



Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Πληροφορική II

Ενότητα 4 : Τεχνολογία λογισμικού

Δρ. Γκόγκος Χρήστος



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Τμήμα Χρηματοοικονομικής & Ελεγκτικής (Παράρτημα Πρέβεζας)

Πληροφορική II

Ενότητα 4 : Τεχνολογία λογισμικού

Δρ. Γκόγκος Χρήστος
Επίκουρος Καθηγητής

Άρτα, 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.





Χρηματοδότηση

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Κρίση Λογισμικού (1968)

- Στην δεκαετία του 1970 παρατηρήθηκαν μαζικά:
 - Μεγάλες καθυστερήσεις στην ολοκλήρωση κατασκευής λογισμικών
 - Μεγαλύτερα κόστη ανάπτυξης λογισμικού από αυτά που είχαν προβλεφθεί αρχικά
 - Λογισμικά με προβλήματα αξιοπιστίας
 - Λογισμικά δύσκολα στην συντήρηση
 - Λογισμικά δύσκολα στην χρήση
 - Λογισμικά με χαμηλές επιδόσεις



Λόγοι αποτυχίας λογισμικού

- Ανεπαρκής τρόπος κατασκευής λογισμικού
- Πολυπλοκότητα υπολογιστικών συστημάτων
- Πολυπλοκότητα εφαρμογών
- Μεγάλες προσδοκίες από τους πελάτες
- Σύγκλιση τεχνολογιών (επικοινωνίες, δίκτυα, γραφικά, βάσεις δεδομένων,...)

Η ικανότητα του ανθρώπου να φαντάζεται πολύπλοκες κατασκευές πάντα θα υπερέχει της δυνατότητάς του να τις κατασκευάζει. (G. Booch)



Τεχνολογία Λογισμικού (Software Engineering)

- Τεχνολογία Λογισμικού είναι η εγκαθίδρυση και η χρήση έγκυρων μεθόδων και αρχών για την δημιουργία αξιόπιστου λογισμικού το οποίο λειτουργεί σε πραγματικά μηχανήματα (1969).



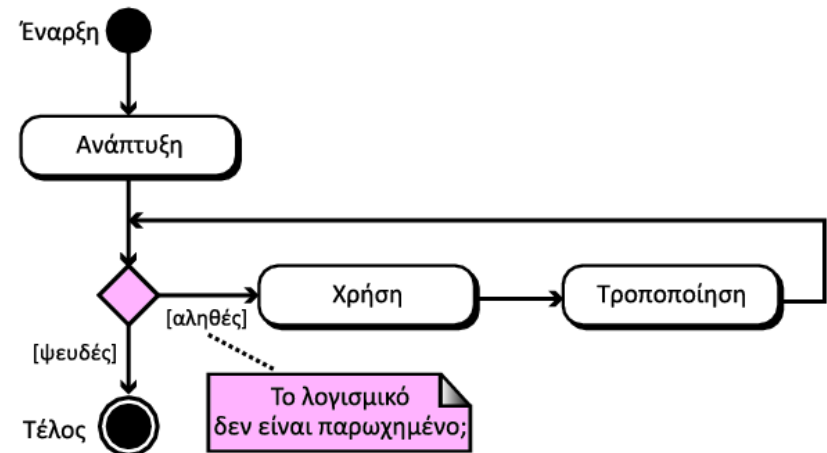
Στόχοι τεχνολογίας λογισμικού

- Λογισμικό που αναπτύσσεται με οικονομικά αποδεκτό τρόπο
- Λογισμικό που ολοκληρώνεται εντός των προθεσμιών
- Λογισμικό που καλύπτει τις προσδοκίες των χρηστών



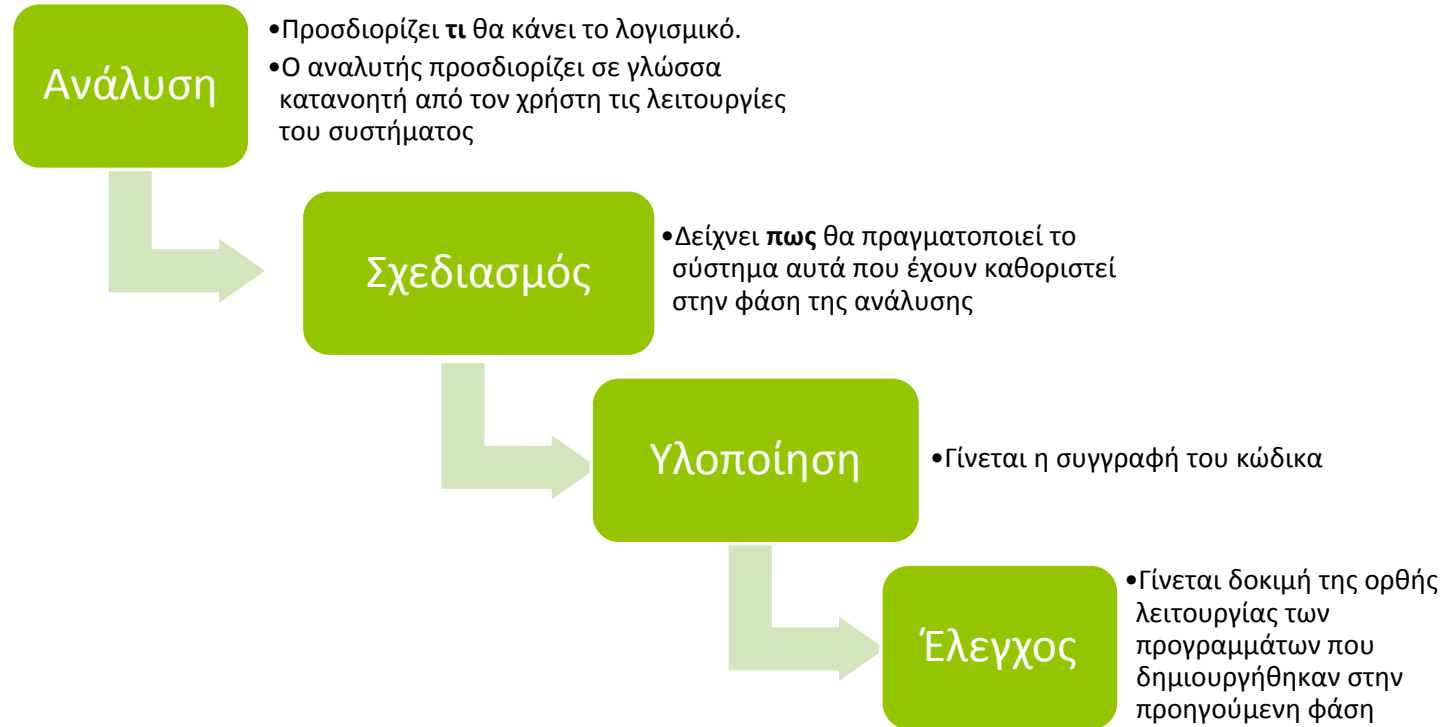
Κύκλος ζωής λογισμικού

- Το λογισμικό περνάει από διάφορες φάσεις ορισμένες από τις οποίες επαναλαμβάνονται και τελικά απορρίπτεται όταν κριθεί ως παρωχημένο





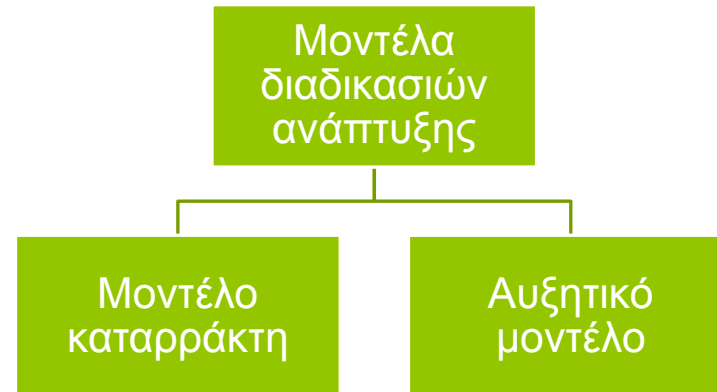
Φάσεις διαδικασίας ανάπτυξης





Βασικά μοντέλα ανάπτυξης λογισμικού

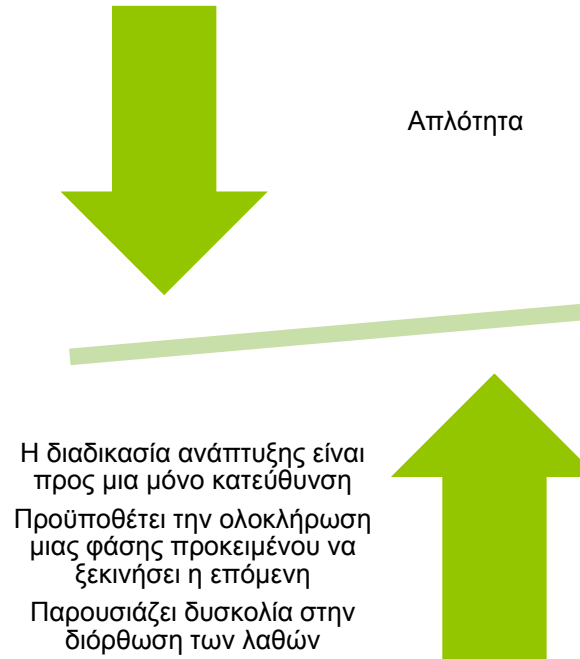
- Η διαδικασία της ανάπτυξης στον κύκλο ζωής λογισμικού περιλαμβάνει τέσσερις φάσεις: ανάλυση, σχεδιασμό, υλοποίηση, και έλεγχο.





Μοντέλο καταρράκτη

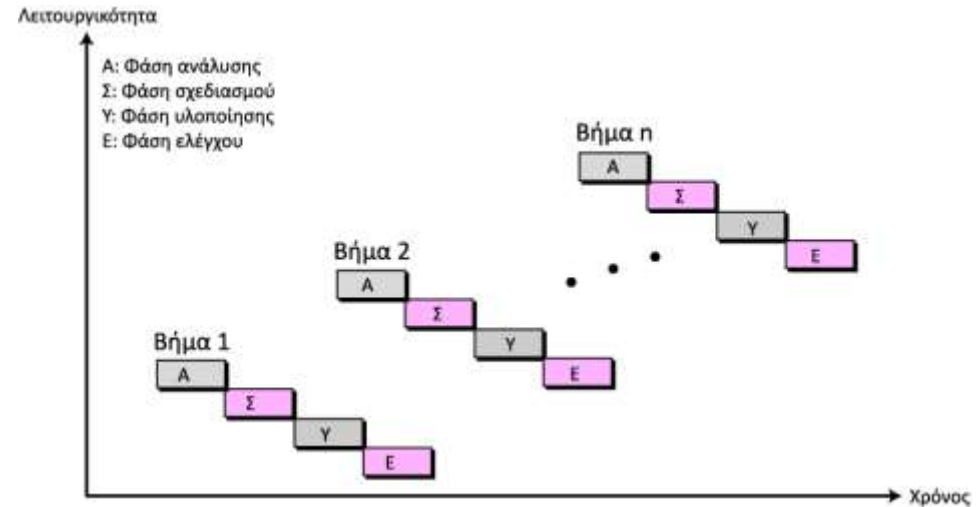
- Στο μοντέλο καταρράκτη η διαδικασία ανάπτυξης είναι προς μια μόνο κατεύθυνση
- Στάδια
 - Ανάλυση
 - Σχεδίαση
 - Υλοποίηση
 - Έλεγχος





Αυξητικό μοντέλο

- Η διαδικασία αναπτύσσεται σε μια σειρά από επαναλήψεις
- Αρχικά ολοκληρώνεται μια απλοποιημένη έκδοση του λογισμικού
- Η πρώτη έκδοση περιέχει τις κύριες λειτουργίες ενώ οι υπόλοιπες συμπληρώνονται σε επόμενες εκδόσεις
- Η διαδικασία συνεχίζεται μέχρι την έκδοση του λογισμικού που περιέχει υλοποιημένες όλες τις λειτουργίες





Φάση ανάλυσης

- Στη φάση ανάλυσης, προσδιορίζεται με όρους κατανοητούς από τους χρήστες, τι θα παρέχει ως λειτουργικότητα το προτεινόμενο σύστημα
- Η φάση ανάλυσης δεν προσδιορίζει πως θα επιτυγχάνονται οι λειτουργίες
- Μπορούν να χρησιμοποιηθούν 2 προσεγγίσεις:
 - Διαδικασιακή ανάλυση
 - Αντικειμενοστραφής ανάλυση



Διαδικασιακή ανάλυση (δομημένη ανάλυση)

- Χρησιμοποιείται όταν πρόκειται η υλοποίηση να γίνει με διαδικασιακές γλώσσες προγραμματισμού (π.χ. C, Pascal, Basic)
- Γίνεται χρήση:
 - Διαγραμμάτων ροής δεδομένων
 - Διαγραμμάτων οντοτήτων συσχετίσεων
 - Διαγράμματα καταστάσεων



Αντικειμενοστραφής ανάλυση

- Χρησιμοποιείται όταν πρόκειται η υλοποίηση να γίνει με αντικειμενοστραφείς γλώσσες προγραμματισμού (π.χ. C++, Java, ...)
- Χρησιμοποιεί:
 - Διαγράμματα περιπτώσεων χρήσης
 - Διαγράμματα κλάσεων
 - Διαγράμματα καταστάσεων



Φάση σχεδιασμού

- Στη φάση σχεδιασμού καθορίζεται πως θα πραγματοποιήσει το σύστημα αυτά που έχουν καθοριστεί στη φάση ανάλυσης
- Μπορούν να χρησιμοποιηθούν 2 προσεγγίσεις:
 - Διαδικασιακός σχεδιασμός
 - Αντικειμενοστραφής σχεδιασμός



Σχεδιασμός

Διαδικασιακός

- Το σύστημα χωρίζεται σε υπομονάδες (modules)
- Χρησιμοποιεί διαγράμματα δομής στα οποία απεικονίζονται οι επιμέρους υπομονάδες του συστήματος

Αντικειμενοστραφής

- Καθορίζονται οι λεπτομέρειες των κλάσεων



Τμηματικότητα (modularity)

- Η τμηματικότητα επιβάλλει την διαίρεση του λογισμικού σε μικρότερες υπομονάδες
- Οι υπομονάδες επικοινωνούν μεταξύ τους για την διεκπεραίωση των λειτουργιών



Σύζευξη – Συνεκτικότητα



Υψηλή
συνεκτικότητα



Χαμηλή
σύζευξη

- **Σύζευξη (coupling)**. Δείχνει πόσο στενά συνδεδεμένες είναι δύο υπομονάδες μεταξύ τους. Επιθυμητό είναι να υπάρχει χαλαρή σύζευξη (loose coupling) διότι:
 - Οι χαλαρές υπομονάδες μπορούν να επαναχρησιμοποιηθούν ευκολότερα.
 - Οι χαλαρές υπομονάδες με σφάλματα έχουν μικρότερη πιθανότητα δημιουργίας σφαλμάτων σε υπομονάδες με τις οποίες επικοινωνούν.
 - Οι χαλαρές υπομονάδες έχουν ευκολότερη συντήρηση.
- **Συνεκτικότητα (cohesion)**. Δείχνει πόσο στενά είναι συνδεδεμένες μεταξύ τους οι διεργασίες του ίδιου υποπρογράμματος. Επιθυμητό είναι να υπάρχει ισχυρή συνεκτικότητα σε ένα υποπρόγραμμα (strong cohesion).



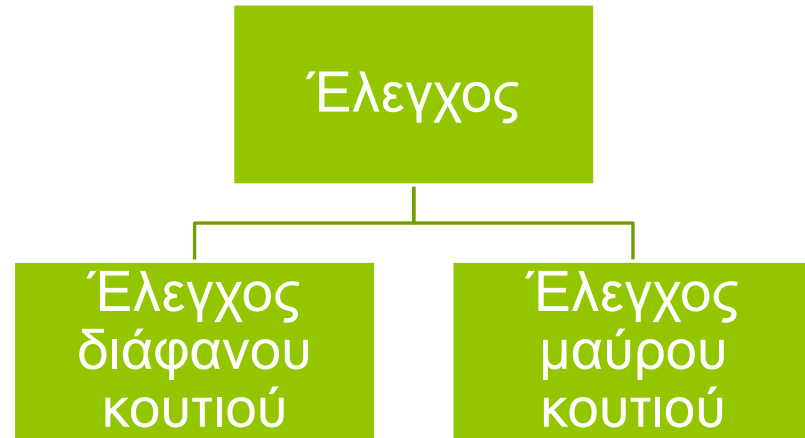
Φάση υλοποίησης

- Επιλογή γλώσσας προγραμματισμού
- Οι προγραμματιστές γράφουν τον κώδικα για τις υπομονάδες που έχουν οριστεί στην φάση σχεδίασης
- Οι οδηγίες από την φάση σχεδίασης θα πρέπει να είναι σαφείς έτσι ώστε να επιτρέπουν την δημιουργία του αντίστοιχου κώδικα



Φάση ελέγχου

- Ο σκοπός της φάσης ελέγχου είναι ο εντοπισμός σφαλμάτων





Έλεγχος «γυάλινου» κουτιού

- Ο ελεγκτής είναι γνώστης του προγράμματος
- Ο προγραμματιστής θα πρέπει να έχει κάνει σχέδια ελέγχου (σενάρια) τα οποία να εξετάζουν την λειτουργία του συστήματος ειδικά κάτω από ασυνήθιστες καταστάσεις
- Για ένα τμήμα προγράμματος συνήθως ένας αριθμός από 20 ως 30 σενάρια είναι επαρκής
- Το συνολικό σχέδιο ελέγχου θα πρέπει να διατηρηθεί από τον προγραμματιστή, έτσι ώστε να έχει την δυνατότητα επαναχρησιμοποίησης του σε περίπτωση που απαιτηθεί κάποια βελτίωση
- Έλεγχος διαδρομών βάσης: εξετάζει αν κάθε εντολή του προγράμματος εκτελείται τουλάχιστον μια φορά (McCabe)



Έλεγχος «μαύρου» κουτιού

- Γίνεται από τον τεχνικό έλεγχο του συστήματος και τον χρήστη
- Ο έλεγχος γίνεται χωρίς να απασχολεί τον ελεγκτή ο τρόπος με τον οποίο εκτελούνται οι εσωτερικά εργασίες
- Ελέγχεται ότι οι απαιτήσεις που προσδιορίστηκαν στην φάση της ανάλυσης ικανοποιούνται
- Μέθοδοι ελέγχου μαύρου κουτιού
 - Διεξοδικός έλεγχος
 - Τυχαίος έλεγχος
 - Έλεγχος οριακών τιμών



Τεκμηρίωση (documentation)

- Η τεκμηρίωση είναι το υλικό σε έντυπη ή σε ηλεκτρονική μορφή που παρέχει οδηγίες και πληροφορίες για το λογισμικό
- Η τεκμηρίωση αποτελεί συνεχή διαδικασία και ακολουθεί τις τροποποιήσεις του πακέτου





Τεκμηρίωση χρήστη (user manual)

- Περιλαμβάνει βήμα προς βήμα οδηγίες χρήσης του πακέτου
- Η σημασία της τεκμηρίωσης χρήστη στο μάρκετινγκ είναι τεράστια





Τεκμηρίωση συστήματος

- Αναφέρεται στο λογισμικό ως πρόγραμμα
- Παρέχει τις απαιτούμενες πληροφορίες έτσι ώστε να επιτρέπει την συντήρηση και τροποποίηση του πακέτου από διαφορετικά άτομα σε σχέση με τους αρχικούς προγραμματιστές
- Η τεκμηρίωση συστήματος πρέπει να καλύπτει και τις 4 φάσεις της ανάπτυξης



Τεχνική τεκμηρίωση

- Περιγράφει την διαδικασία εγκατάστασης και συντήρησης του λογισμικού
- Η εγκατάσταση μπορεί να αφορά την εγκατάσταση του λογισμικού σε διάφορους τύπους υπολογιστικών συστημάτων
- Η συντήρηση μπορεί να περιλαμβάνει τον τρόπο με τον οποίο θα ενημερώνεται το λογισμικό



Ποιότητα λογισμικού

- Ποιοτικό είναι το λογισμικό που ικανοποιεί τις άμεσες και έμμεσες απαιτήσεις του χρήστη, είναι καλά τεκμηριωμένο, καλύπτει τα λειτουργικά πρότυπα της εταιρείας και εκτελείται αποτελεσματικά στο υλικό για το οποίο έχει αναπτυχθεί
- Αριθμητική εκτίμηση της ποιότητας λογισμικού
 - Μέσος αριθμός σφαλμάτων ανά χιλιάδα γραμμών κώδικα.
 - Μέσος χρόνος μεταξύ αστοχιών
 - Αριθμός αιτήσεων χρηστών για αλλαγές
 - Χρόνος απόκρισης λογισμικού

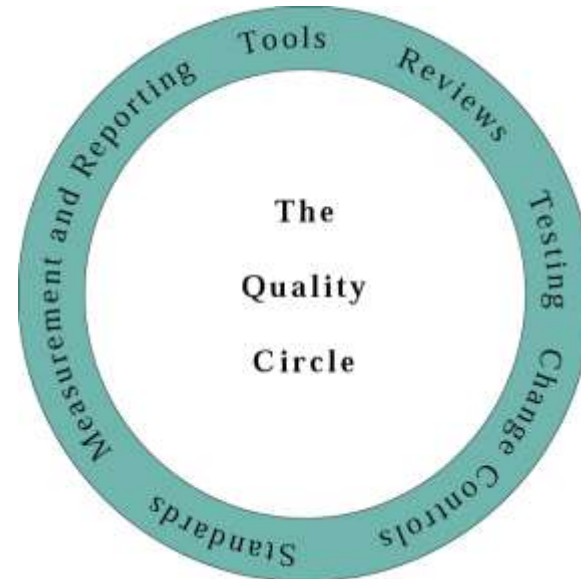


Ο κύκλος της ποιότητας

Υπάρχουν 6 βήματα για την παραγωγή ποιοτικού λογισμικού.

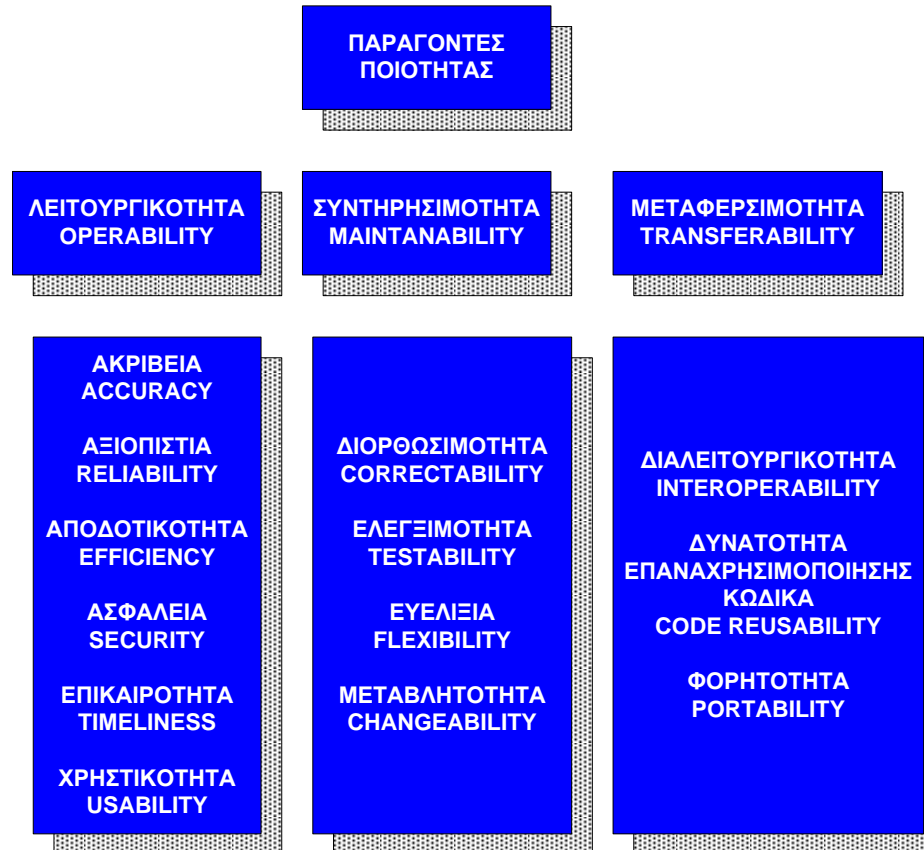
- **Εργαλεία διασφάλισης ποιότητας** (CASE tools, debuggers, unit testers).
- **Τεχνικές ανασκοπήσεις.** Με την ολοκλήρωση επιμέρους σημαντικών εργασιών η ομάδα που εργάστηκε εξηγεί σε επιτροπή το σκεπτικό των αποφάσεών της
- **Τυπικός έλεγχος.** Διασφαλίζει ότι το σύστημα λειτουργεί ως σύνολο
- **Έλεγχος μεταβολών.** Επειδή η ανάπτυξη ενός μεγάλου συστήματος μπορεί να διαρκέσει μήνες ή χρόνια οι αλλαγές που παρουσιάζονται στις απαιτήσεις και στον σχεδιασμό πρέπει να ελέγχονται και να εγκρίνονται από μια επιτροπή ελέγχου μεταβολών.
- **Μετρήσεις ποιοτικών χαρακτηριστικών και αναφορά αποτελεσμάτων**
- Συμμόρφωση με **διεθνή πρότυπα**

- Η ποιότητα ενός πακέτου πρέπει να απορρέει από τον σχεδιασμό του.
- Η ποιότητα δεν μπορεί να προστεθεί εκ των υστέρων.
- Η επιδίωξη ποιότητα είναι μια συνεχής διαδικασία που περνά από όλες τις φάσεις ανάπτυξης ενός πακέτου.





Παράγοντες ποιότητας λογισμικού





Λειτουργικότητα

- Η λειτουργικότητα αναφέρεται στις βασικές λειτουργίες που μπορεί να πραγματοποιεί το λογισμικό
 - **Ακρίβεια:** Το σύστημα δεν θα πρέπει να δίνει λανθασμένα αποτελέσματα
 - **Αποδοτικότητα:** Το σύστημα θα πρέπει να είναι αρκετά γρήγορο έτσι ώστε να διευκολύνει την εργασία των χρηστών
 - **Αξιοπιστία:** Μετρά κατά πόσο οι χρήστες μπορούν να βασίζονται στο σύστημα για να κάνουν τη δουλειά τους
 - **Ασφάλεια:** Εξετάζει πόσο εύκολα μπορούν να αποκτήσουν πρόσβαση στα δεδομένα του συστήματος μη εξουσιοδοτημένα άτομα
 - **Επικαιρότητα:** Εξετάζει αν τα δεδομένα είναι επίκαιρα όταν παρουσιάζονται στον χρήστη
 - **Χρηστικότητα:** Εξετάζει κατά πόσο το σύστημα έχει εύκολη χρήση;



Συντηρησιμότητα

- Η συντηρησιμότητα αναφέρεται στην ευκολία με την οποία ένα σύστημα μπορεί να ανανεώνεται και να συνεχίζει να εκτελείται σωστά
- Μεταβλητότητα: Εξετάζει τον χρόνο που απαιτεί η υλοποίηση μιας αλλαγής
- Διορθωσιμότητα: Εξετάζει τον χρόνο που χρειάζεται για να επαναλειτουργήσει ένα λογισμικό μετά από κάποια αστοχία
- Ευελιξία: Εξετάζει την ευκολία πραγματοποίησης αλλαγών στο λογισμικό
- Ελεγχισιμότητα: Εξετάζει την ευκολία ελέγχου ορθής λειτουργίας του συστήματος



Μεταφερσιμότητα

- Η μεταφερσιμότητα αναφέρεται στην δυνατότητα μεταφοράς δεδομένων κώδικα από μια υπολογιστική πλατφόρμα σε μια άλλη
- **Επαναχρησιμοποίηση:** Εξετάζει αν οι υπομονάδες που αποτελούν το λογισμικό μπορούν να χρησιμοποιηθούν σε άλλα λογισμικά
- **Διαλειτουργικότητα:** Εξετάζει αν το λογισμικό έχει την δυνατότητα αποστολής δεδομένων σε άλλα συστήματα
- **Φορητότητα:** Εξετάζει την δυνατότητα μεταφοράς του λογισμικού από μια πλατφόρμα υλικού σε μια άλλη



Λόγοι αποτυχίας έργων κατασκευής λογισμικού σήμερα

- Μη εφαρμογή αρχών τεχνολογίας λογισμικού
- Αποτυχία διαχείρισης των ρίσκων που συνεπάγεται η ανάπτυξη ενός πακέτου λογισμικού
- «Προδοσία» εκ μέρους της τεχνολογίας
- Πολυπλοκότητα
- Ενσωμάτωση παλιότερων (legacy) συστημάτων στα νέα συστήματα
- Ανομοιογένεια



Βιβλιογραφία

1. Forouzan B., Mosharaf F. Εισαγωγή στην επιστήμη των υπολογιστών. Εκδόσεις Κλειδάριθμος (2010)
2. Σταυρακούδης Α. Εισαγωγή στις υπολογιστικές μεθόδους για τις οικονομικές και επιχειρησιακές σπουδές. Κλειδάριθμος (2012)
3. Ταμπακάς Β. Εισαγωγής τις βάσεις δεδομένων. Εκδότης Β. Ταμπακάς (2009)
4. Γιαννακουδάκης Ε. Σχεδιασμός και διαχείριση Βάσεων Δεδομένων. Εκδόσεις Ευγενία Σ. Μπένου (2009).
5. Biermann A. Σπουδαίες ιδέες στην επιστήμη των υπολογιστών. Πανεπιστημιακές εκδόσεις Κρήτης (2008).
6. Brookshear J.G. Η επιστήμη των υπολογιστών, μια ολοκληρωμένη παρουσίαση. Εκδόσεις Κλειδάριθμος (2009).
7. Πληροφοριακά συστήματα επιχειρήσεων II. Πολλάλης, Γιαννακόπουλος, Δημόπουλος. Εκδόσεις Σταμούλη (2004).



Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Δρ. Γκόγκος Χρήστος.
Πληροφορική II.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή
διεύθυνση:

<http://eclass.teiep.gr/OpenClass/courses/ACC137/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης
Άρτα, 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Τέλος Ενότητας

Τεχνολογία λογισμικού



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

