσφέρει** αναφέροντας τις εμπειρίες του από λύσεις που έχουν δοθεί σε παρόμοια προβλήματα, καταθέτοντας έτοιμα τμήματα λογισμικού από παλαιότερες εργασίες, περιγράφοντας τις δυνατότητες που έχει η γλώσσα προγραμματισμού και το περιβάλλον στο οποίο θα γίνει η υλοποίηση.

Στη φάση της υλοποίησης των προγραμμάτων συμμετέχει κατά κύριο λόγο ο προγραμματιστής. Το στυλ του προγραμματισμού αποτελεί τον πιο σημαντικό παράγοντα δημιουργίας ποιοτικού λογισμικού. Το ποιοτικό πρόγραμμα ταυτίζεται με το απλό μιας και κάνει ευκολότερη την υλοποίηση και τη συντήρηση. Συνήθως, τα απλά προγράμματα είναι και εύκαμπτα (flexible).

Η δουλειά του προγραμματιστή είναι να προσφέρει ένα πρόγραμμα με όσο το δυνατόν λιγότερα σφάλματα. Καλός προγραμματιστής είναι αυτός ο οποίος γράφει προγράμματα αποδοτικά, τα οποία συνοδεύονται από καλή τεκμηρίωση και ο οποίος βρίσκει εύκολα τα λάθη του.

Πολλοί προγραμματιστές περνούν στη φάση της υλοποίησης πολύ γρηγορότερα από ότι θα έπρεπε. Αυτή η βιασύνη να προχωρήσουν σε κωδικοποίηση, πριν δοθούν τελικές λύσεις, οδηγεί σε προβληματικές καταστάσεις στο μέλλον. Επίσης, οι νέοι προγραμματιστές

πιεζόμενοι πολλές φορές από το χρόνο παράδοσης γράφουν συχνά πρόχειρα, γιατί νομίζουν ότι μπορούν να επανέλθουν αργότερα και να κάνουν διορθώσεις. Αυτό σπάνια γίνεται και ένα κακογραμμένο πρόγραμμα παραμένει για χρόνια σ' αυτήν την κατάσταση.