Διάλεξη 1

Εισαγωγή στην Τεχνολογία Λογισμικού

Δομή Μαθήματος

- Εισαγωγή
- Μοντέλα διεργασιών ανάπτυξης λογισμικού
- Διαχείριση έργων λογισμικού
- Το μοντέλο CoCoMo
- Εξαγωγή απαιτήσεων
- Περιπτώσεις χρήσης
- Μοντέλα χρήσης
- Μοντέλα αλληλεπίδρασης
- Σχεδίαση λογισμικού
- Αρχιτεκτονική λογισμικού
- Έλεγχος λογισμικού
- Ποιότητα λογισμικού
- Σύγχρονες μεθοδολογίες ανάπτυξης λογισμικού (Agile)

Απαιτήσεις

- Τελική εξέταση (65%)
- Project (35%)
  - Σε μεγάλες ομάδες ~6 άτομα
  - Σε φάσεις
  - Με στενή παρακολούθηση
  - Όλες οι φάσεις ενός έργου λογισμικού (και κώδικας αλλά όχι μόνο!)
  - Εξοικείωση με εργαλεία
- Προαπαιτούμενα: ΗΥ120
  - Απαραίτητη πολύ καλή γνώση C/C++/Java ...
  - ... και άλλων τεχνολογιών

Προσωπικό

- Διδάσκων: Χ. Αντωνόπουλος
  - Γραφείο Β3/5, Γκλαβάνη 37
  - Ώρες γραφείου: Στη σελίδα του μαθήματος
  - E-mail: cda@inf.uth.gr, cdantonop@gmail.com
  - Τηλ: 24210 74578
  - URL: http://www.inf.uth.gr/~cda
- Υπεύθυνη project: Χ. Τσαλαπάτα
  - Γραφείο Γ4, Ιάσονος 10
  - E-mail: htsalapa@inf.uth.gr
  - Τηλ: 24210 74513
- Μεταπτυχιακός Βοηθός
  - Δημήτρης Μαχαίρας

Ιστοσελίδα μαθήματος / Λίστα e-mail / Forum

- Ιστοσελίδα μαθήματος
  - http://www.inf.uth.gr/courses/CE420
- Λίστα ηλεκτρονικού ταχυδρομείου
  - http://inf-server.inf.uth.gr/mailman/listinfo/CE420
- Forum προγραμματισμού
  - http://courses.inf.uth.gr/coding

Συγγράμματα

- Βασικές Αρχές Τεχνολογίας Λογισμικού, I. Sommerville, Εκδόσεις Κλειδάριθμος ΕΠΕ, 2009, Αθήνα

- Αντικειμενοστρεφής Ανάπτυξη Λογισμικού με τη UML, Β. Γερογιάννης, Γ. Κακαρόντζας, Α. Καμέας, Ι. Σταμέλος, Π. Φιτσιλής, Εκδόσεις Κλειδάριθμος ΕΠΕ, 2006, Αθήνα

Περιεχόμενα

- Η σημασία της Τεχνολογίας Λογισμικού
- Το ιστορικό της Τεχνολογίας Λογισμικού
- Η έννοια του «καλού λογισμικού»
- Προσέγγιση από πλευράς συστήματος και σχεδίασης

Το λογισμικό

- 1950, αρχή
- Έδωσε δυνατότητες σε υπάρχουσες τεχνολογίες (πχ. Τηλεπικοινωνίες)
- Δημιούργησε νέες επιστήμες (π.χ. γενετική βιολογία)
- Βελτίωση ποιότητα ζωής
  - Υγεία, τηλεϋπηρεσίες, έρευνα, κ.λπ.

Που βρίσκεται το λογισμικό;

- Άϋλο («εκτελείται», αποθηκεύεται)
- Δεν είναι αντικείμενο των 5 αισθήσεών μας
- Δεν φθείρεται με τον χρόνο
- Δεν υπακούει στους νόμους παραγωγής-ανάπτυξης των υλικών αγαθών
- Διαφοροποιείται από τον τρόπο αξιοποίησης των υλικών αγαθών

Ορισμός λογισμικού

- **Σύνολο προγραμμάτων που περιλαμβάνουν**
  - **δομές δεδομένων (διαχείριση πληροφορίας)**
  - **Εντολές (παρέχουν λειτουργίες)**
  - **Τεκμηρίωση (περιγράφει τρόπο λειτουργίας των προγραμμάτων)**

Η τεχνολογία λογισμικού

- «Είναι κλάδος της πληροφορικής που ασχολείται με την μελέτη και την εφαρμογή των συστηματικών, μεθοδικών και ποσοτικοποιημένων προσεγγίσεων για την ανάπτυξη, λειτουργία και συντήρηση του λογισμικού» (IEEE 90)

- Δεν έχει αποκτήσει ακόμη τα χαρακτηριστικά της επιστήμης, που θα της επιτρέπουν να αποδεικνύει την ύπαρξη βέλτιστης, ορθής, πλήρους και αξιόπιστης λύσης

Ο ρόλος και η σημασία του λογισμικού

- Μέρος κάθε μηχανής που χρησιμοποιούμε είναι λογισμικό
- Διεκπεραιώνει 2 κατηγορίες λειτουργιών
  - Μετασχηματιστής πληροφορίας
    - Διαχειρίζεται γενικά την πληροφορία
    - (π.χ. λογισμικό δοσοληψιών ΑΤΜ)
  - Ελεγκτής-συντονιστής
    - Διαχειρίζεται τους πόρους μηχανής/ών για την εξυπηρέτηση των αναγκών των χρηστών
    - (π.χ. δικτυακό πρόγραμμα επικοινωνίας ΑΤΜ – κεντρ. Η/Υ)

Κατηγορίες λογισμικού

- Συστήματος
- Εφαρμογής
- Επιστημονικό
- Ενσωματωμένο
- Γραμμής παραγωγής
- Διαδικτυακών εφαρμογών
- Τεχνητής νοημοσύνης
- Ανοικτού κώδικα

Παράγοντες που ενισχύουν την σημασία του λογισμικού

- Οι αλλαγές στον λόγο κόστους υλικού προς κόστος λογισμικού (μειώνεται)
- Η αυξανόμενη σημασία της συντήρησης λογισμικού
- Οι εξελίξεις στο υλικό
- Οι εξελίξεις στις τεχνικές λογισμικού
- Οι αυξανόμενες απαιτήσεις για λογισμικό

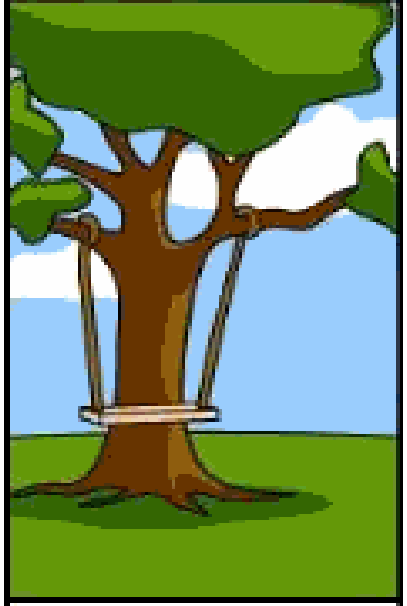
Η Τεχνολογία Λογισμικού

- Τέλη 1960 «κρίση λογισμικού»
- Μεγάλο ποσοστό των έργων παρουσίαζαν:
  - Αποκλίσεις από
    - Λειτουργικότητα που παρήγγειλαν οι πελάτες
    - Χρονοδιάγραμμα ανάπτυξης
    - Οικονομικό προγραμματισμό
    - Προβλεπόμενο κόστος περιβάλλοντος λειτουργίας
  - Αδυναμία εξέλιξης του λογισμικού ώστε να προσαρμοστεί στις ανάγκες των πελατών

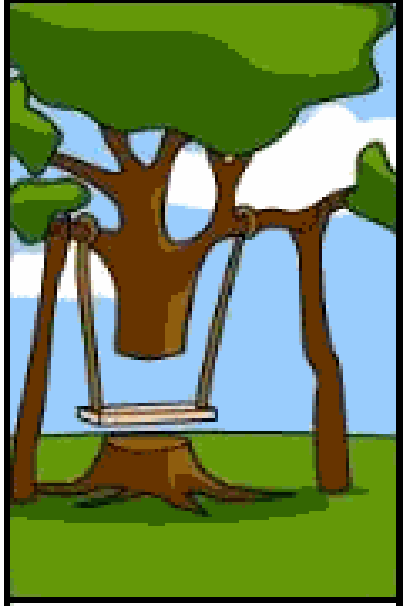
Γιατί είναι σημαντικό το μάθημα?

- Τα περισσότερα λάθη (54%) ανακαλύπτονται μετά την κωδικοποίηση και τον έλεγχο.
- Σχεδόν μισά (45%) από τα λάθη εντοπίζονται στις απαιτήσεις και τη σχεδίαση.
- Τα περισσότερα λάθη κατά τη φάση της ανάλυσης απαιτήσεων (77%) δεν οφείλονται στους αναλυτές αλλά προκύπτουν λόγω λανθασμένων δεδομένων, ασυνεπειών, παραλήψεων και ασαφειών.
- Η διόρθωση λαθών στις απαιτήσεις μπορεί να κοστίσει και 100 φορές παραπάνω από τα λάθη υλοποίησης.
- Τα λάθη στην καταγραφή απαιτήσεων μπορούν να εντοπιστούν καθώς οι τεχνικές επιθεώρησης έχουν αποδειχτεί αποτελεσματικές και μπορούν να εφαρμοστούν και στο σχέδιο και στον κώδικα

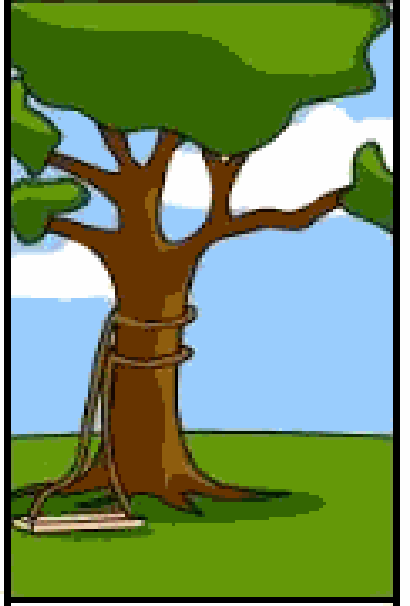
How the customer explained it



How the Project Leader understood it

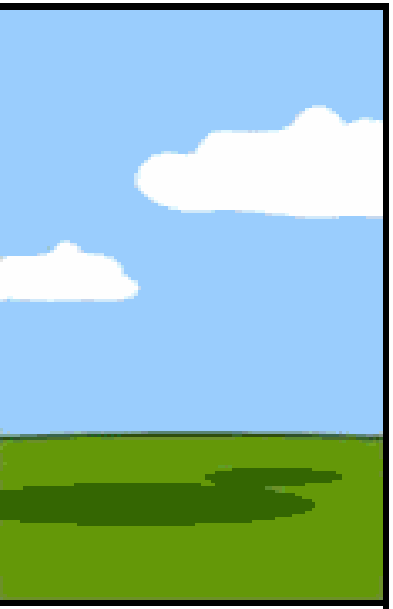


How the Analyst designed it

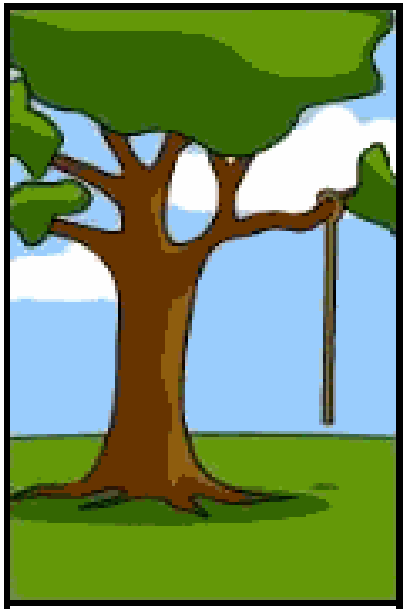


How the Programmer wrote t

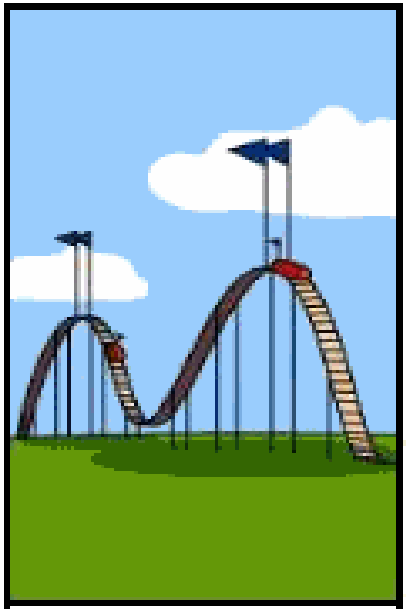
How the Business Consultant described it



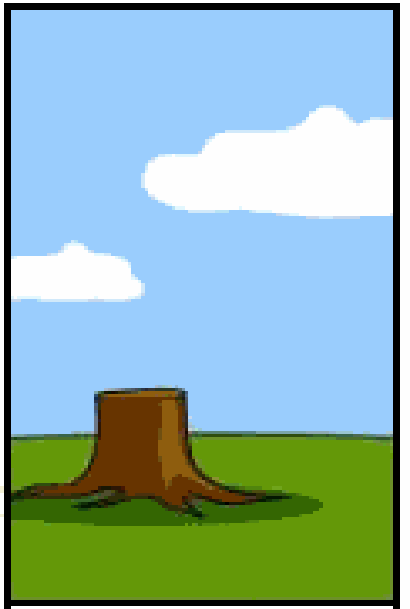
How the project was documented



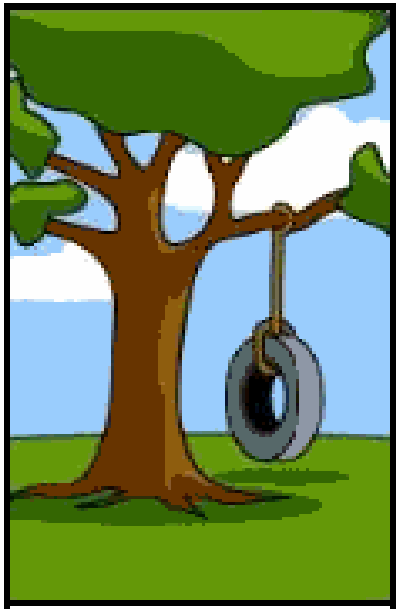
What operations installed



How the customer was billed

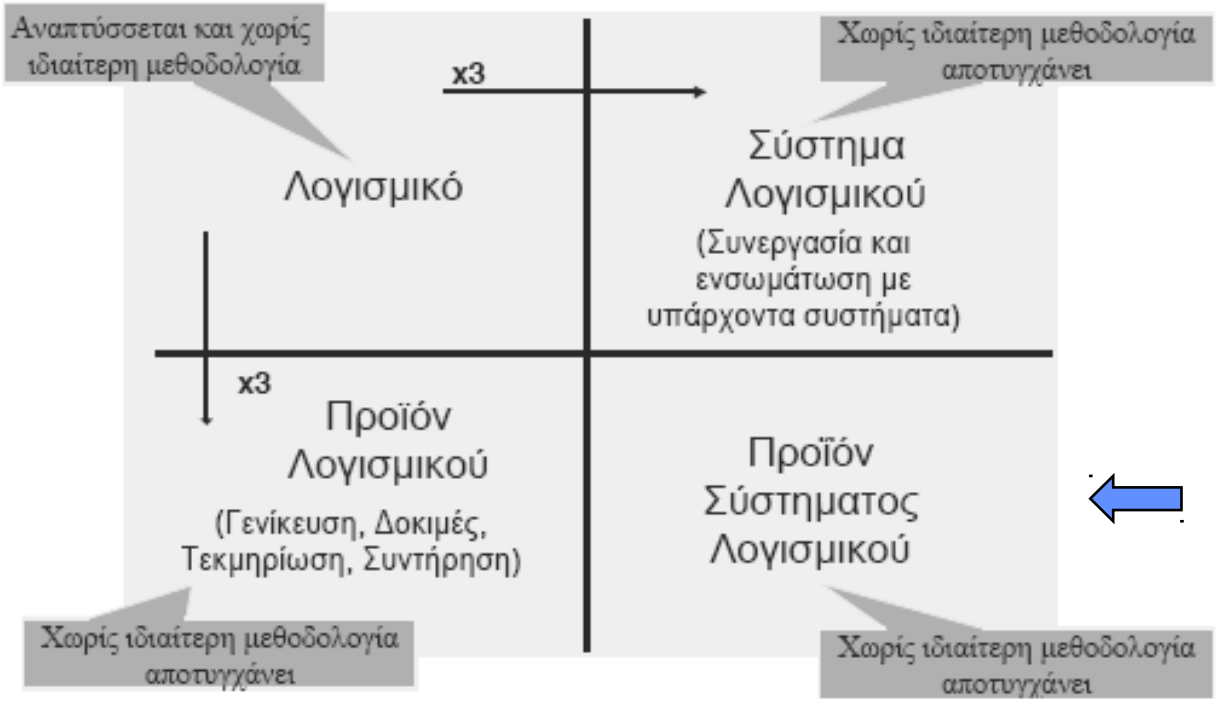


How it was supported



What the customer really needed

Λογισμικό, σύστημα λογισμικού, προϊόν λογισμικού (1/2)



Πληροφοριακά Συστήματα

Από το βιβλίο «The Mythical Mam Month», Frederick Brooks

Λογισμικό, σύστημα λογισμικού, προϊόν λογισμικού (2/2)

- Δείχνει ότι αν Χ το κόστος ανάπτυξης ενός προγράμματος τότε
  - Αν πρόκειται να μετουσιωθεί σε προϊόν τότε το κόστος είναι τριπλάσιο
  - Αν πρέπει να συνεργαστεί με υπάρχοντα προγράμματα τρίτων τότε το κόστος είναι πάλι τριπλάσιο
  - Επομένως αν πρέπει να μετουσιωθεί σε προϊόν που πρέπει να συνεργαστεί με άλλα υπάρχοντα τότε το κόστος είναι εννιαπλάσιο…
  - Είναι εύκολο να αναπτύξεις χωρίς ιδιαίτερη οργάνωση ένα πρόγραμμα αλλά αδύνατο να αναπτύξεις ένα καλό προϊόν λογισμικού που πρέπει να συνεργάζεται με άλλα προϊόντα λογισμικού.

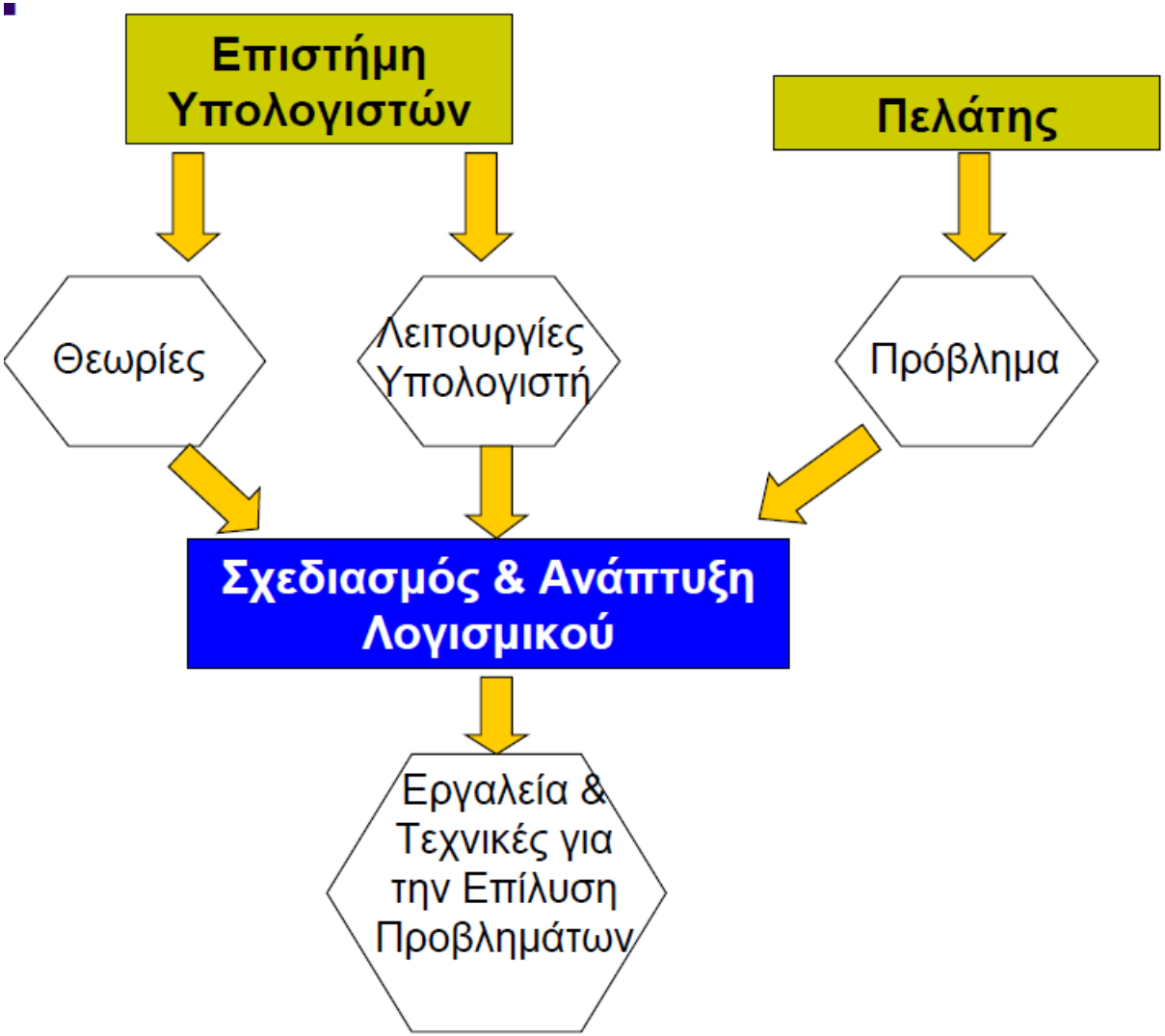
Τεχνολογία λογισμικού (1/2)

Προϊόν συστήματος λογισμικού

- Λογισμικό
  - Προγράμματα υπολογιστή
- Προϊόν λογισμικού
  - Λογισμικό που μπορεί να πουληθεί σε κάποιο πελάτη (γενικής χρήσης, κατά παραγγελία). Περιλαμβάνεται και η σχετική τεκμηρίωση όπως απαιτήσεις, σχέδιο, εγχειρίδιο.
- Σύστημα Λογισμικού
  - Επιτρέπει τη συνεργασία και την ενσωμάτωση με άλλα συστήματα λογισμικού.

Τεχνολογία Λογισμικού
είναι ο τεχνικός κλάδος που ασχολείται με όλες τις πτυχές ανάπτυξης λογισμικού από τα πρώτα στάδια της εξαγωγής προδιαγραφών μέχρι τη συντήρηση του συστήματος (Somerville)

Ο ρόλος του Τεχνολόγου Λογισμικού



Τεχνολογία Λογισμικού

- Επιλύει προβλήματα με τη βοήθεια του λογισμικού
  - ανάλυση προβλήματος
  - σύνθεση της λύσης
- Με τη χρήση:
  - Μεθόδων ή τεχνικών: διαδικασία παραγωγής αποτελέσματος
  - Εργαλείων: βοήθημα ή αυτοματοποιημένο σύστημα
  - Διαδικασιών: συνδυασμό εργαλείων και μεθόδων
  - Υποδειγμάτων: συγκεκριμένη προσέγγιση ή μεθοδολογία

- **Η Τεχνολογία Λογισμικού έχει ως στόχο την σχεδίαση και υλοποίηση λογισμικού υψηλής ποιότητας.**

Αληθινές ιστορίες…

- Σφάλματα:
  - Ενοχλητικά
  - Κόστος σε χρόνο και χρήμα
  - Ζωτικής σημασίας
- 1992, σύστημα ασθενοφόρων Λονδίνου
- 1993, διαχείριση αποσκευών αεροδρ. Ντένβερ, 175,6 εκατ. $
- 1996, πύραυλος Ariane-5, 350 εκατ. $
- 1970, 2004, Διαχείριση υποθέσεων FBI, 2005 ακύρωση
- 2004, κέντρο ελέγχου εναέριας κυκλοφορίας του Λος Αντζελες
- Ερευνα 1993-2000 κατέγραψε ότι το 2000:
  - 23% των έργων ματαιώθηκαν
  - 28% ολοκληρώθηκαν σύμφωνα με τον χρονικό και οικονομικό προγραμματισμό
  - 49% ολοκληρώθηκαν με αποκλίσεις

Συχνότερες αιτίες αποτυχίας

- Μη ρεαλιστικοί στόχοι/ στόχοι που δεν είναι καλά καθορισμένοι
- Λάθος ορισμός απαιτήσεων
- Ανικανότητα χειρισμού πολυπλοκότητας έργων
- Κακές εκτιμήσεις για τους απαιτούμενους πόρους
- Κακή αναφορά της κατάστασης του έργου
- Κακή διαχείριση του έργου
- Κακή διαχείριση κινδύνων
- Ελλιπής επικοινωνία πελάτη, προγραμματιστή, χρήστη
- Κακές προγραμματιστικές τεχνικές
- Χρήση ανώριμης τεχνολογίας

Παράγοντες επιτυχίας

Έρευνες έχουν δείξει ότι για την επιτυχή ολοκλήρωση ενός έργου…

- **Επιχειρησιακή υποστήριξη**
  - **Στάση διοίκησης**
  - **Καθορισμός στόχων και εμβέλειας**
- **Εμπλοκή χρηστών**
  - **Συμμετοχή σε όλη την πορεία του έργου**
- **Ικανός διοικητής έργου**
- **Σαφείς επιχειρησιακοί στόχοι**
- **Εστιασμένο πεδίο εφαρμογής του προϊόντος**
  - **Κοινή συμφωνία οριοθέτησης της λειτουργικότητας σε συνδυασμό με εφικτότητα χρονοδιαγράμματος και προϋπολογισμένου κόστους**
- **Πρότυπες υποδομές**
  - **«κώδικας υποδομής»**
  - **Χρήση έτοιμων, διαδεδομένων ώριμων και δοκιμασμένων υποδομών, τεχνολογιών και εργαλείων**

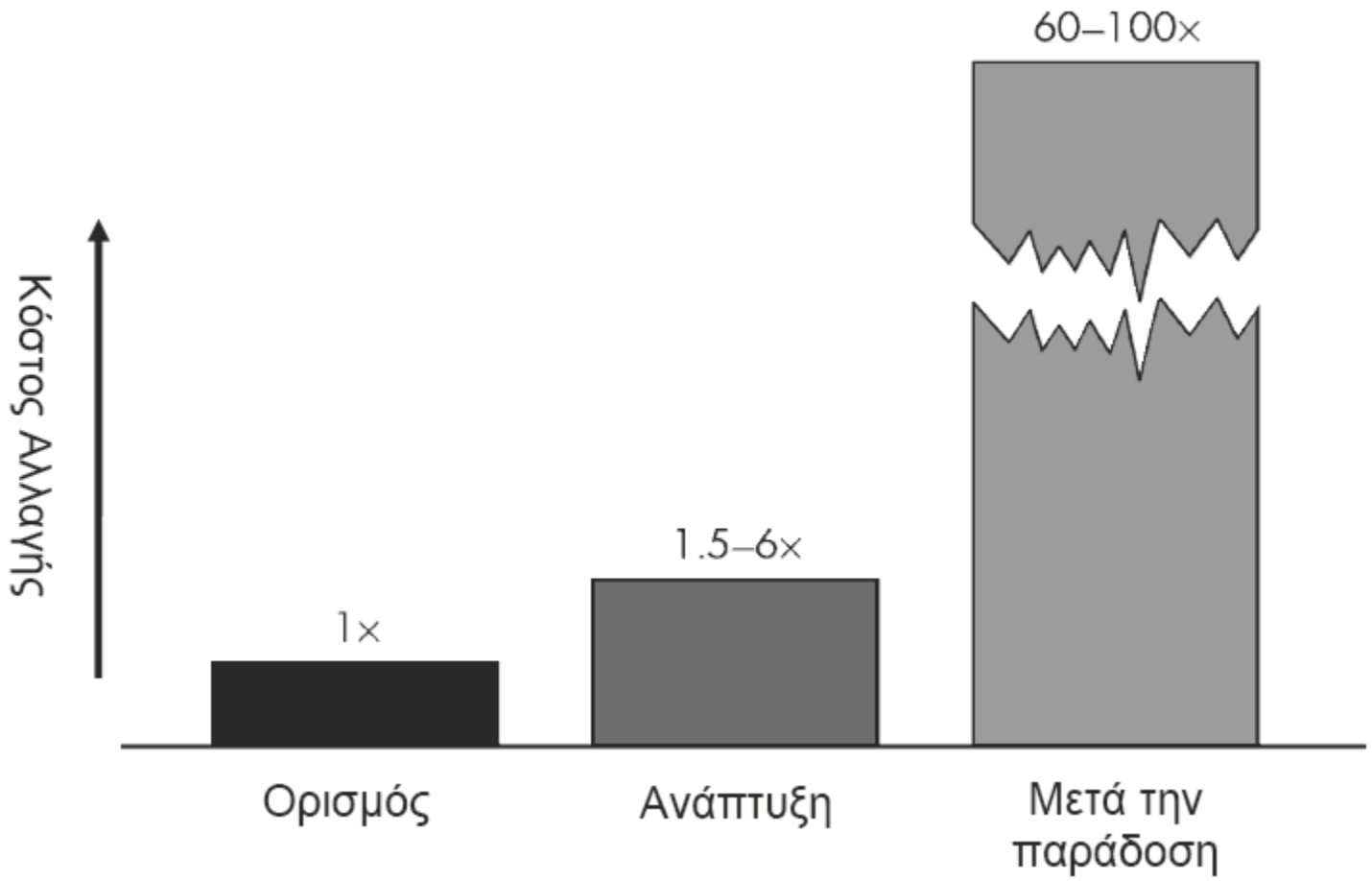
Μύθοι σχετικά με το λογισμικό (1/3)

- **Μύθοι του management**
  - **Υπάρχουν στην εταιρία πρότυπα και διαδικασίες για την ανάπτυξη λογισμικού. Άρα παρέχονται στους developers όλα όσα χρειάζονται να ξέρουν.**
  - **Η εταιρία διαθέτει «state of the art » συστήματα ανάπτυξης λογισμικού και hardware.**
  - **Αν ένα έργο ξεφύγει από το χρονοπρογραμματισμό το επαναφέρουμε προσθέτοντας ανθρώπινους πόρους.**

Μύθοι σχετικά με το λογισμικό (2/3)

- **Μύθοι του πελάτη**
  - **Μια γενική διατύπωση των στόχων είναι επαρκής για την αρχή συγγραφής των προγραμμάτων. Οι λεπτομέρειες μπορούν να καθοριστούν αργότερα.**
  - **Οι διαρκώς μεταλλασσόμενες απαιτήσεις ενός έργου μπορούν να αντιμετωπισθούν εύκολα λόγω της ευελιξίας που διαθέτει το λογισμικό.**

Κόστος αλλαγών



Μύθοι σχετικά με το λογισμικό (3/3)

- **Μύθοι του προγραμματιστή**
  - **Το έργο του προγραμματιστή τελειώνει τη στιγμή που θα εκτελεστεί το πρόγραμμα**
  - **Δεν είναι δυνατό να εκτιμηθεί η ποιότητα έως ότου εκτελεστεί το πρόγραμμα**
  - **Το μόνο παραδοτέο έως ότου εκτελεστεί το πρόγραμμα είναι ο κώδικας που τρέχει.**

Μα.. τα δικά μας προγράμματα δουλεύουν…

- **Οι φοιτητές λένε συχνά «εμείς κάνουμε προγράμματα χωρίς να ακολουθούμε τίποτα από όσα λέει η τεχνολογία και δουλεύουν»**

- ***Επομένως για ποιο λόγο να μπει κανείς στην επίπονη διαδικασία της σχεδίασης και των ελέγχων ???***

Τι είναι ένα μοντέλο διεργασίας λογισμικού?

- Μια απλοποιημένη αναπαράσταση της διεργασίας ανάπτυξης λογισμικού παρουσιασμένη με μια συγκεκριμένη οπτική.
- Παραδείγματα οπτικών διεργασίας αποτελούν:
  - Οπτική ροής εργασιών: σειρά δραστηριοτήτων
  - Οπτικής ροής δεδομένων: ροή πληροφοριών
  - Οπτική ρόλου/ δράσης: ποιος κάνει τι
- Γενικά μοντέλα
  - Καταρράκτης
  - Επαναληπτική ανάπτυξη
  - Βασισμένα σε συνιστώσες

Η διεργασία ανάπτυξης λογισμικού

Οικονομικά Θέματα

- Αξίζει να θέσουμε σε εφαρμογή μια νέα μέθοδο κωδικοποίησης ΜΕΚ2, η οποία είναι10% γρηγορότερη από τη μέθοδο ΜΕΚ1 που εφαρμόζουμε ως σήμερα;
  - Κοινή λογική: Φυσικά!
  - Τεχνολόγος Λογισμικού: Τι επιπτώσεις θα έχει μια τέτοια απόφαση σε θέματα συντήρησης του λογισμικού;

Συντήρηση λογισμικού

- Κύκλος Ζωής Λογισμικού (Software Life Cycle):
  - Από τη σύλληψη του έργου μέχρι την απόσυρσή του.
- Ο τρόπος που παράγουμε λογισμικό περιλαμβάνει:
  - Μοντέλο κύκλου ζωής
  - Ανθρώπινο δυναμικό (managers, experts, programmers, …)
  - Εργαλεία CASE
- Φάσεις κύκλου ζωής:
  1. *Απαιτήσεων (Requirements):* Εξαγωγή από ανάγκες πελατών.
  2. *Ορισμού Προδιαγραφών (Specification):* Τι πρέπει το προϊόν να κάνει (specification doc, software project management plan).
  3. *Σχεδιασμού (Design):* Πώς θα το κάνει (architectural and detailed design).

Συντήρηση λογισμικού

4. *Υλοποίησης (Implementation):* **Κώδικας και έλεγχος.**
5. *Συνένωσης Κώδικα (Integration):* **Συνένωση ανεξάρτητων τμημάτων και συνολικός έλεγχος (δημιουργών και πελάτη – acceptance test).**
6. *Συντήρησης (Maintenance):* **Αλλαγές στο προϊόν μετά την αποδοχή του από πελάτη:**
   - **Διορθωτική (corrective or repair) – 17.5%**
     - **Ενισχυτική: Τελειοποίησης(perfective) – 60.5%**
     - **Προσαρμογής(adaptive) –18%**
7. *Απόσυρσης (Retirement)* **προϊόντος από λειτουργία.**

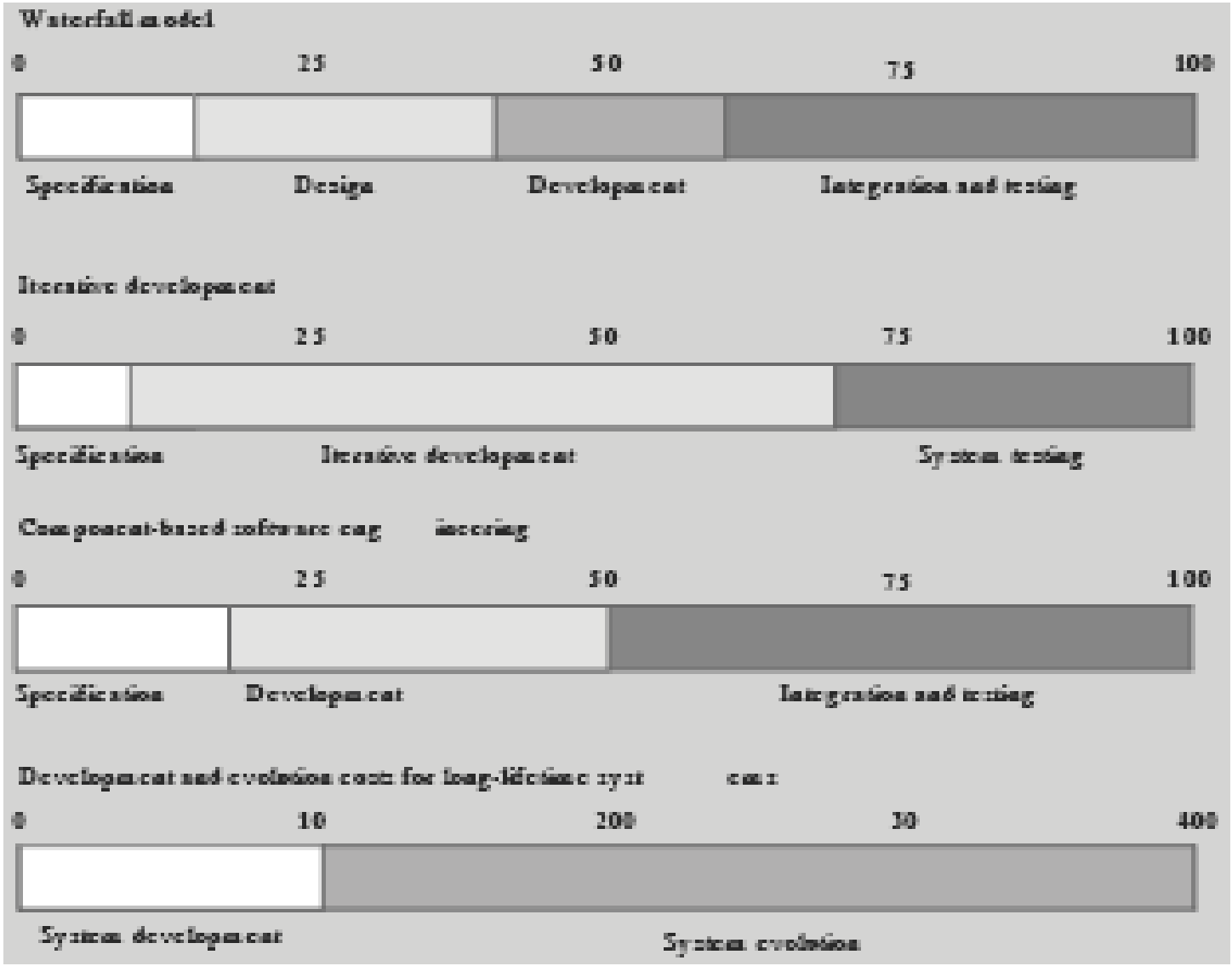
"Καλό"είναι το λογισμικό που συντηρείται—το"κακό"πετιέται

Επιπτώσεις από την εφαρμογή της MEK2 στη φάση της συντήρησης του λογισμικού

Ποιο είναι το κόστος της Τεχνολογίας Λογισμικού

- 60% κόστος ανάπτυξης, 40% κόστος ελέγχου. Συχνά το κόστος εξέλιξης ξεπερνά το κόστος ανάπτυξης
- Το κόστος διαφέρει ανάλογα με :
  - Τον τύπο του συστήματος
  - Τις απαιτήσεις του συστήματος
    - Απόδοση
    - Αξιοπιστία
- Η κατανομή του κόστους διαφέρει ανάλογα με το μοντέλο ανάπτυξης που επιλέγεται.

Κατανομή κόστους στις δραστηριότητες



Θέματα ποιότητας (1/5)

Καλό λογισμικό → ποιότητα

Ποιότητα:

- **Προϊόντος**
- **Διεργασίας**
- **Σε σχέση με το επιχειρηματικό περιβάλλον**

Θέματα ποιότητας (2/5)

- Άποψη χρηστών με βάση τη λειτουργικότητα, τις αστοχίες, την ευκολία χρήσης.
- Άποψη δημιουργών με βάση τα εσωτερικά χαρακτηριστικά.
- Μοντέλα συσχέτισης της άποψης των χρηστών με αυτή των δημιουργών

Θέματα ποιότητας (3/5)

Ποιότητα προϊόντος

1. Θα συναντήσουμε σφάλματα ?
   - Που και πότε?
   - Πως θα τα εντοπίσουμε?
   - Πως θα αποφύγουμε την εξέλιξη του σφάλματος σε αστοχία?

Ποιότητα διεργασίας

1. Πως θα γίνει αποτελεσματικότερη η διεργασία?
   - Capability Maturity Model (CMM)
   - ISO 9000
   - Software Process Improvement and Capability dEtermination (SPICE)

Θέματα ποιότητας (4/5)

Σε σχέση με το επιχειρηματικό περιβάλλον

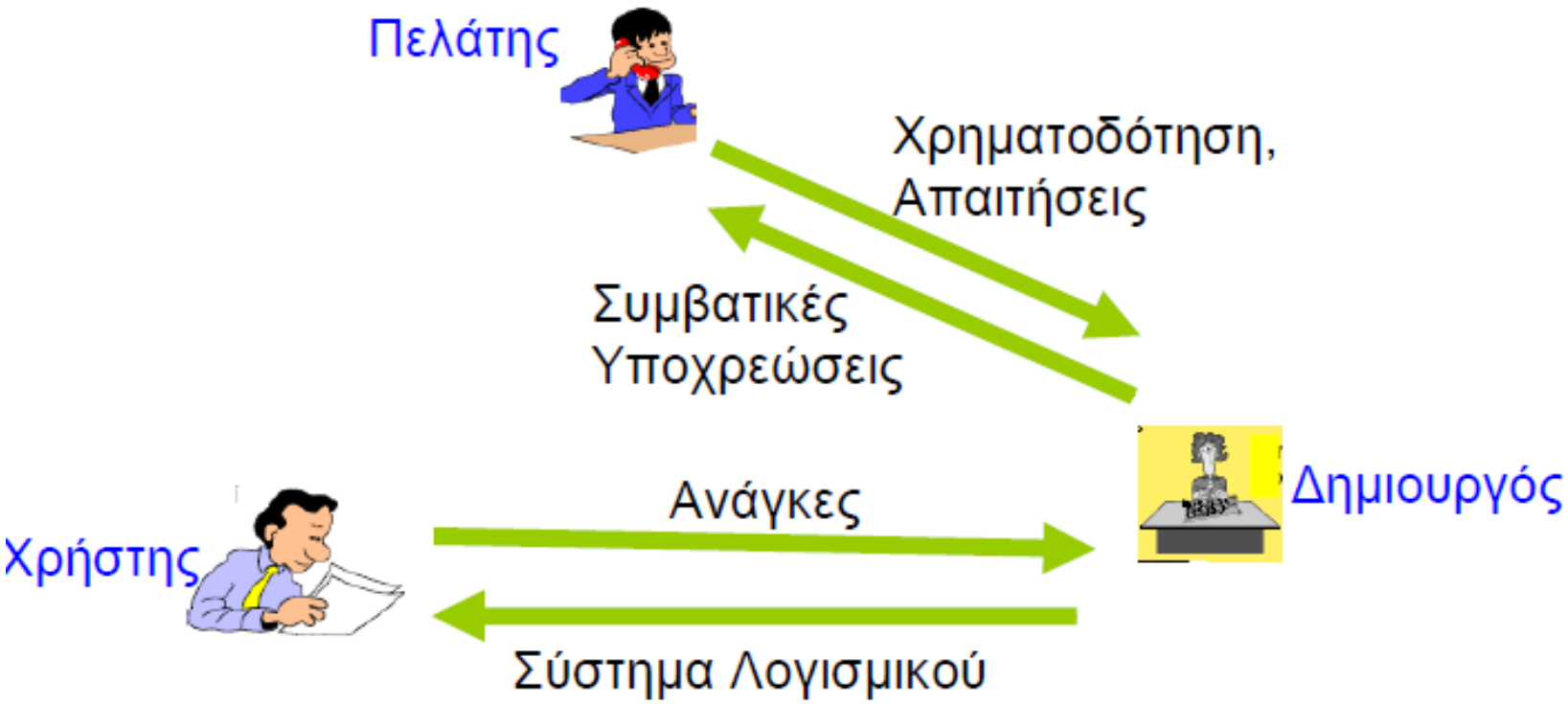
- Η ποιότητα εκτιμάται βάσει των υπηρεσιών και των προϊόντων που παρέχει η εταιρεία στην οποία εγκαθίσταται το λογισμικό.

- Απόδοση επένδυσης (Return On Investment, ROI)
  - Το χρηματικό κόστος (κυβερνήσεις)
  - Την απαιτούμενη προσπάθεια (εταιρείες)
    - Εκπαίδευση
    - Χρονοδιάγραμμα
    - Κίνδυνος
    - Παραγωγικότητα
    - Ποιότητα

Θέματα ποιότητας (5/5)

- Ελάττωμα/ σφάλμα
  - Αφορά την εσωτερική συμπεριφορά του συστήματος
- Αστοχία/ δυσλειτουργία
  - Αφορούν την εξωτερική συμπεριφορά του συστήματος

Συμμετέχοντες σε ένα έργο λογισμικού



Πελάτης

- Καθορίζει τι θα κατασκευαστεί.
- Παρέχει προδιαγραφές των απαιτήσεων.
- Χρηματοδοτεί την ανάπτυξη.
- Παραλαμβάνει και αξιολογεί το τελικό προϊόν.

Χρήστης- Διαχειριστής

- Σύνδεσμος μεταξύ πελάτη – δημιουργού.
- Διαπραγματεύεται χρόνο παράδοσης και κόστος.
- Χρονοπρογραμματίζει και επιβλέπει το έργο.
- Θέτει περιορισμούς στο χρόνο και στην προσπάθεια στο δημιουργό.

Δημιουργός

- Καθορίζει πώς θα κατασκευαστεί το προϊόν.
- Δημιουργεί το προϊόν (λογισμικό).
- Προσπαθεί για την ικανοποίηση του πελάτη.

Σχέσεις μεταξύ συμμετεχόντων

- Απαραίτητη επικοινωνία, ευελιξία και αμοιβαία κατανόηση.
- Πρόβλημα: Αβεβαιότητα.
- Επικάλυψη ρόλων κατά την πρόοδο του έργου

Η ομάδα ανάπτυξης