Αρχιτεκτονικές Πληροφοριακών Συστημάτων

Τεχνολογία Λογισμικού

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)

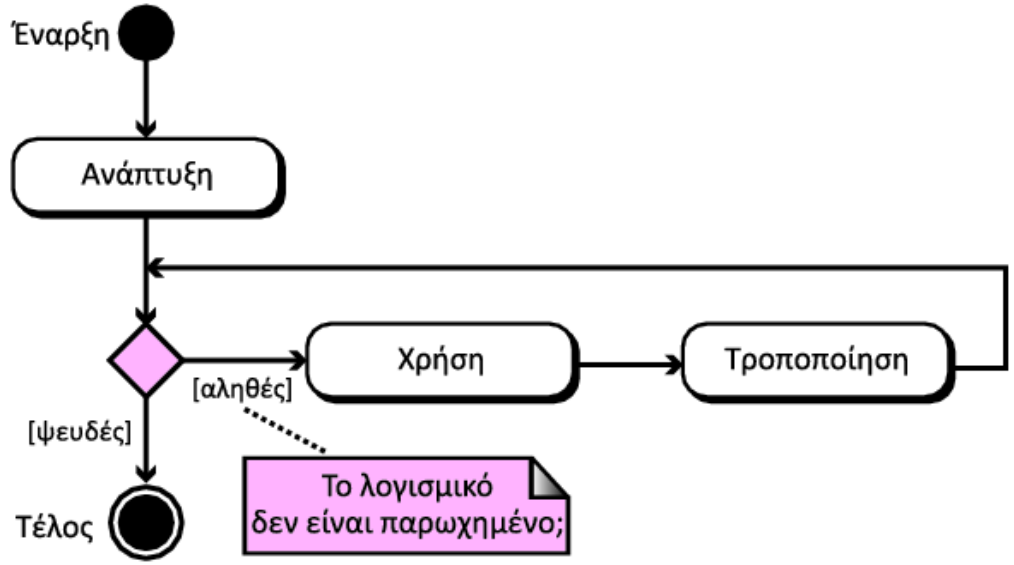
- Τεχνολογία Λογισμικού είναι η εγκαθίδρυση και η χρήση έγκυρων μεθόδων και αρχών για την δημιουργία αξιόπιστου λογισμικού το οποίο λειτουργεί σε πραγματικά μηχανήματα (1969).

Στόχοι τεχνολογίας λογισμικού

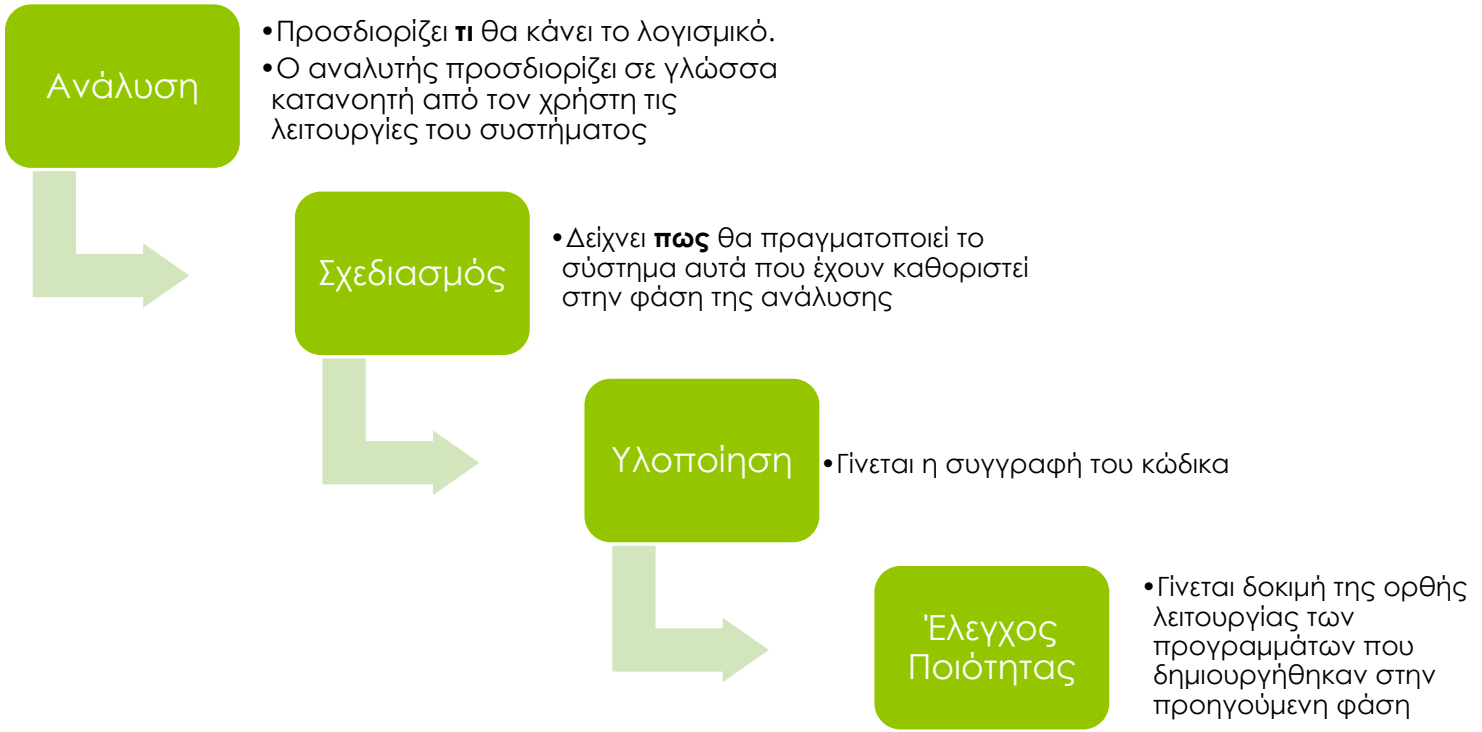
- Λογισμικό που αναπτύσσεται με οικονομικά αποδεκτό τρόπο
- Λογισμικό που ολοκληρώνεται εντός των προθεσμιών
- Λογισμικό που καλύπτει τις προσδοκίες των χρηστών

Κύκλος ζωής λογισμικού

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

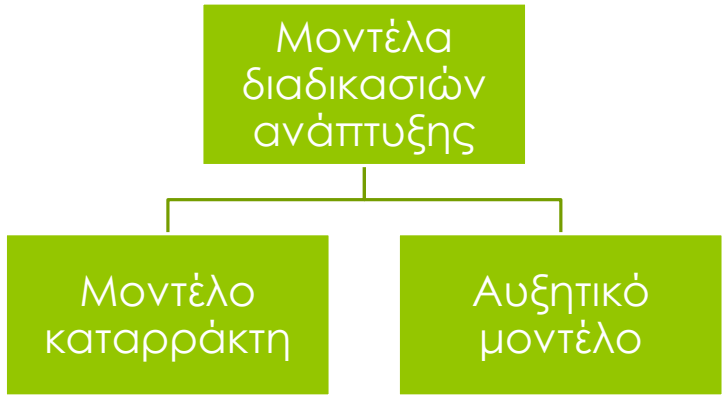


Φάσεις διαδικασίας ανάπτυξης



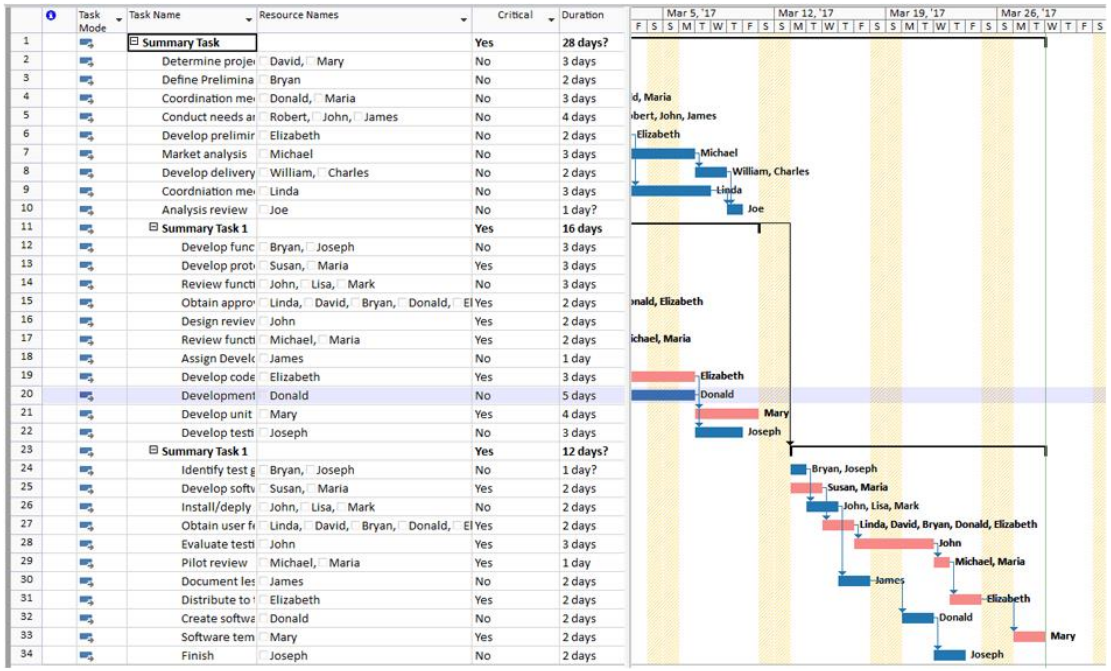
Βασικά μοντέλα ανάπτυξης λογισμικού

- 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



	Task Mode	Task Name	Resource Names	Critical	Duration
1		Summary Task		Yes	28 days?
2		Determine proje	David, Mary	No	3 days
3		Define Prelimina	Bryan	No	2 days
4		Coordination me	Donald, Maria	No	3 days
5		Conduct needs a	Robert, John, James	No	4 days
6		Develop prelimir	Elizabeth	No	2 days
7		Market analysis	Michael	No	3 days
8		Develop delivery	William, Charles	No	2 days
9		Coordniation me	Linda	No	3 days
10		Analysis review	Joe	No	1 day?
11		Summary Task 1		Yes	16 days
12		Develop func	Bryan, Joseph	No	3 days
13		Develop prot	Susan, Maria	Yes	3 days
14		Review functi	John, Lisa, Mark	No	3 days
15		Obtain appro	Linda, David, Bryan, Donald, El	Yes	2 days
16		Design reviev	John	Yes	2 days
17		Review functi	Michael, Maria	Yes	2 days
18		Assign Develc	James	No	1 day
19		Develop code	Elizabeth	Yes	3 days
20		Development	Donald	No	5 days
21		Develop unit	Mary	Yes	4 days
22		Develop testi	Joseph	No	3 days
23		Summary Task 1		Yes	12 days?
24		Identify test g	Bryan, Joseph	No	1 day?
25		Develop softv	Susan, Maria	Yes	2 days
26		Install/deply	John, Lisa, Mark	No	2 days
27		Obtain user fe	Linda, David, Bryan, Donald, El	Yes	2 days
28		Evaluate testi	John	Yes	3 days
29		Pilot review	Michael, Maria	Yes	1 day
30		Document les	James	No	2 days
31		Distribute to t	Elizabeth	Yes	2 days
32		Create softwa	Donald	No	2 days
33		Software tem	Mary	Yes	2 days
34		Finish	Joseph	No	2 days

Μοντέλο καταρράκτη

Απλότητα

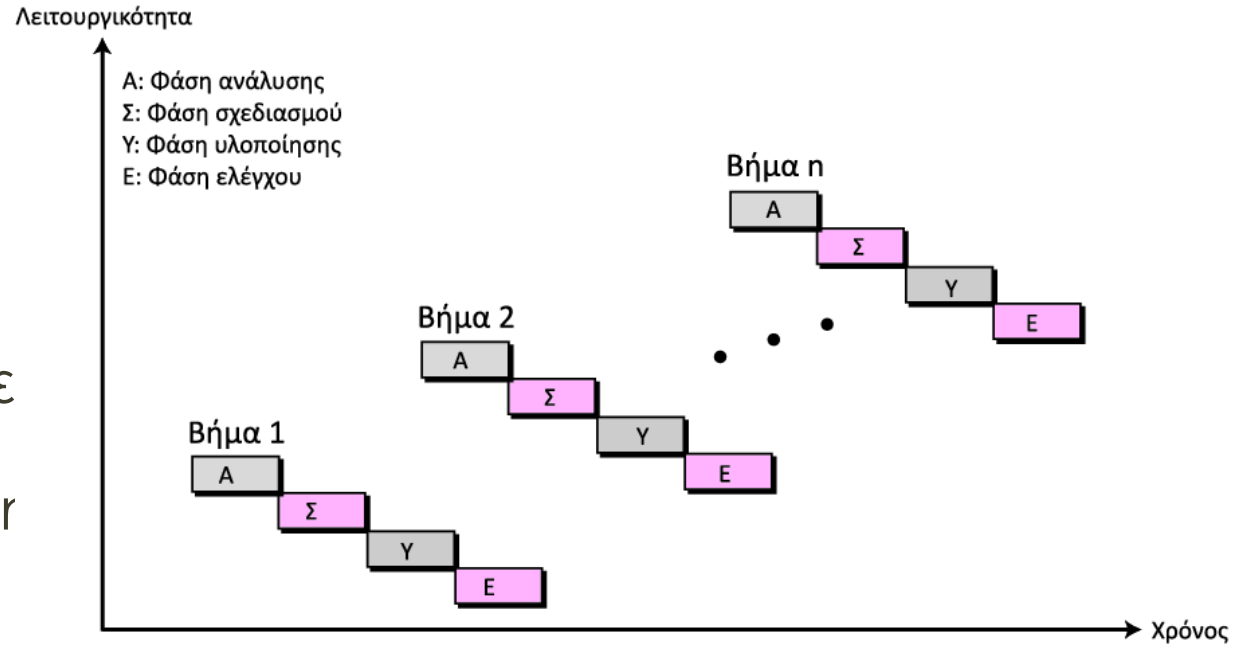
Η διαδικασία ανάπτυξης είναι προς μια μόνο κατεύθυνση

Προϋποθέτει την ολοκλήρωση μιας φάσης προκειμένου να ξεκινήσει η επόμενη

Παρουσιάζει δυσκολία στην διόρθωση των λαθών

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 τη έκδοση του λογισμικού που περιέχει υλοποιημένες όλες τις λειτουργίες



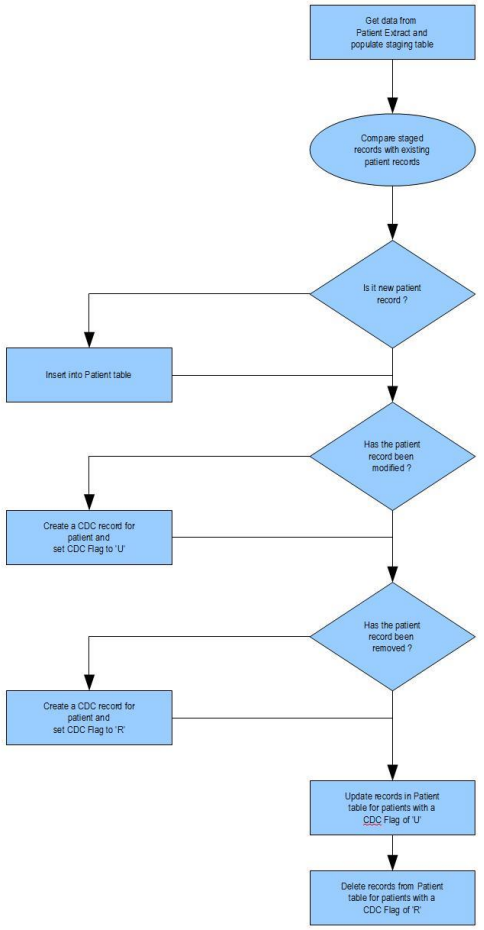
Φάση ανάλυσης

- 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)

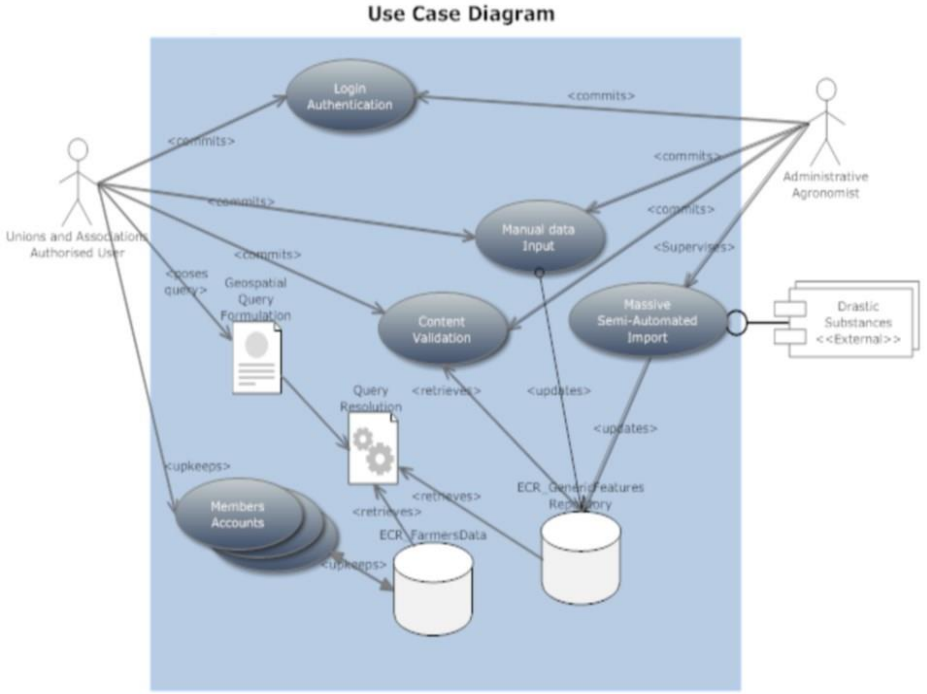
- Γίνεται χρήση:
  - Διαγραμμάτων ροής δεδομένων
  - Διαγραμμάτων οντοτήτων συσχετίσεων
  - Διαγράμματα καταστάσεων



Διάγραμμα ροής

Αντικειμενοστραφής ανάλυση

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



Unified Modeling Language: Ενοποιημένη αναπαράσταση σχεδιασμού και λειτουργικότητας αντικειμένων λογισμικού

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

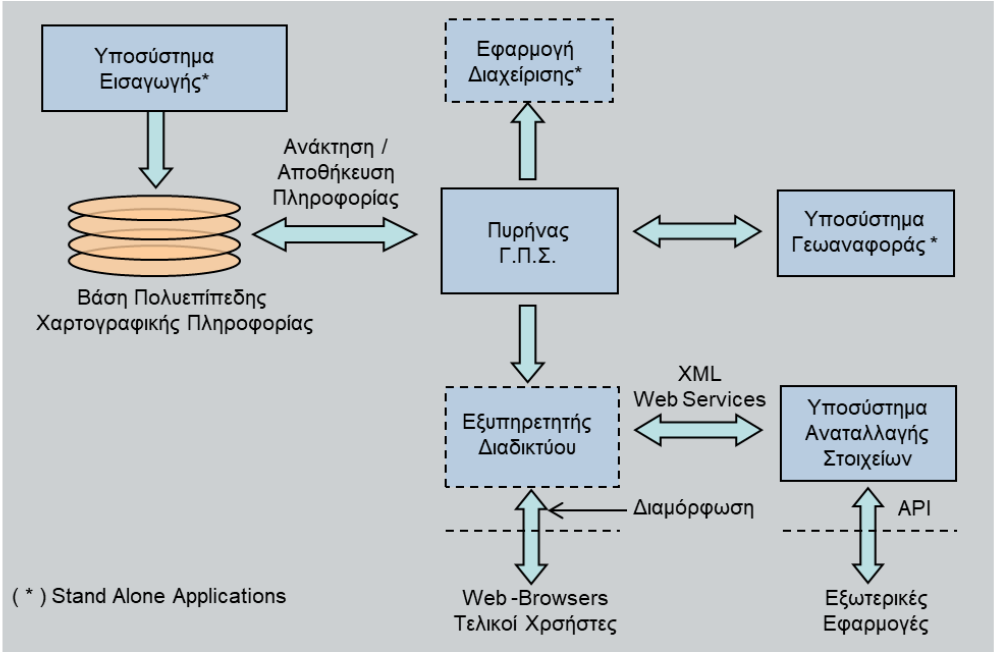
- Το σύστημα χωρίζεται σε υπομονάδες (modules)
- Χρησιμοποιεί διαγράμματα δομής στα οποία απεικονίζονται οι επιμέρους υπομονάδες του συστήματος

Αντικειμενοστραφής

- Καθορίζονται οι λεπτομέρειες των κλάσεων

Τμηματικότητα (modularity)

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

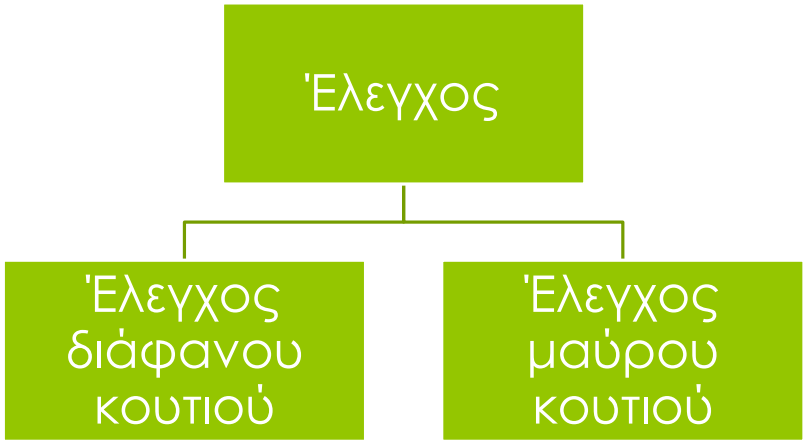


Φάση υλοποίησης

- 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
- Οι περισσότερες σύγχρονες εταιρείες ανάπτυξης λογισμικού διαθέτουν ειδικά τμήματα για το λόγο αυτό (Quality Assurance)



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

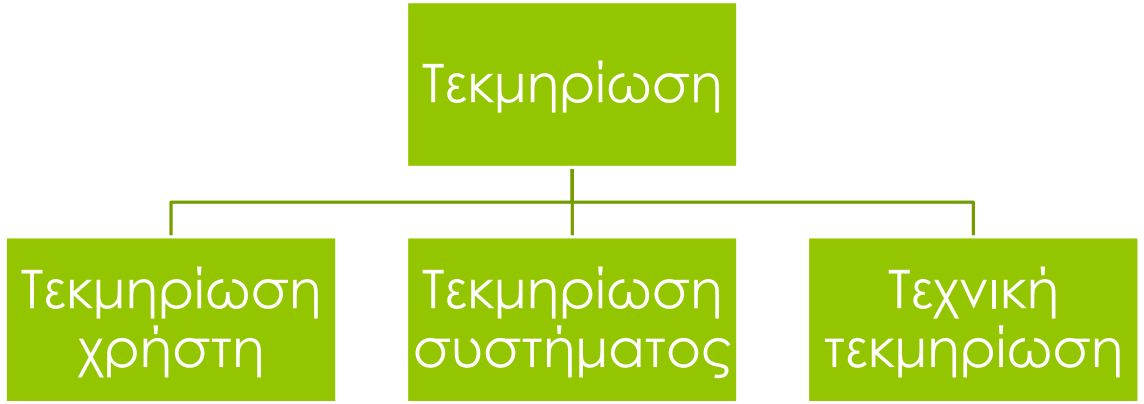
- Έλεγχος διαδρομών βάσης: εξετάζει αν κάθε εντολή του προγράμματος εκτελείται τουλάχιστον μια φορά (McCabe)

Έλεγχος «μαύρου» κουτιού

- Γίνεται από τον τεχνικό ελέγχου 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

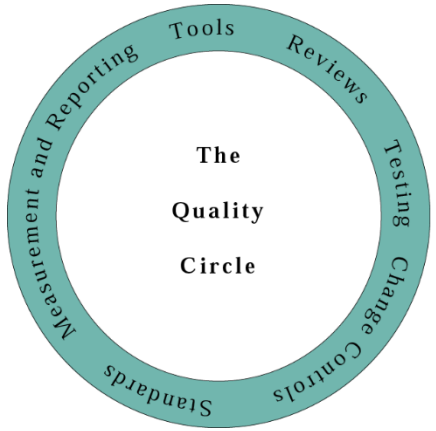
Ποιότητα λογισμικού

- 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

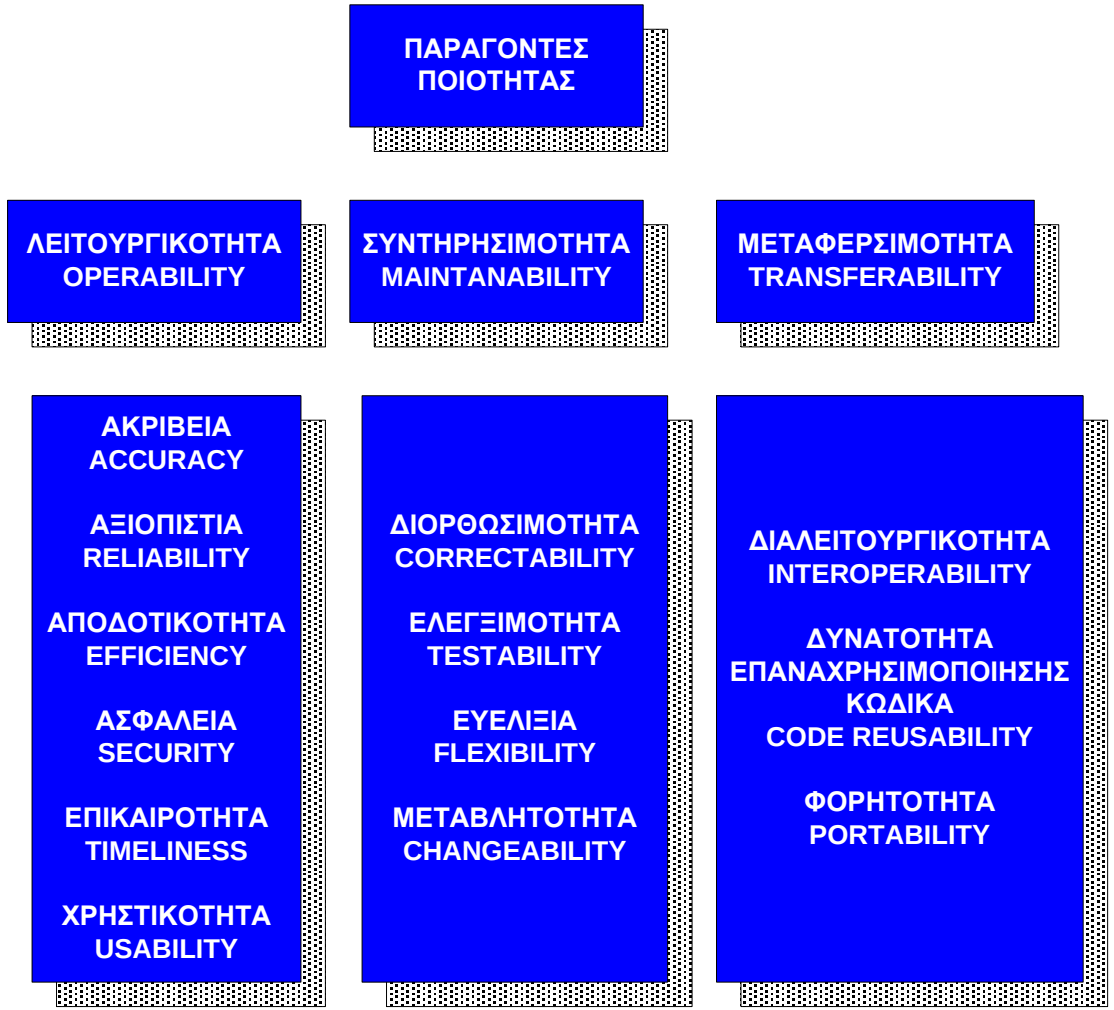
Ο κύκλος της ποιότητας

- 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
- **Ελεγξιμότητα**: 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

Νέες τάσεις στη διαχείριση ανάπτυξης λογισμικού

Ευέλικτες διαδικασίες

Ευέλικτη ανάπτυξη λογισμικού

- Αξιοποιώντας την εμπειρία στην ανάπτυξης λογισμικού, αρκετές σύγχρονες ευέλικτες διαδικασίες έχουν προταθεί και εφαρμόζονται στη διαχείριση των project από τις μεγάλες εταιρείες λογισμικού

- Αυτές περιγράφονται με τον όρο "**Agile processes**"

Agile = Ευελιξία

- Η αυστηρά ιεραρχική εταιρική οργάνωση σε τμήματα (π.χ. τμήμα Βάσεων Δεδομένων, Τμήμα GUI, κλπ) δίνει τη θέση της στην οργάνωση στη βάση μικρών ομάδων, οι οποίες διαθέτουν άτομα από κάθε απαραίτητη ειδικότητα.
- Οι ομάδες και τα άτομα που συμμετέχουν σε αυτές έχουν την τάση να αυτοοργανώνονται, και να αλληλοεπικαλύπτονται
- Κάθε ομάδα διαλέγει μόνη της το πώς είναι αποδοτικότερο να λύσει το πρόβλημα που της τίθεται κάθε φορά
- Η διαδικασίες Agile εστιάζουν στην αποτελεσματικότητα και παραμερίζουν άγονες και λιγότερο ουσιώδεις διοικητικές διαδικασίες

Agile: Βασικές αρχές

1. Ικανοποίηση του πελάτη από την έγκαιρη και διαρκή παράδοση πολύτιμου λογισμικού
2. Αλλαγές στις απαιτήσεις είναι καλοδεχούμενες, ακόμα και σε προχωρημένα στάδια του έργου
3. Παράδοση τμημάτων λειτουργικού λογισμικού σε σύντομο διάστημα (1-2 εβδομάδες συνήθως)
4. Διαρκής καθημερινή συνεργασία μεταξύ των προγραμματιστών και των manager της εταιρείας
5. Εμπιστοσύνη στα μέλη της ομάδας τα οποία έχουν προσωπικό κίνητρο
6. Η επαφή πρόσωπο με πρόσωπο είναι η αποδοτικότερη μορφή επικοινωνίας

Agile: Βασικές αρχές

7. Λογισμικό που λειτουργεί είναι το βασικό μέτρο προόδου της διαδικασίας
8. Αειφορική ανάπτυξη, ικανή να διατηρήσει μια σταθερή πορεία
9. Διαρκής προσοχή στη διασφάλιση της τεχνικής υπεροχής και καλού σχεδιασμού του λογισμικού
10. Απλότητα- η τέχνη της μεγιστοποίησης του κομματιού της δουλειάς που δεν χρειάζεται να γίνει
11. Οι καλύτερες αρχιτεκτονικές, απαιτήσεις και σχεδιασμοί ανακύπτουν από αυτοοργανωνόμενες ομάδες
12. Η ομάδα αντιδρά μόνη της και εφευρίσκει τρόπους να προσαρμόζεται κατάλληλα και να γίνεται αποδοτικότερη

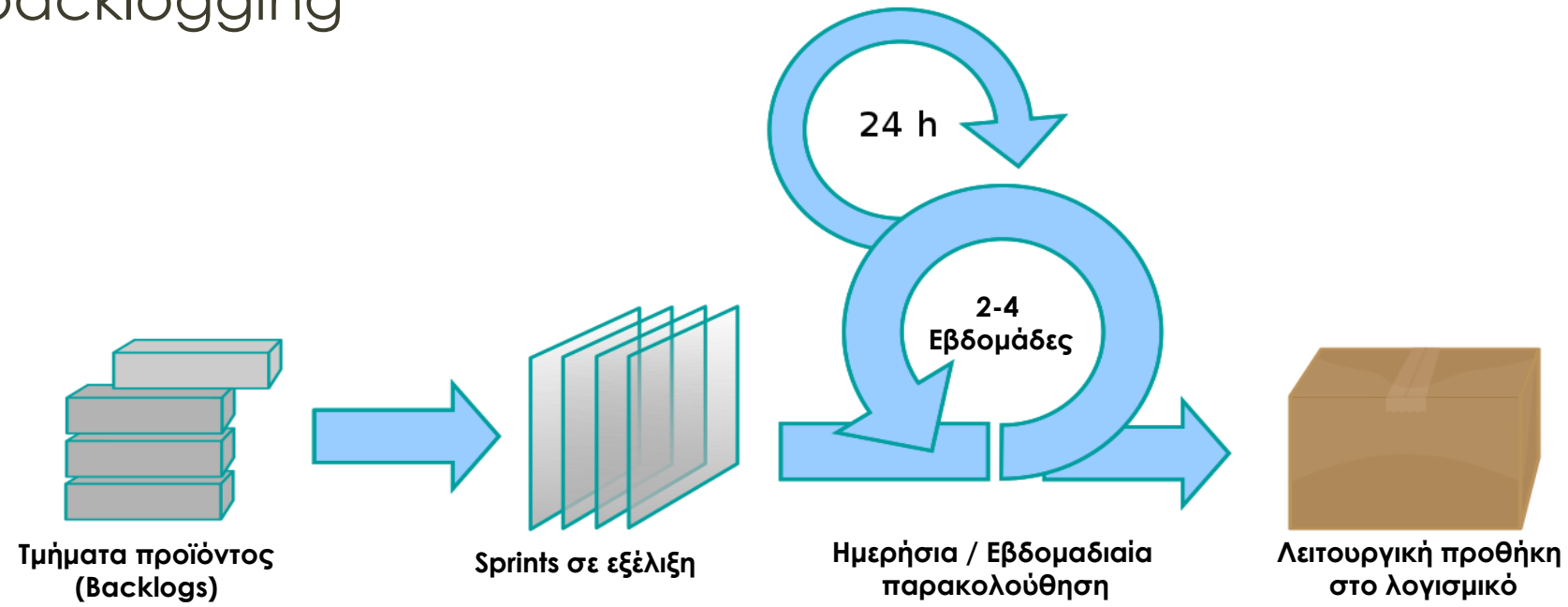
Agile: Πως δουλεύει

- Δύο είναι οι επικρατέστερες μέθοδοι οι οποίες υλοποιούν διαδικασίες Agile:
  - Η μέθοδος Kanban
  - Η μέθοδος Scrum

Η δεύτερη κερδίζει διαρκώς έδαφος στις περισσότερες περιπτώσεις

Agile: Μέθοδος "Scrum"

- Το κάθε «πρόβλημα» σπάει σε μια σειρά από επιμέρους προβλήματα. Αυτό είναι γνωστό με τον όρο "backlogging"



Agile: Μέθοδος “Scrum”

- Η ομάδα συναντιέται κάθε πρωί για 15’ εξετάζει προβλήματα και θέτει στόχους μέρα
- Υπάρχει ο υπεύθυνος scrum κάθε ομάδας, ο οποίος έχει ως βασική του αρμοδιότητα την τήρηση του εσωτερικού χρονοδιαγράμματος και την έγκαιρη ολοκλήρωση του sprint
- Από τη στιγμή που θα ξεκινήσει ένα sprint δεν σταματάει αν δεν ολοκληρωθεί
- Τα sprint υλοποιούν ένα κομμάτι έργου διάρκειας περίπου 2-4 εβδομάδων
- Το αποτέλεσμα κάθε sprint είναι αυτοτελές, και ενσωματώνεται στο υφιστάμενο λογισμικό, το οποίο «ξαναχτίζεται» διαρκώς σε ημερήσια-εβδομαδιαία βάση