

## Τεχνολογία Λογισμικού (software engineering)

Τομέας που πραγματεύεται τεχνικές, μεθοδολογίες, πρακτικές και εργαλεία για την συστηματική, μεθοδική και ποσοτικοποιημένη προδιαγραφή, σχεδίαση, υλοποίηση, έλεγχο και συντήρηση συστημάτων λογισμικού υψηλής ποιότητας και εντός δεδομένου προϋπολογισμού και χρόνου εκτέλεσης (IEEE Standard 610.12)

### Προγραμματισμός

- Περιορισμένου όγκου έργα
- Υλοποιημένα από μικρές ομάδες
- Με απλές λειτουργικές και μη απαιτήσεις
- Για μια συγκεκριμένη εφαρμογή
- Χρησιμοποιούνται για σχετικά σύντομο διάστημα
- Έχουν μικρό σχετικά κόστος

### Τεχνολογία Λογισμικού

- Μεγάλου όγκου έργα
- Υλοποιημένα από πολλές ομάδες
- Με πολύπλοκες απαιτήσεις
- Σύνολο εφαρμογών
- Έχουν μεγάλο χρονικό ορίζοντα λειτουργίας
- Έχουν μεγάλο κόστος

**Λειτουργικές Απαιτήσεις (Functional Requirements)** → Περιγράφουν τι πρέπει να κάνει το σύστημα. Περιγράφουν τις δυνατότητες του συστήματος και πρέπει να είναι πλήρεις και συνεπείς.

Παραδείγματα **Λειτουργικών Απαιτήσεων**:

1. Transactions
2. Reporting Requirements
3. Administrative functions
4. Authentication
5. Authorization levels
6. External Interfaces
7. Historical Data management
8. Legal or Regulatory Requirements
9. The background color for all windows in the application will be blue and have a hexadecimal RGB color value of 0x0000FF.
10. Only Managerial level employees have the right to view revenue data.

11. Authentication of a user when he/she tries to log into the system.
12. System shutdown in the case of a cyber attack.
13. Verification email is sent to user whenever he/she registers for the first time on some software system.

**Μη Λειτουργικές Απαιτήσεις** → Περιγράφουν το πως (ή το πόσο καλά) το σύστημα θα υποστηρίζει τις λειτουργικές απαιτήσεις.

Παραδείγματα **Μη Λειτουργικών Απαιτήσεων**:

1. Scalability
2. Capacity
3. Availability
4. Reliability
5. Recoverability
6. Data Integrity
7. Service ability
8. Usability
9. Interoperability
10. Every unsuccessful attempt by a user to access an item of data shall be recorded on an audit trail.
11. A website should be capable enough to handle 2 million users without affecting its performance
12. The software should be portable. So moving from one OS to other OS does not create any problem.
13. Modified data in a database should be updated for all users accessing it within 2 seconds.
14. Emails should be sent with a latency of no greater than 12 hours.
15. Each request should be processed within 10 seconds.
16. The site should load in 3 seconds when the number of simultaneous users are > 10000

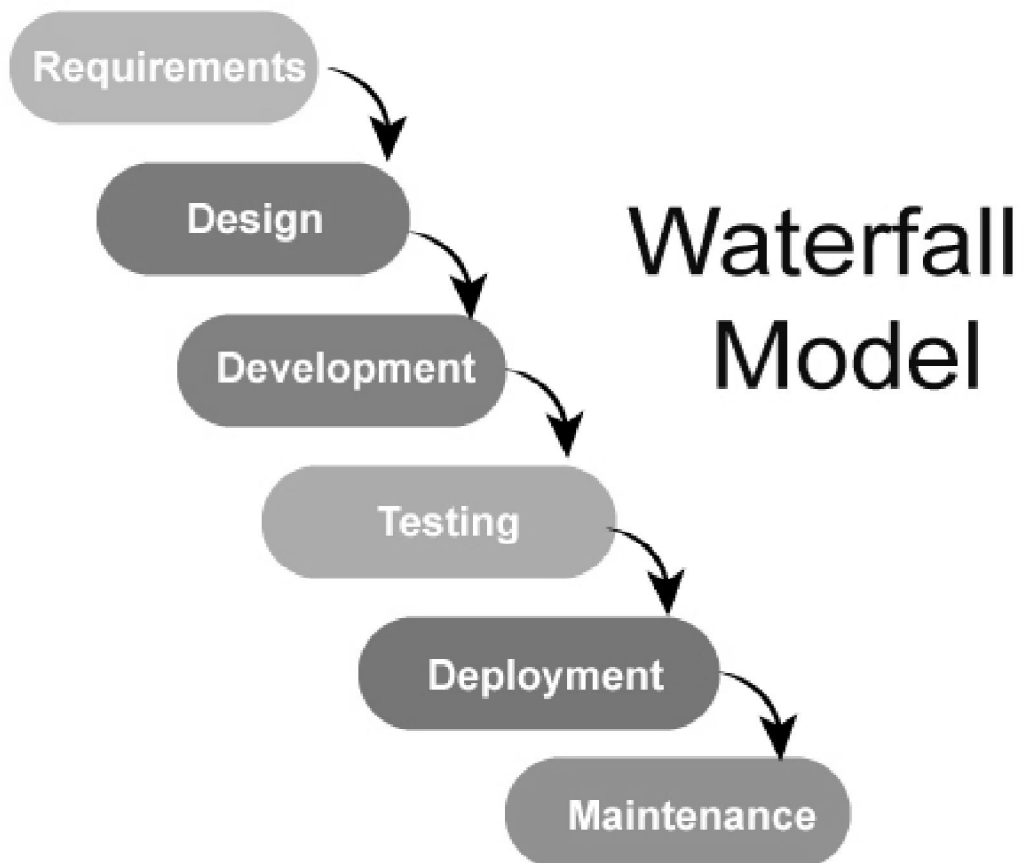
**Λειτουργική Απαίτηση** = Ένα σύστημα φορτώνει ένα ιστότοπο όταν κάποιος χρήστης κλικάρει την αντίστοιχη σελίδα. Η αντίστοιχη **Μη Λειτουργική Απαίτηση** = καθορίζει πόσο γρήγορα η ιστοσελίδα θα φορτωθεί. Μια καθυστέρηση στο φόρτωμα της σελίδας θα δημιουργήσει αρνητική αίσθηση στον χρήστη σχετικά με την ποιότητα του συστήματος, παρότι η λειτουργική απαίτηση ικανοποιείται πλήρως.

## Κύκλος Ζωής Λογισμικού

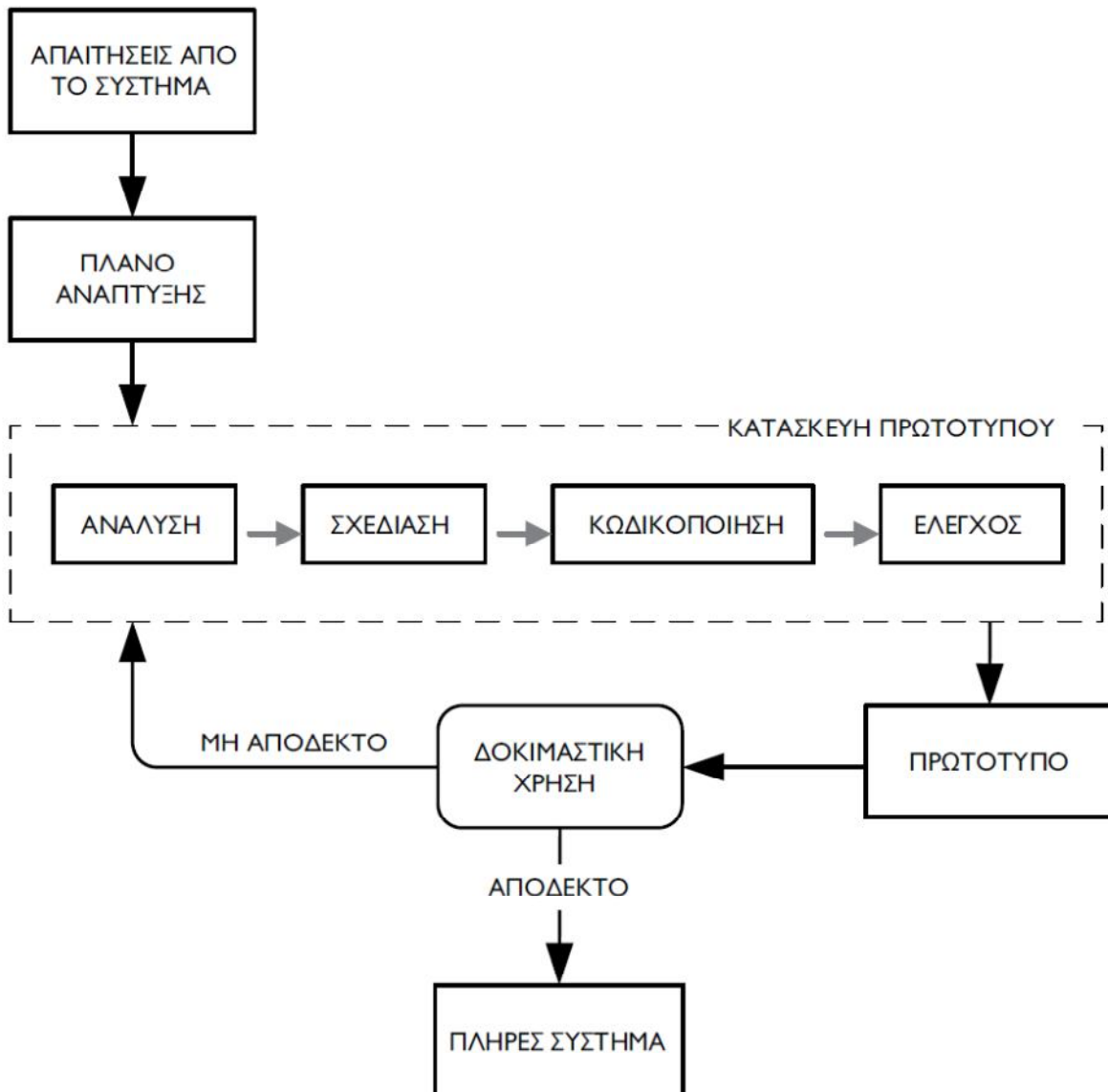
1. Φάση Προδιαγραφής Λειτουργικών και Μη-Λειτουργικών Απαιτήσεων
2. Φάση Ανάλυσης
3. Φάση Σχεδίασης
4. Φάση Υλοποίησης
5. Φάση Ελέγχου και Πιστοποίησης
6. Φάση Ενοποίησης και Εγκατάστασης
7. Φάση Συντήρησης
8. Φάση Αποχώρησης

# Μοντέλα Κύκλου Ζωής

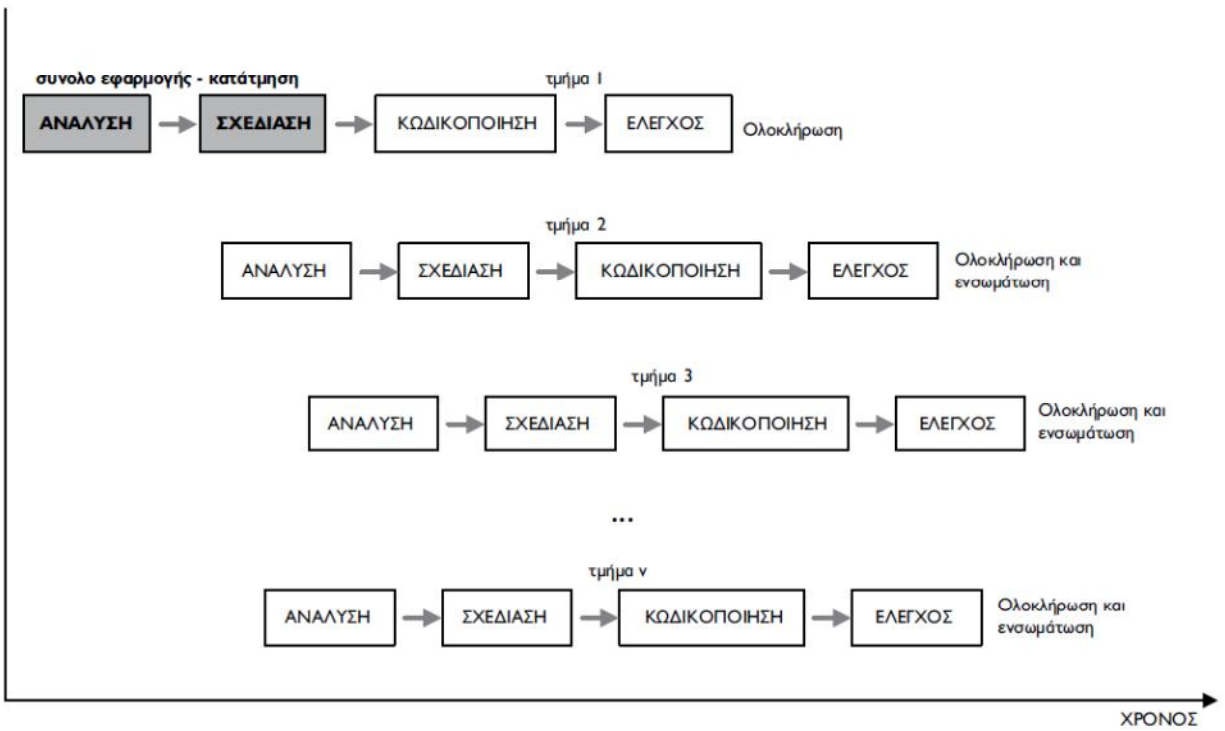
## 1. Το μοντέλο του καταρράκτη



## 2. Το μοντέλο της πρωτοτυποποίησης



### 3. Το μοντέλο της λειτουργικής επαύξεσης





# Rational Unified Process

