



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τεχνολογία Λογισμικού, 7ο/9ο εξάμηνο 2018-2019

# Τεχνολογία Λογισμικού

Ν. Παπασπύρου, Αν. Καθ. ΣΗΜΜΥ, [nickie@softlab.ntua.gr](mailto:nickie@softlab.ntua.gr)

Β. Βεσκούκης, Αν. Καθ. ΣΑΤΜ, [v.vescoukis@cs.ntua.gr](mailto:v.vescoukis@cs.ntua.gr)

Κ. Σαΐδης, ΠΔ 407, [saiko@softlab.ntua.gr](mailto:saiko@softlab.ntua.gr)

# Καλώς ήλθατε!

## Β.Βεσκούκης

- Δρ. ΗΜΜΥ (1995 ☹ )
- Αν.Καθ., Σχολή ΑΤΜ "Τεχνολογία Λογισμικού στη Γεωπληροφορική"
- ΕΤΗ Zurich (visiting professor, 2014)
- Groningen (NL), (visiting professor 2015)
- Ελληνικό Ανοικτό Πανεπιστήμιο, ΠΛΗ, ΠΛΣ
- Πανεπιστήμιο Πειραιά
  
- [users.ntua.gr/vvesk](http://users.ntua.gr/vvesk), [www.linkedin.com/in/vvescoukis/](http://www.linkedin.com/in/vvescoukis/)

Hello!



How the customer explained it



How the project leader understood it



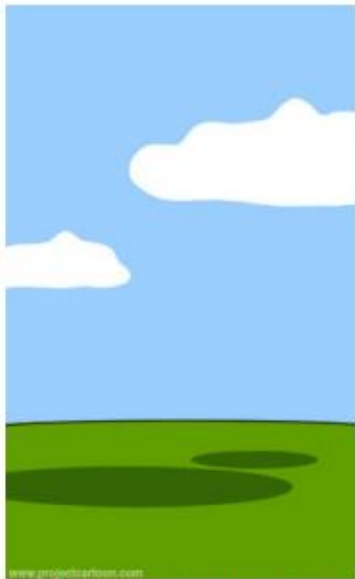
How the analyst designed it



How the programmer wrote it



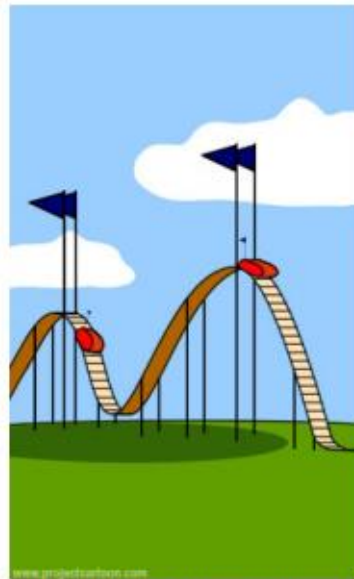
How the business consultant described it



How the project was documented



What operations installed



How the customer was billed



How it was supported



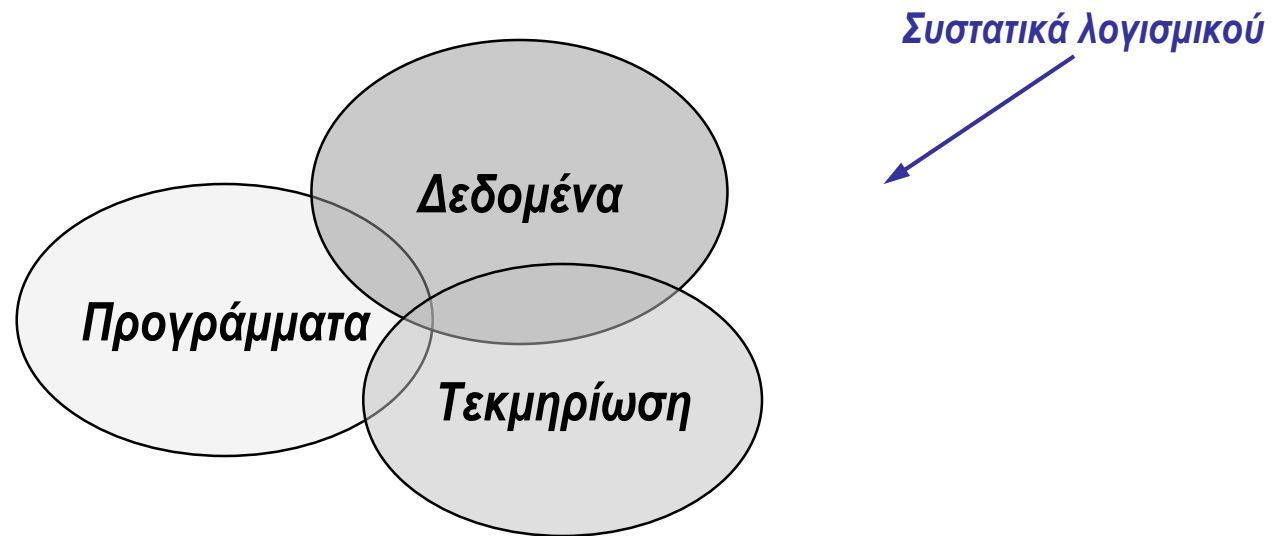
What the customer really needed

# Τι είναι Λογισμικό;

Προγράμματα ηλεκτρονικού υπολογιστή

Δομές δεδομένων με μόνιμη ή προσωρινή αποθήκευση

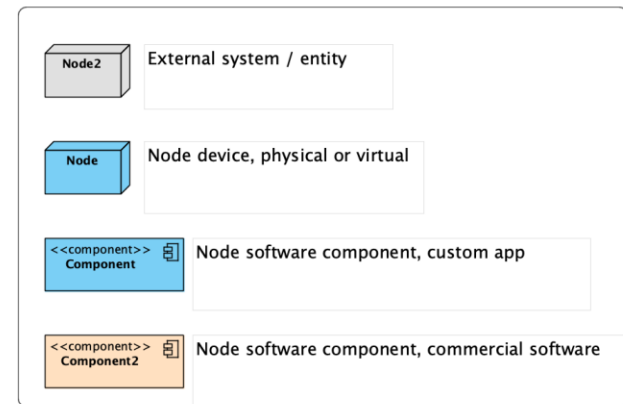
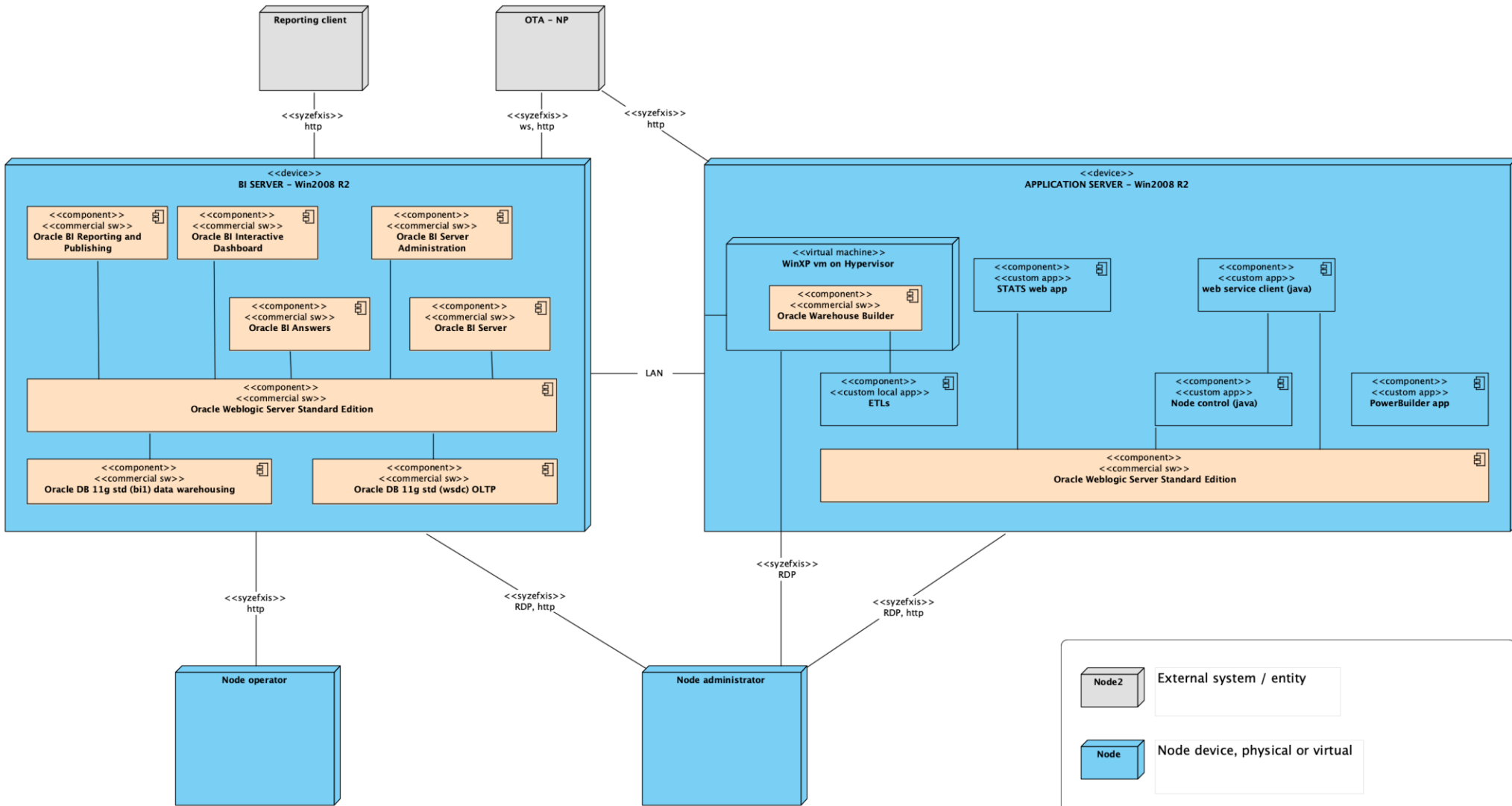
Τεκμηρίωση σχετικά με τα παραπάνω



# Συστατικά Λογισμικού

Software {components, elements, artifacts} είναι ΌΛΑ τα παρακάτω:

- Πηγαίος κώδικας
- Προδιαγραφές, εκθέσεις, αναφορές, κείμενα γενικά
- Σχέδια
- Διαγράμματα
- Ο,τιδήποτε υποστηρίζει / τεκμηριώνει / συμπληρώνει τα παραπάνω



# Τι αφορά η Τεχνολογία λογισμικού;

## Τι ΔΕΝ αφορά

- Μικρές εφαρμογές λογισμικού για χρήση σε μη-"κρίσιμες αποστολές"
- Εργασίες σε μαθήματα προγραμματισμού (σε γενικές γραμμές)
- Ο,τιδήποτε φαίνεται "βατό" (ή ενός περιορισμών) για υλοποίηση με τη λογική "Ανάλυση-> προγραμματισμός"

## Τι αφορά

- Λογισμικό μεγάλης κλίμακας
- Εφαρμογές για "κρίσιμες αποστολές"



# Τεχνολογία λογισμικού:

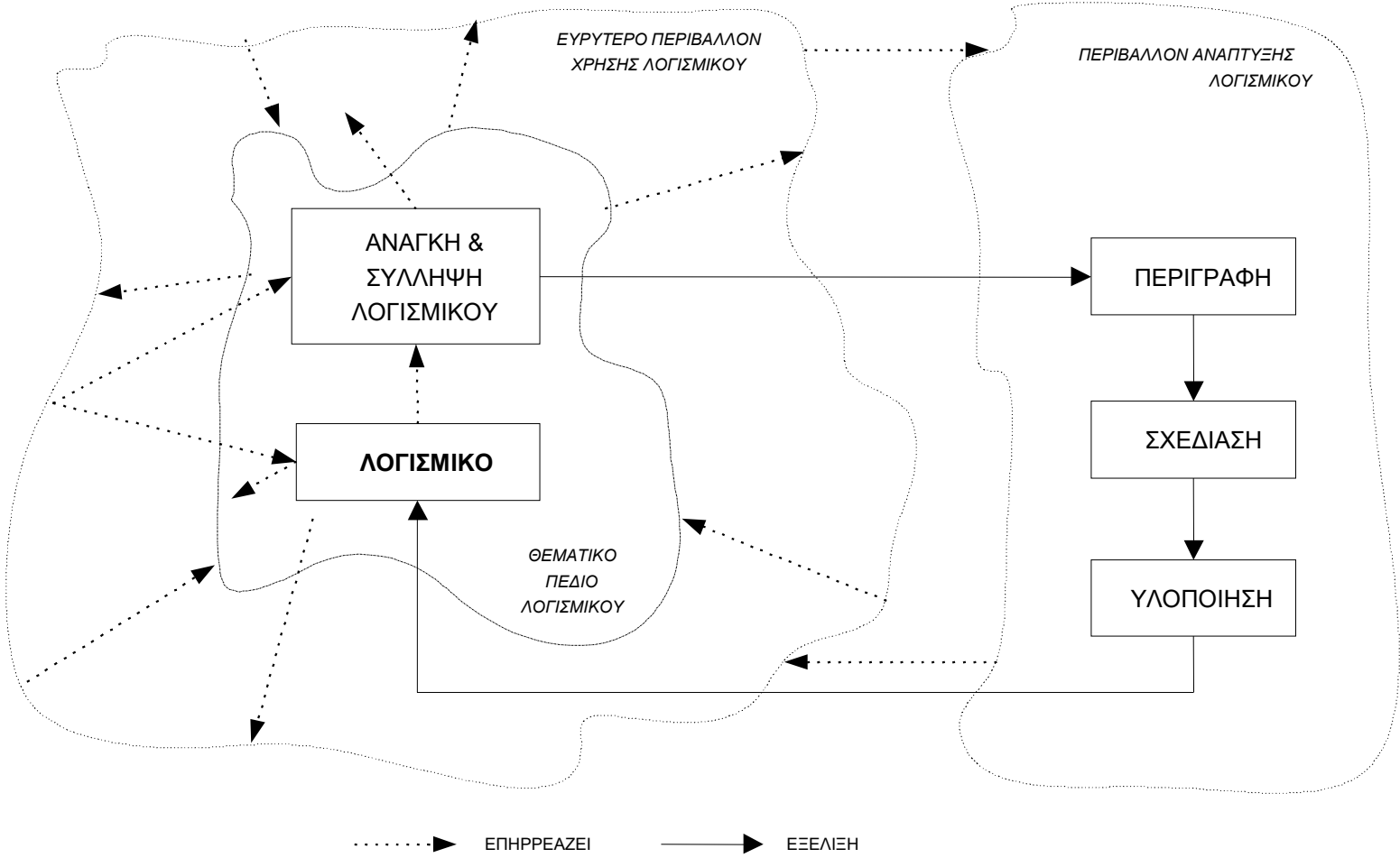
## Τεχνολογία: «Λόγος περί Τέχνης»

- Περιγραφή του τρόπου (τέχνης!) υλοποίησης τεχνικών κατασκευών, εν προκειμένω του Λογισμικού.

## Λογισμικό και άλλες τεχνικές κατασκευές

- Τεχνικές κατασκευές
  - Έχουν υλική υπόσταση
  - Περιγράφονται και γίνονται αντιληπτές με μοναδικό τρόπο
- Το Λογισμικό είναι μια ιδεατή οντότητα: δεν υπάρχει με την έννοια της ύλης
  - Γίνεται αντιληπτό από τα αποτελέσματα της χρήσης του
  - Δεν περιγράφεται εύκολα και μοναδικά

# Το λογισμικό στο περιβάλλον του



# Περί τίνος πρόκειται;

**Όπως ακριβώς** στις τεχνικές κατασκευές (γέφυρες, δρόμοι κτίρια, κλπ) απαιτούμε την τεκμηριωμένη σχεδιάσή τους πριν την κατασκευή τους,

- Αρχιτεκτονικά, στατικά, υδραυλικά, ηλεκτρικά, κλπ σχέδια
- Προδιαγραφές αισθητικής, ασφάλειας, επικοινωνίας, καταλληλότητας
- Κανονισμοί, νόμοι, διατάξεις

Σχεδίαση: προσδιορισμός των συστατικών στοιχείων μιας κατασκευής, των συσχετίσεων μεταξύ αυτών, και του τρόπου χρήσης κάθε συστατικού

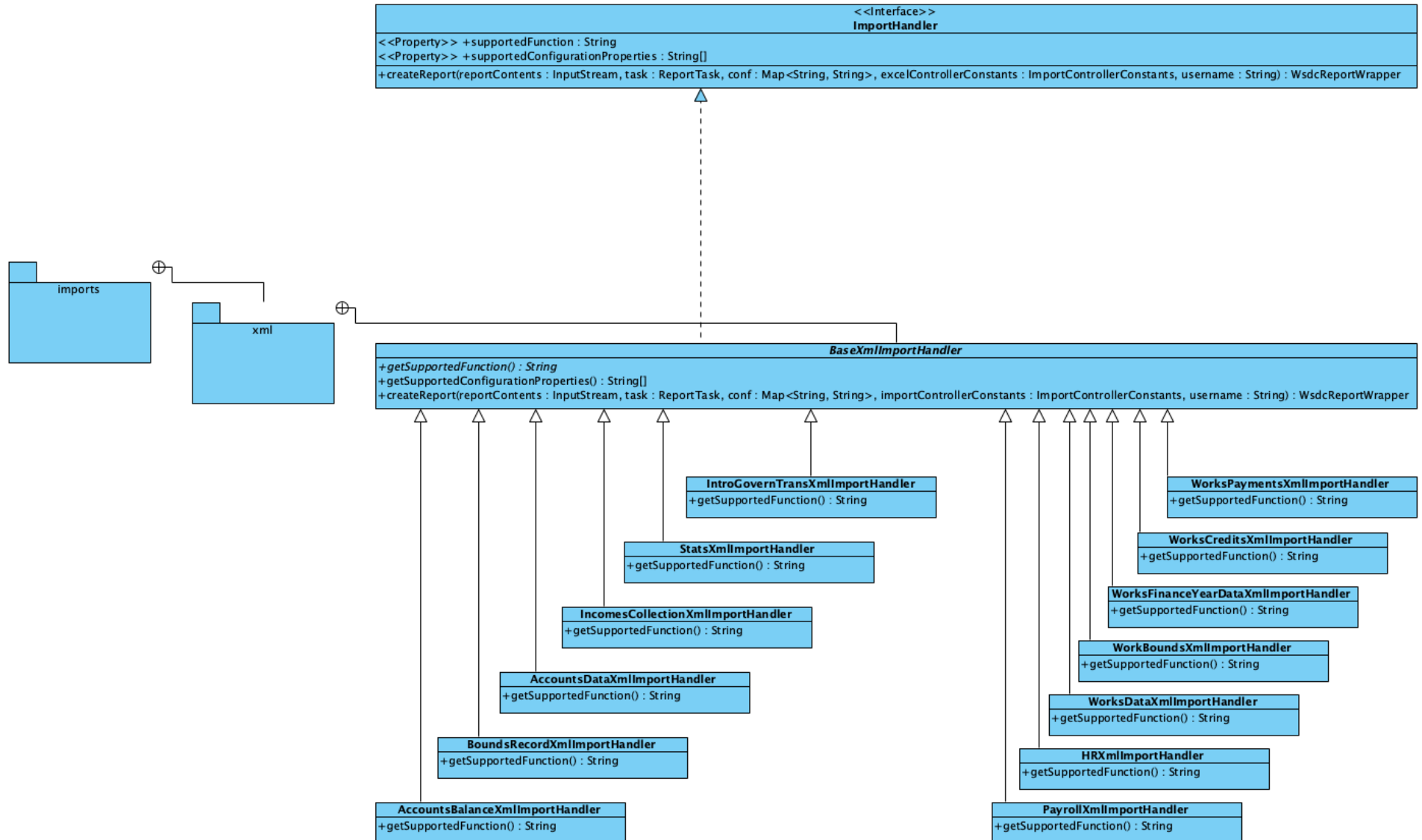
...

Ετσι και στο λογισμικό:

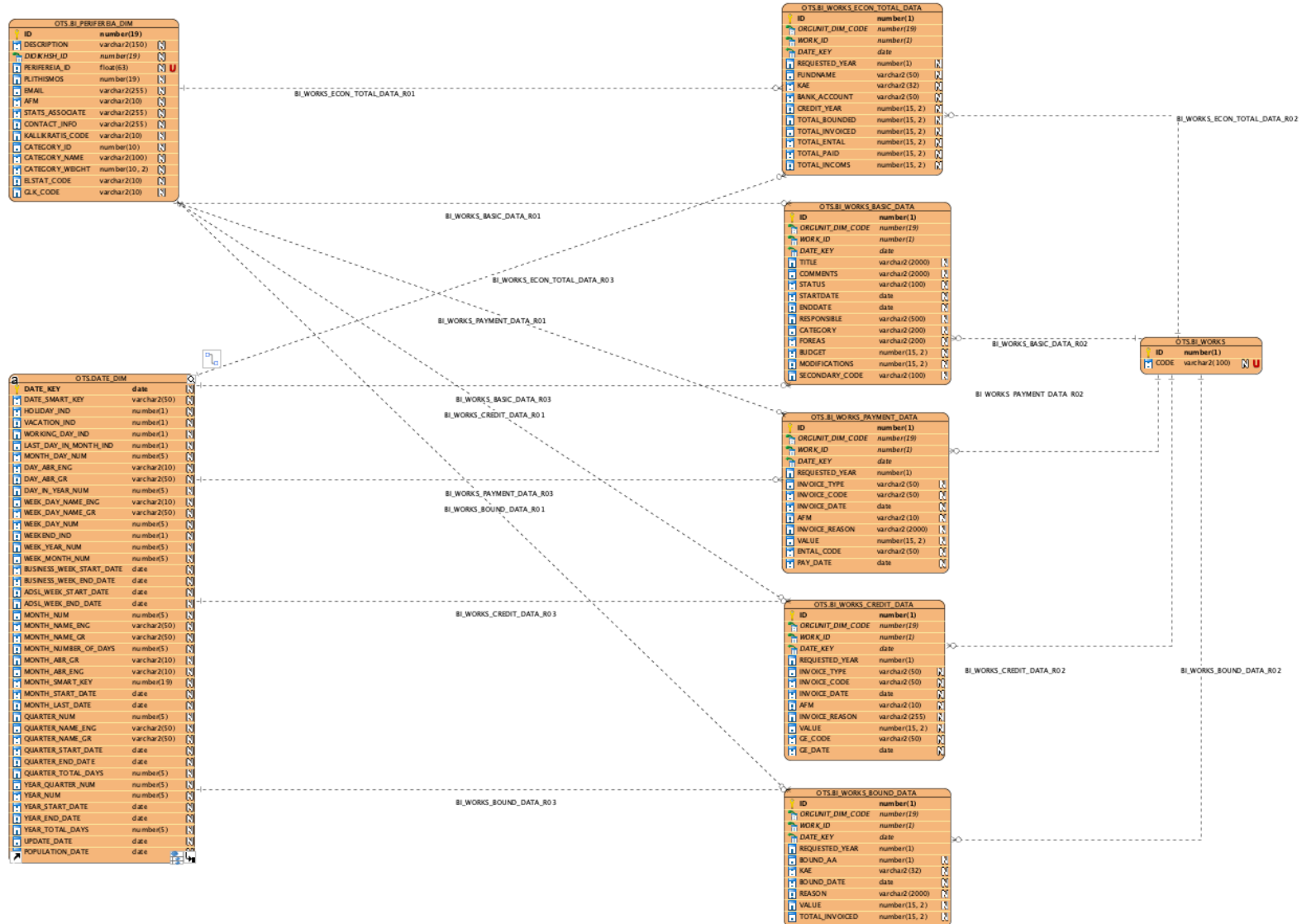
επιδιώκουμε τον προσδιορισμό των στοιχείων που (πρέπει να) το αποτελούν, ώστε να το κατασκευάσουμε «σωστά»

- Συναρτήσεις, διαδικασίες, κλάσεις, υπορουτίνες, modules, πακέτα, εντολές, δεδομένα, κλπ
- Χωρίς δυνατότητα αντίληψης της ίδιας της κατασκευής δια των αισθήσεων, αλλά μόνο δια του αποτελέσματος...
- Εντός χρόνου, κόστους και ποιότητας

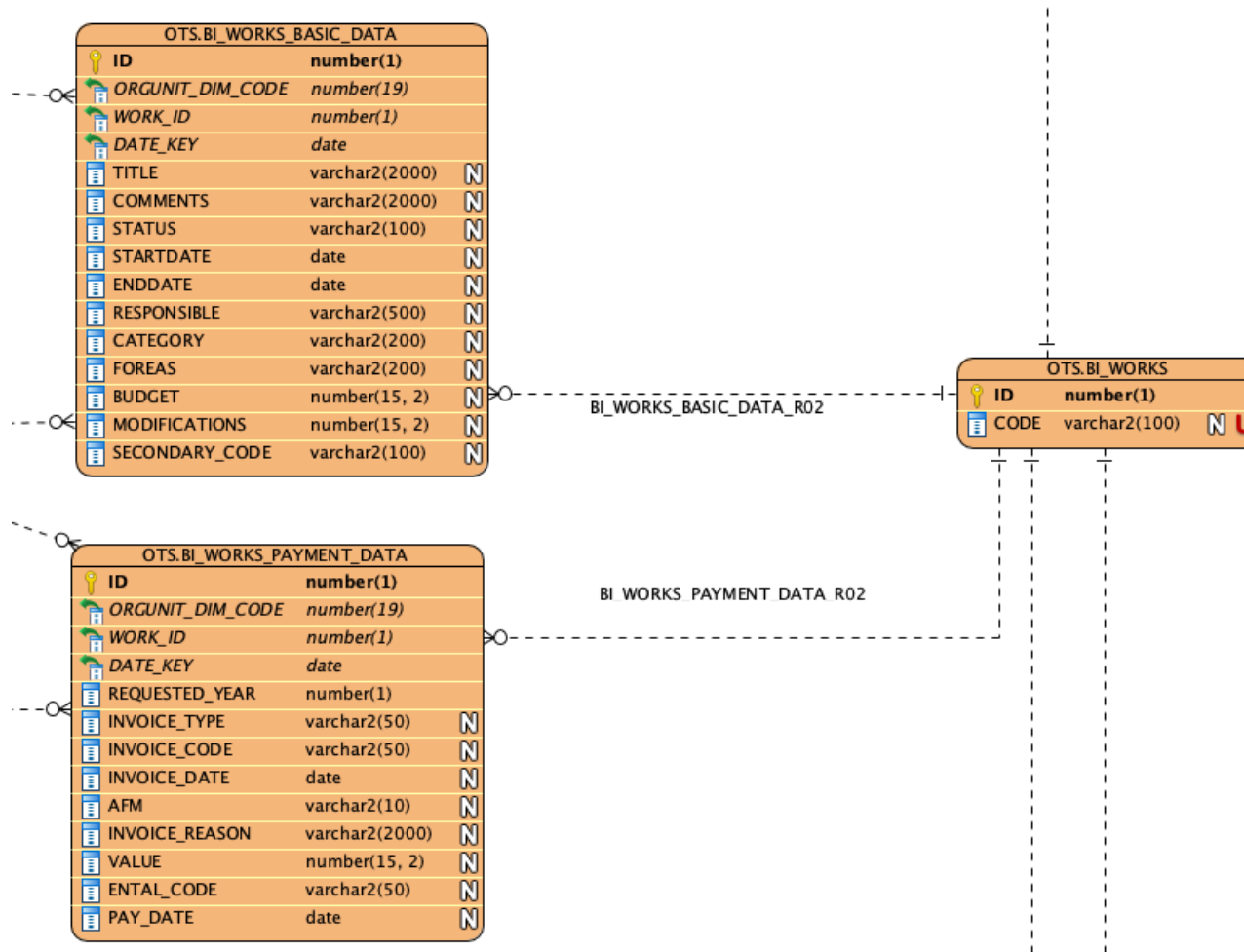
# "...προσδιορισμός των στοιχείων του λογισμικού"



# "...προσδιορισμός των στοιχείων του λογισμικού"



"...προσδιορισμός των στοιχείων του λογισμικού"



# Κρίση λογισμικού (όρος του 1968!)

<ul style="list-style-type: none"><li>• <b>Εξαιρετικά δύσκολη διαδικασία κατασκευής</b></li></ul>	<i>Δεν είναι πάντα σαφές ποια βήματα πρέπει να γίνουν, με ποια σειρά, με ποια ενδιάμεσα προϊόντα κλπ</i>
<ul style="list-style-type: none"><li>• <b>Ανεπαρκής ή και κακή ποιότητα τελικού προϊόντος</b></li></ul>	<i>Λάθη στην κατασκευή, μη ικανοποίηση του σκοπού</i>
<ul style="list-style-type: none"><li>• <b>Μη τήρηση χρονοδιαγραμμάτων</b></li></ul>	<i>Υπερβολικές και "αδικοιολόγητες" καθυστερήσεις</i>
<ul style="list-style-type: none"><li>• <b>Υπερβάσεις προϋπολογισμών</b></li></ul>	<i>Κακές αρχικές εκτιμήσεις κόστους. Τελικά προϊόντα με πολλαπλάσιο κόστος από το αρχικά προϋπολογισθέν</i>
<ul style="list-style-type: none"><li>• <b>Μεγάλη δυσκολία και συνεπαγόμενο κόστος συντήρησης</b></li></ul>	<i>Παρενέργειες μεταβολών σε στοιχεία που πριν λειτουργούσαν, πρόχειρες λύσεις</i>
<ul style="list-style-type: none"><li>• <b>Δύσκολη κατανόηση εγγράφων, σχεδίων κλπ από διαφορετικούς κατασκευαστές</b></li></ul>	<i>Στην πράξη η κατανόηση ενός συστήματος λογισμικού από τρίτους πλην των κατασκευαστών του, είναι συχνά αδύνατη ή ιδιαίτερα ασύμφορη</i>



# Τις πταίει;

Πολλοί συμμετέχοντες, με αντικρουόμενα συμφέροντα

- Χρήστες
- Πελάτες
- Διοίκηση (ακόμα και Μέτοχοι ή Επενδυτές)
- Προγραμματιστές
- Δοκιμαστές
- Σχεδιαστές διεπαφών
- Υπεύθυνοι έργου
- Κ.ά.

## Τις πταίει;

Κάθε ένας από τους συμμετέχοντες έχει διαφορετικές εμπειρίες, βιώματα, αφετηρίες, στόχους και επιδιώξεις από την υλοποίηση του (ίδιου!) έργου.

Είναι δύσκολο να επιτευχθεί ένας κοινός τόπος, σωστά ιεραρχημένος για το όφελος του έργου και, συνάμα, ικανοποιητικός για όλους τους συμμετέχοντες.

Πολυπλοκότητα!

Robert L. Glass, "Facts and Fallacies of Software Engineering"

- Most software estimates are performed at the beginning of the life cycle. This makes sense until we realize that estimates are obtained before the requirements are defined and thus before the problem is understood. Estimation, therefore, usually occurs at the wrong time.

Τεχνολογία Λογισμικού

# Αντικείμενα ενασχόλησης της ΤΛ

Να περιγράψουμε

- Τι πρέπει να κάνουμε, Πώς να το κάνουμε
- Τι θα παράξουμε, Πώς θα το περιγράψουμε

Στην πορεία...

- Από τη σύλληψη - απαίτηση κατασκευής
- Έως τη χρήση του λογισμικού

Μεθοδολογικά εργαλεία δουλειάς, μακρο- και μικρο- οργάνωσης:

- εργασίες, μοντέλα κύκλου ζωής,
- πρότυπα, προδιαγραφές,
- σχέδια, μοντέλα παράστασης λογισμικού, κλπ

# Τεχνολογία λογισμικού

## Δι-επιστημονική περιοχή

- Προγραμματισμός
- Υλικό
- Δίκτυα
- User interfaces
- Βάσεις δεδομένων
- Τεχνολογία πληροφοριών γενικότερα

## ΑΛΛΑ και πεδίο επιχειρηματικής δραστηριότητας

- Ανταγωνισμός
- Κόστος παραγωγής
- Τεχνογνωσία

# Η ανάγκη για «καλό λογισμικό»

Είναι αυτονόητη και επιτακτική διότι...

Ο ρόλος των υπολογιστών...

- Στην οικονομία και την παραγωγή
- Στην ενημέρωση και την ψυχαγωγία
- Στην εκπαίδευση και αλλού

...εκπληρώνεται μόνο με Λογισμικό το οποίο:

- Κάνει **σωστά** τη δουλειά του,
- ... παράγεται με **λογικό** κόστος,
- ... σε **λογικό** χρόνο,
- ... και είναι καλής **ποιότητας**

# Αντικείμενο της τεχνολογίας λογισμικού είναι...

Η επιστημονική θεμελίωση...

- Του κύκλου ζωής
- Της διαδικασίας παραγωγής
- Του τρόπου περιγραφής
- Του τρόπου συντήρησης
- Του οργανωτικού πλαισίου χρήσης

...του Λογισμικού

Οι εργασίες αυτές είναι ιδιαίτερα δύσκολες:

- Από τη φύση και την πολυπλοκότητα του Λογισμικού
- Από την υπόστασή του ως προϊόν

# Αντικείμενο της τεχνολογίας λογισμικού είναι...

Η αναζήτηση συστηματικών τρόπων για να αποκαλύπτουμε

- Ποια συστατικά στοιχεία λογισμικού πρέπει να κατασκευάσουμε, για ποιο λόγο, με ποια σειρά, πώς να τα περιγράψουμε, κ.ά.,

ώστε να...

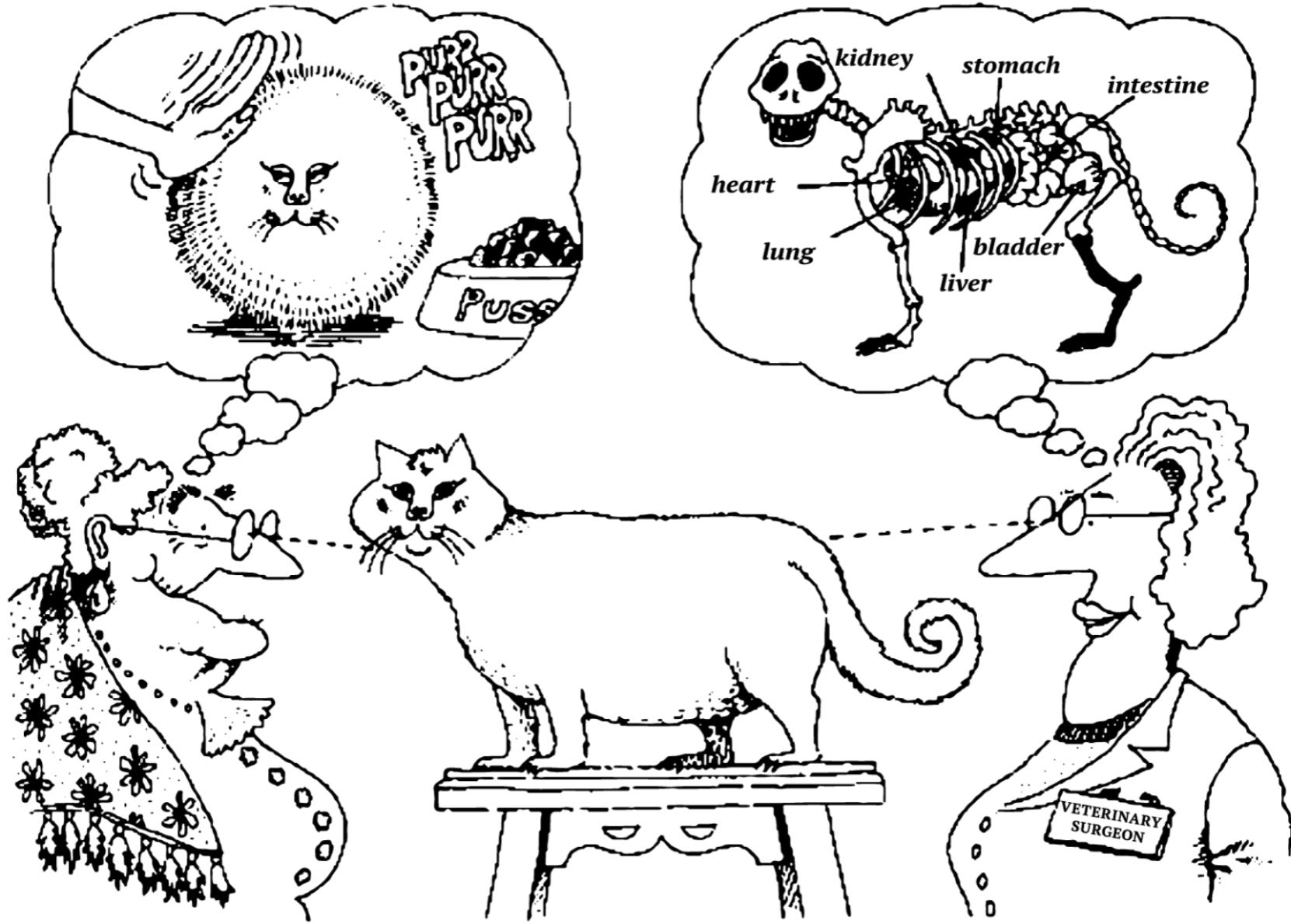
- κατασκευάσουμε μια εφαρμογή λογισμικού που κάνει κάποια χρήσιμη εργασία και την κάνει σωστά και μπορεί να συντηρηθεί

Ανάλογο:

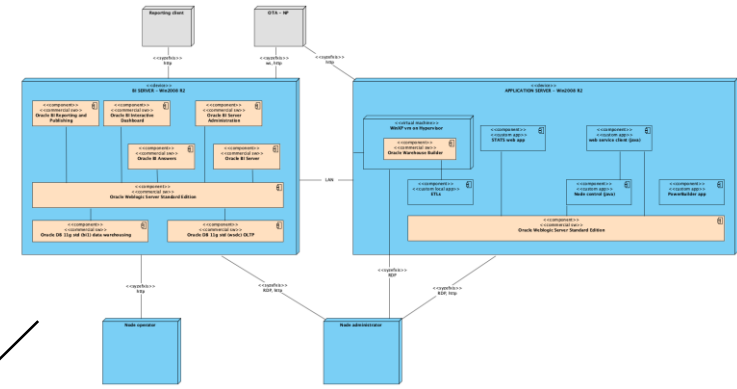
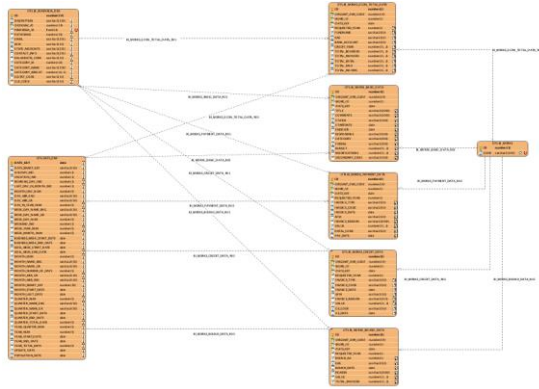
- Η δουλειά του πολιτικού μηχανικού που υπολογίζει τα δομικά χαρακτηριστικά του σπιτιού...
- ...ώστε αυτό να έχει τους χώρους που θέλουμε, να μας αρέσει, αλλά και να είναι στατικά επαρκές, να μπορούμε να το συντηρήσουμε, κ.ά.



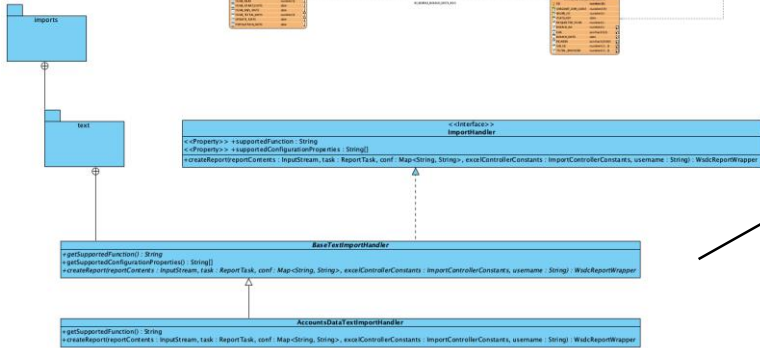
# Αφαίρεση (abstraction): μια κεντρική έννοια



# Αφαίρεση (abstraction)

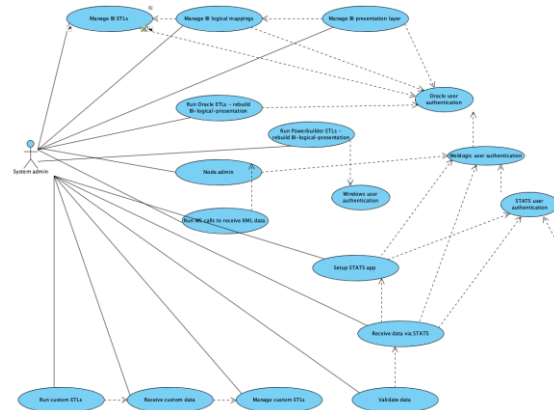


# λογισμικό



**ΠΡΟΤΥΠΟ ΔΟΜΗ 3**  
**Έγγραφο απαιτήσεων λογισμικού (SRS)**  
 (ΣΥΝΔΟΣ ΣΥΣΤΗΜΑΤΟΣ - ΣΥΣΤΗΜΟΥ ΑΔΙΣΤΕΡΟΤΗΤΑ)

1. Εισαγωγή
  - 1.1 Σκοπός του εγγράφου
  - 1.2 Γαβάνια του εγγράφου
  - 1.3 Εξουσιοθέτηση του εγγράφου
  - 1.3.1 Αποδοχή
  - 1.3.2 Αποδοχή
  - 1.3.3 Αποδοχή
  - 1.3.4 Αποδοχή
2. Αναφορές - πηγές πληροφοριών



# Πλάνο μαθήματος

- 03.10.2018 Διεξαγωγή του μαθήματος  
Θεματολογία, πλάνο  
Κύκλος ζωής λογισμικού  
Διαχείριση έργων λογισμικού (I)
- 10.10.2018 Μοντέλα και μεθοδολογίες  
Περιβάλλοντα ανάπτυξης  
Διαχείριση έργων λογισμικού (II)  
Φροντιστήριο: GIT
- 17.10.2018 Ανάλυση απαιτήσεων I (εισαγωγή, είδη απαιτήσεων, συνοπτικά)  
Δομημένη ανάλυση και σχεδιασμός (συνοπτικά)  
1η συζήτηση για την εργασία  
Αντικειμενοστρεφής ανάλυση και σχεδιασμός  
Φροντιστήριο JVM technology

- 24.10.2018 **Ανάλυση απαιτήσεων II (πρότυπα, requirements engineering)**  
Αρχιτεκτονικά πρότυπα λογισμικού
- 31.10.2018 **UML - Visual Paradigm CE**  
Αρχιτεκτονικά πρότυπα στον παγκόσμιο ιστό, Δομικά πρότυπα σχεδίασης
- 07.11.2018 **UML, ISO/IEC/IEEE standards**  
Κατασκευαστικά πρότυπα σχεδίασης, RESTful APIs, AJAX, MVVM, JDBC
- 14.11.2018 Invited lecture I**
- 21.11.2018 **Εργαστήριο για την εργασία**  
Φροντιστήριο: Gradle, Java web app με χρήση gradle
- 28.11.2018 **Front-end UI toolkits**  
Google maps APIs, OpenLayers

**05.12.2018** Invited lecture II

12.12.2018 Διασφάλιση ποιότητας λογισμικού, Ανίχνευσιμότητα απαιτήσεων και σχεδιαστικών αποφάσεων

Πρότυπα σχεδίασης συμπεριφοράς, αναδιάρθρωση κώδικα

19.12.2018 Recap, απορίες, συζήτηση για εργασίες κλπ

Ελεγχος και επαλήθευση κώδικα

Spock testing framework

## Κύκλος Ζωής του Λογισμικού

- Καταγραφή και ανάλυση απαιτήσεων.
- Σχεδιασμός.
- Υλοποίηση.
- Επαλήθευση και επικύρωση.
- Εγκατάσταση, έλεγχος, παραμετροποίηση και ολοκλήρωση λογισμικού στο παραγωγικό του περιβάλλον.
- Συντήρηση και επέκταση.

## Μοντέλα ανάπτυξης λογισμικού

- Ακολουθιακή διαδικασία (sequential)
- Μοντέλο Καταρράκτη (waterfall)
- Επαναληπτική διαδικασία (iterative)
- Αυξητική διαδικασία (incremental)
- Ευέλικτη διαδικασία (agile)



## Μεθοδολογίες ανάπτυξης λογισμικού

- Ταχεία Πρωτοτυποποίηση (Rapid prototyping).
- «Ακραίος Προγραμματισμός» (Extreme Programming).
- Ανάπτυξη βασισμένη σε ελέγχους (Test-driven Development).
- Συνεχής παράδοση (Continuous Delivery)
- Ανάπτυξη / Λειτουργία («DevOps»).

## Ανάλυση απαιτήσεων

- Ανάλυση και μοντελοποίηση απαιτήσεων
- Σύνταξη προδιαγραφών και παραδοτέων

## Σχεδιασμός Λογισμικού

- Βασικές αρχές
- Αντικειμονοστρεφής σχεδιασμός συστημάτων
- Συστατικά λογισμικού

## Γλώσσα μοντελοποίησης UML

- Διαγράμματα κλάσεων (Class diagrams)
- Διαγράμματα ακολουθιών (Sequence diagrams)
- Διαγράμματα δραστηριοτήτων (Activity diagrams)
- Διαγράμματα σεναρίων χρήσης (Use-case diagrams)
- κτλ.

Εργαλείο: Visual Paradigm (CE)

Πρότυπο: ISO/IEC/IEEE 29148/2011

## Πρότυπα Σχεδίασης (Design Patterns)

- Κατασκευαστικά πρότυπα (Creational patterns)
- Δομικά πρότυπα (Structural patterns)
- Πρότυπα συμπεριφοράς (Behavioral patterns)

# Αρχιτεκτονική Λογισμικού

- Αρχιτεκτονικός σχεδιασμός σύνθετων κατανεμημένων συστημάτων
  - Βασικές έννοιες
  - Έμφαση σε εφαρμογές διαδικτύου
  - Αρχιτεκτονική REST (Representation State Transfer)
- Ανάπτυξη RESTful Application Programming Interfaces (APIs)
- Θέματα ασφάλειας
- Θέματα απόδοσης

# Σχεδιασμός διεπαφής χρήστη

- Ευχρηστία, διαδραστικότητα και αποκρισιμότητα
- Σύγχρονες μεθοδολογίες ανάπτυξης διεπαφής χρήστη
  - Πρότυπα σχεδίασης Model-View-Controller και Observable
  - Έμφαση σε τεχνολογίες εφαρμογών διαδικτύου (HTML5, CSS3, Javascript)
  - Εφαρμογές μιας σελίδας (Single-page applications)
  - Ασύγχρονες τεχνικές (AJAX, Promises, Reactive frameworks)



# Διοίκηση και Διαχείριση Έργου Λογισμικού

- Διοίκηση έργου (γενικά)
  - Εισαγωγή και βασικές έννοιες
  - Εκτίμηση κόστους έργου
- Διοίκηση ομάδας ανάπτυξης λογισμικού
  - Συνήθεις ρόλοι σε μια ομάδα ανάπτυξης λογισμικού
  - Αλληλεπιδράσεις μεταξύ ρόλων
- Καλές διεθνείς πρακτικές

# Τεχνικές και Εργαλεία Διοίκησης, Παρακολούθησης και Ελέγχου της Ανάπτυξης Λογισμικού

- Έλεγχος εκδόσεων (Version Control)
  - Έμφαση στο σύστημα Git (το πλέον διαδεδομένο Version Control System)
- Αυτοματισμός διαδικασίας «χτισίματος» λογισμικού (Build automation)
  - Έμφαση στο σύστημα Gradle (χρησιμοποιείται στην ανάπτυξη Android εφαρμογών)
- Ανάλυση προγράμματος (program analysis) και αυτόματος εντοπισμός σφαλμάτων (bug detection)

- Σενάρια ελέγχου
  - Unit testing, Regression testing, Functional testing, Integration testing, test coverage
- Συνεχής ολοκλήρωση (Continuous integration)
- Αξιοπιστία λογισμικού
- Η έννοια του «τεχνικού/σχεδιαστικού χρέους» (technical debt).

## Διαχείριση συστατικών του λογισμικού

- Συστατικά λογισμικού και αλληλεξαρτήσεις τους (software components and dependencies)
- Αποθήκες συστατικών λογισμικού (Software artifact repositories)
- Διαχείριση εκδόσεων λογισμικού (software releases)
- Μέθοδοι αρίθμησης εκδόσεων (versioning schemes)