



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τεχνολογία Λογισμικού, 7ο εξάμηνο 2019-2020

# 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!

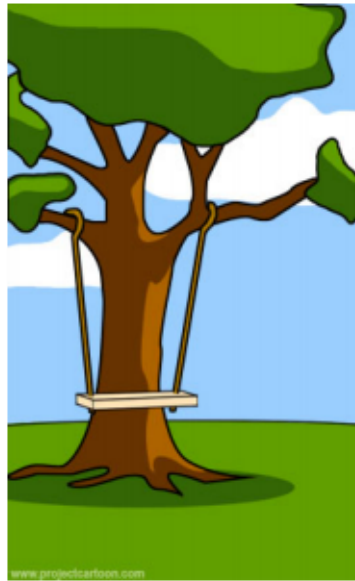
## Β.Βεσκούκης

- Δρ. ΗΜΜΥ (1995 )
- Αν.Καθ., Σχολή ΑΤΜ "Τεχνολογία Λογισμικού στη Γεωπληροφορική"
- ETH 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



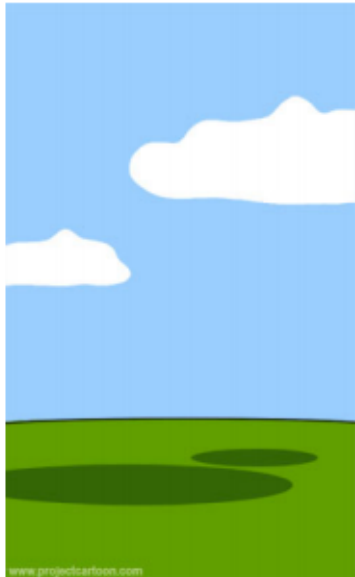
How the analyst designed it



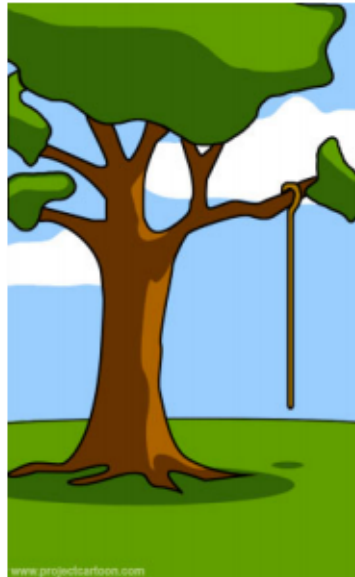
How the programmer wrote it



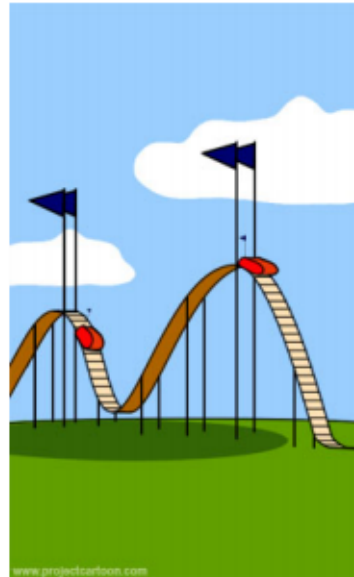
How the business consultant described it



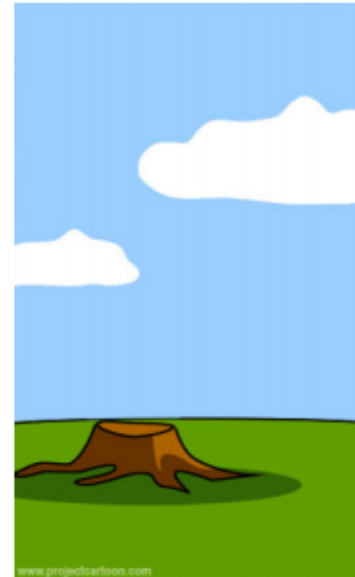
How the project was documented



What operations installed



How the customer was billed



How it was supported



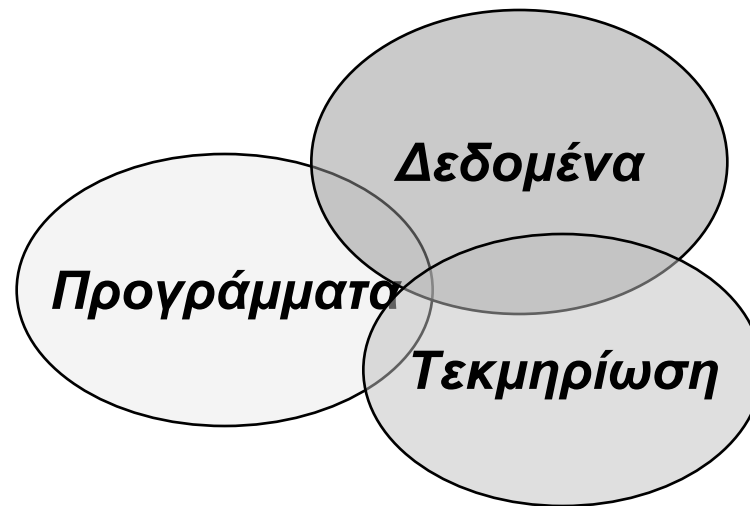
What the customer really needed

# Τι είναι Λογισμικό;

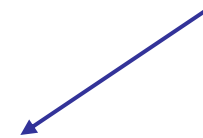
Προγράμματα ηλεκτρονικού υπολογιστή

Δομές δεδομένων με μόνιμη ή προσωρινή αποθήκευση

Τεκμηρίωση σχετικά με τα παραπάνω



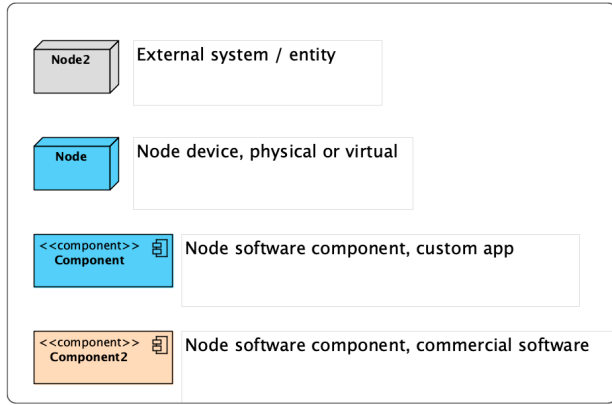
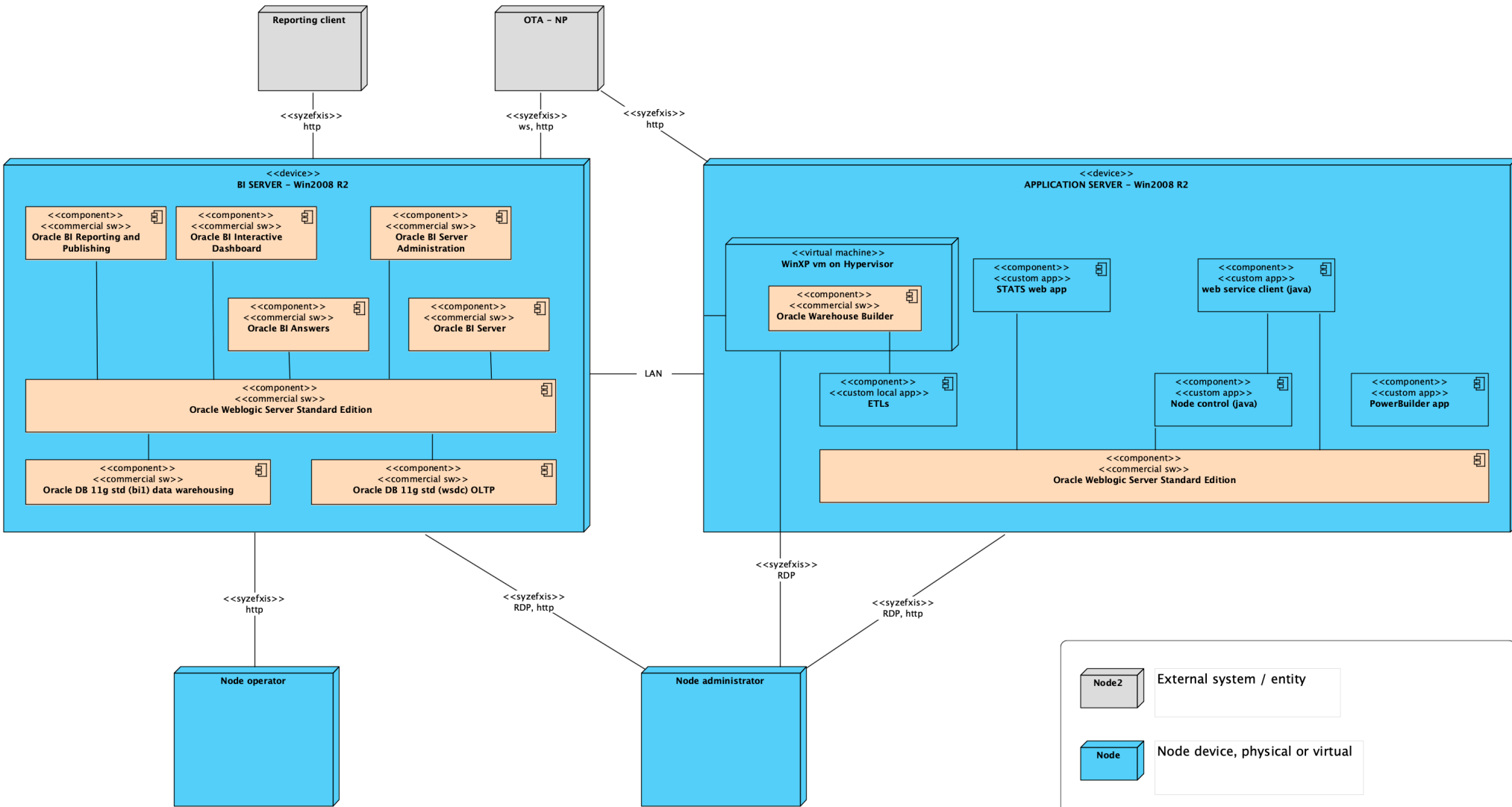
*Συστατικά λογισμικού*



# 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:

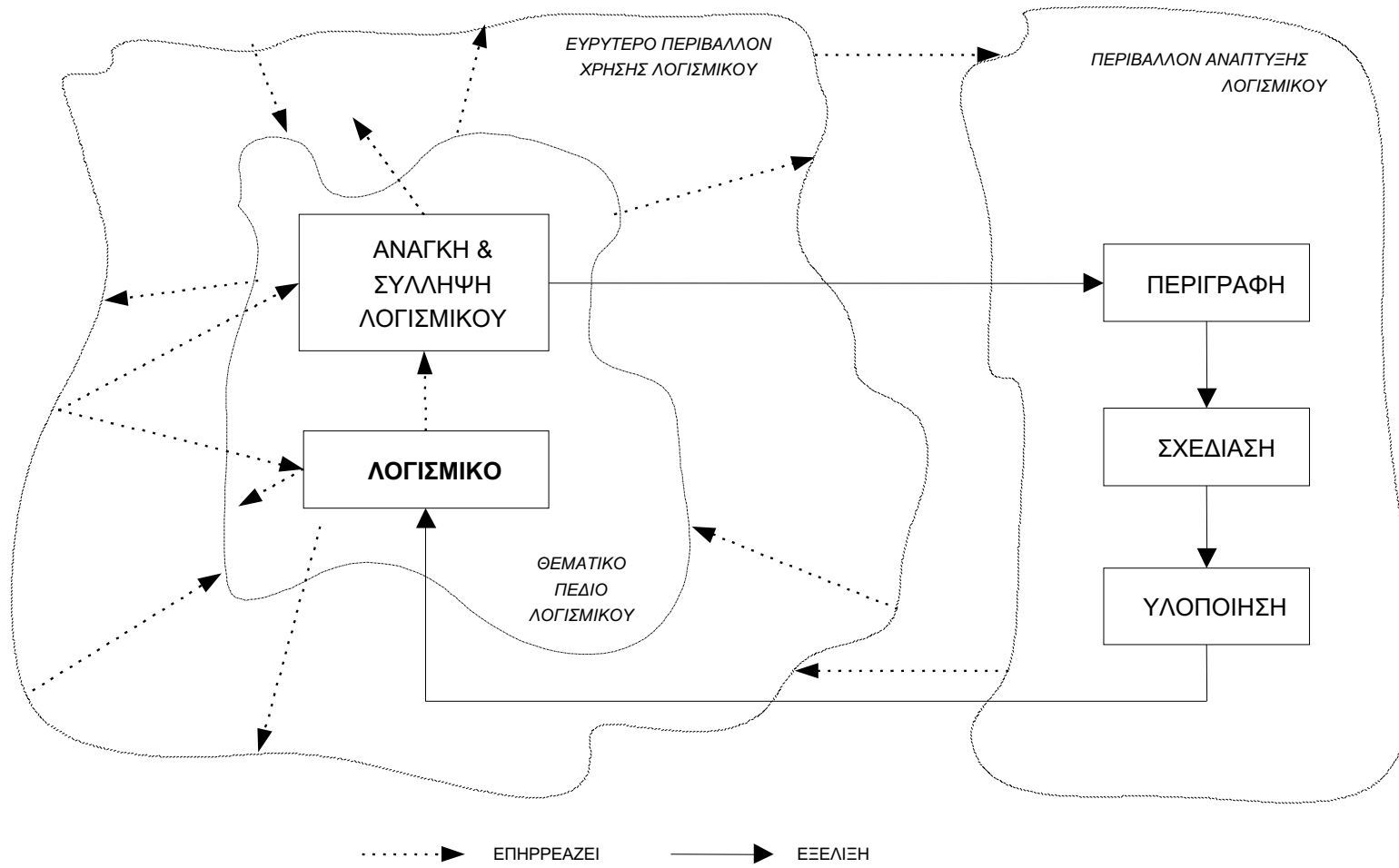
## Τεχνολογία: «Λόγος περί Τέχνης»

- 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 σχεδίασή τους πριν την κατασκευή τους,

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

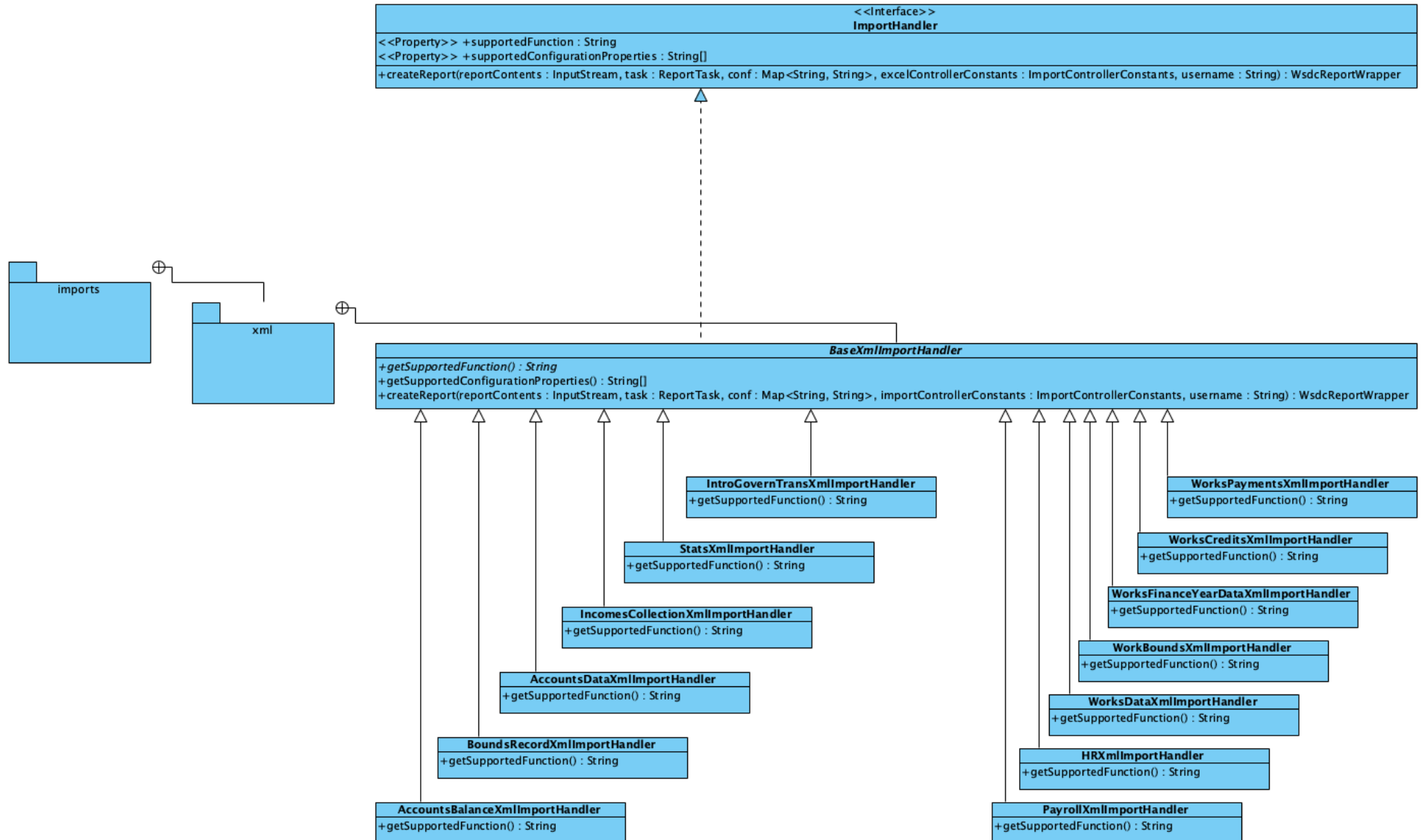
Σχεδίαση: προσδιορισμός των συστατικών στοιχείων μιας κατασκευής, των συσχετίσεων μεταξύ αυτών, και του τρόπου χρήσης κάθε συστατικού

...

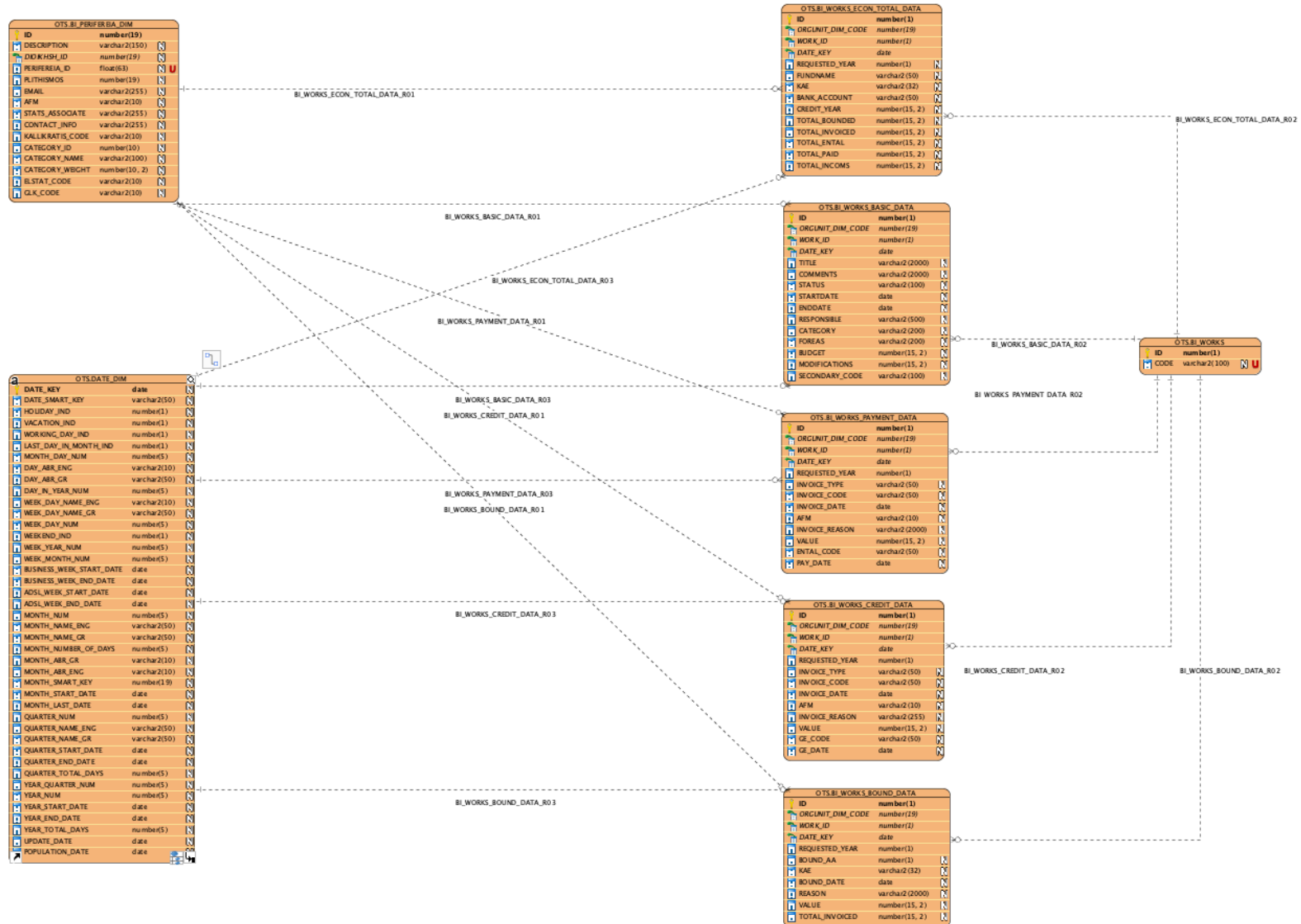
Ετσι και στο λογισμικό:  
επιδιώκουμε τον προσδιορισμό των στοιχείων που (πρέπει να) το αποτελούν, ώστε να το κατασκευάσουμε «σωστά»

- 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

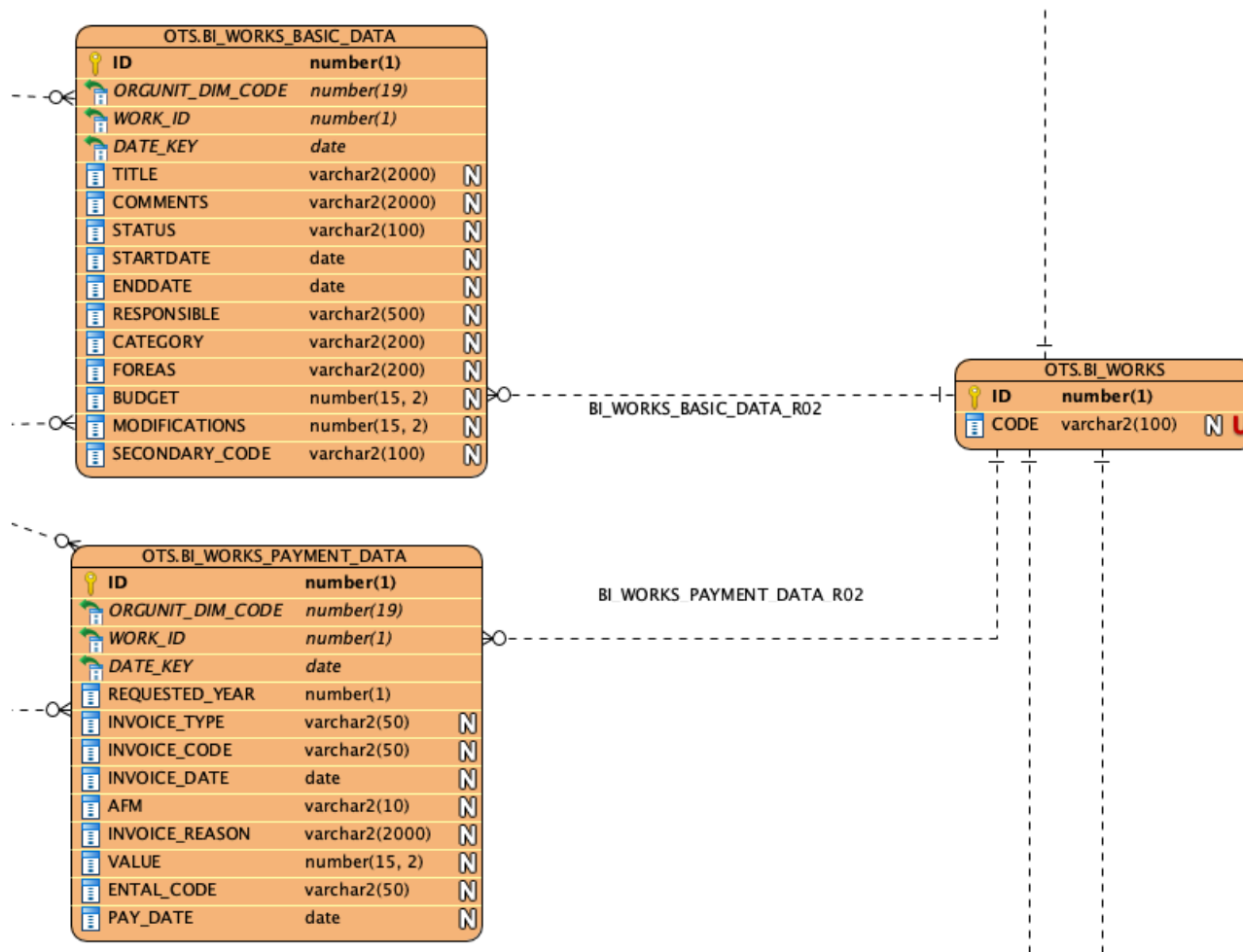
# "...προσδιορισμός των στοιχείων του λογισμικού"



# "...προσδιορισμός των στοιχείων του λογισμικού"



"...προσδιορισμός των στοιχείων του λογισμικού"



# Κρίση λογισμικού (όρος του 1968!)

• <b>Εξαιρετικά δύσκολη διαδικασία κατασκευής</b>	<i>Δεν είναι πάντα σαφές ποια βήματα πρέπει να γίνουν, με ποια σειρά, με ποια ενδιάμεσα προϊόντα κλπ</i>
• <b>Ανεπαρκής ή και κακή ποιότητα τελικού προϊόντος</b>	<i>Λάθη στην κατασκευή, μη ικανοποίηση του σκοπού</i>
• <b>Μη τήρηση χρονοδιαγραμμάτων</b>	<i>Υπερβολικές και "αδικοιολόγητες" καθυστερήσεις</i>
• <b>Υπερβάσεις προϋπολογισμών</b>	<i>Κακές αρχικές εκτιμήσεις κόστους. Τελικά προϊόντα με πολλαπλάσιο κόστος από το αρχικά προϋπολογισθέν</i>
• <b>Μεγάλη δυσκολία και συνεπαγόμενο κόστος συντήρησης</b>	<i>Παρενέργειες μεταβολών σε στοιχεία που πριν λειτουργούσαν, πρόχειρες λύσεις</i>
• <b>Δύσκολη κατανόηση εγγράφων, σχεδίων κλπ από διαφορετικούς κατασκευαστές</b>	<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...

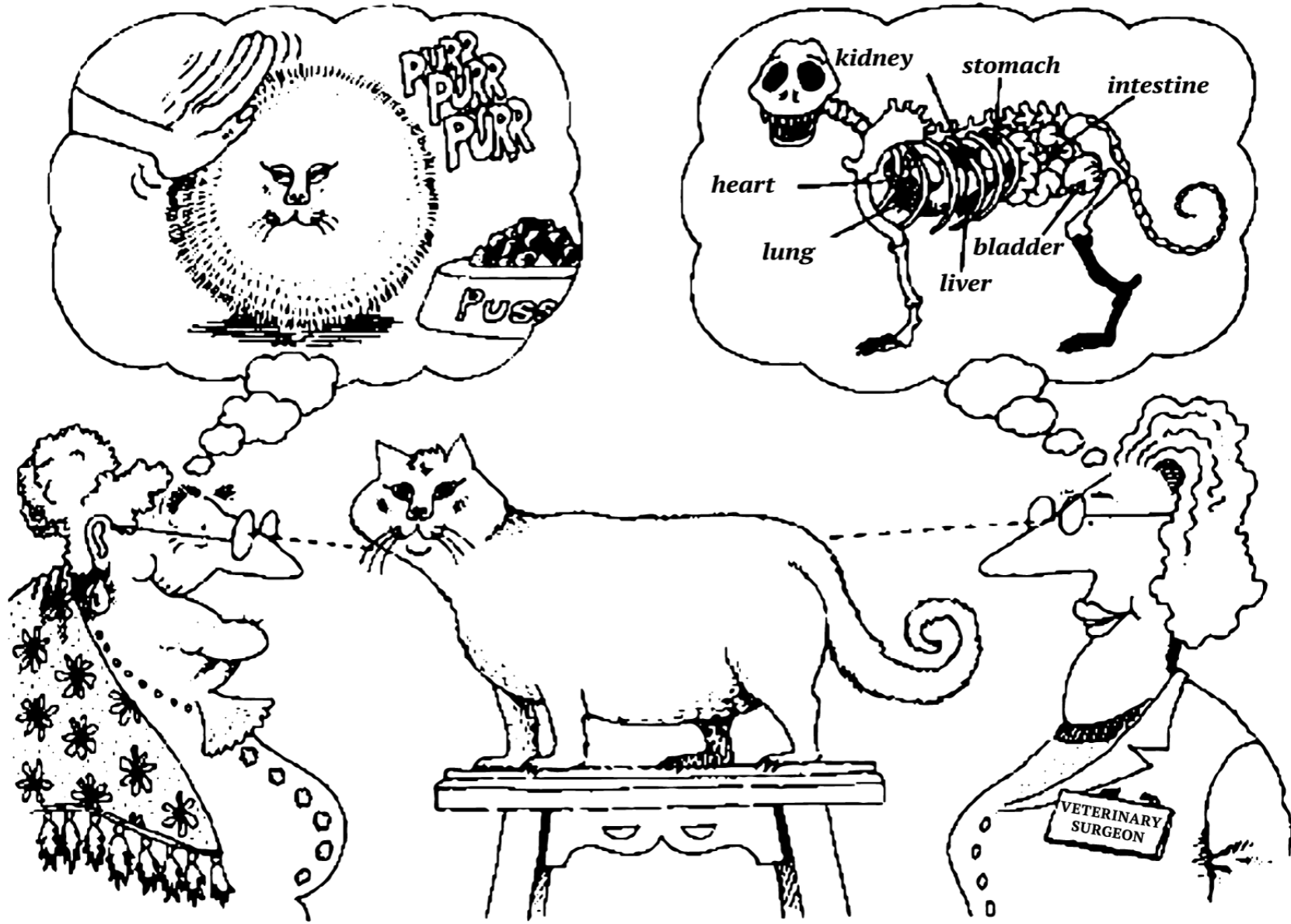
- κατασκευάσουμε μια εφαρμογή λογισμικού που κάνει κάποια χρήσιμη εργασία και την κάνει σωστά και μπορεί να συντηρηθεί

Ανάλογο:

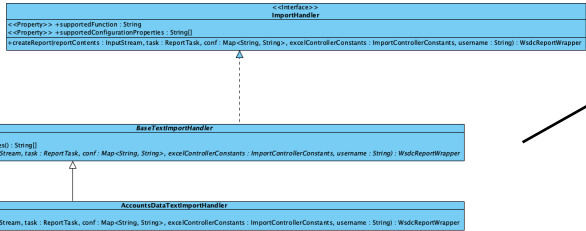
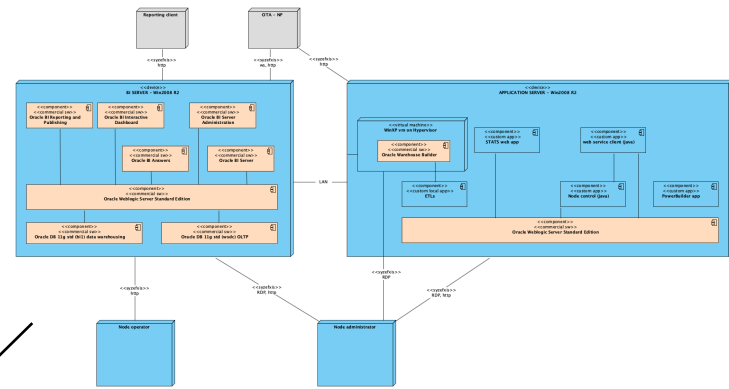
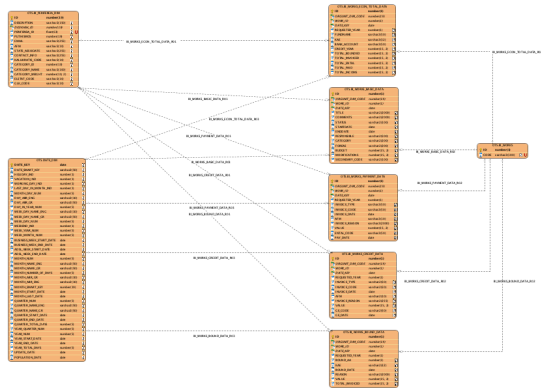
- Η δουλειά του πολιτικού μηχανικού που υπολογίζει τα δομικά χαρακτηριστικά του σπιτιού...
- ...ώστε αυτό να έχει τους χώρους που θέλουμε, να μας αρέσει, αλλά και να είναι στατικά επαρκές, να μπορούμε να το συντηρήσουμε, κ.ά.



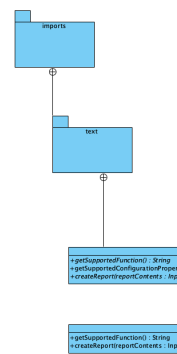
# Αφαίρεση (abstraction): μια κεντρική έννοια



# Αφαίρεση (abstraction)



## Λογισμικό



**ΠΡΟΤΥΠΟ ΔΟΜΗ 3**  
Έγγραφο απαιτήσεων Λογισμικού (SRS)

(ΤΙΤΛΟΣ ΣΥΣΤΗΜΑΤΟΣ - ΣΤΟΙΧΕΙΟ ΛΟΓΙΣΜΙΚΟΥ)

1. Εισαγωγή

1.1 Σκοπός του λογισμικού

1.2 Δομή του λογισμικού

1.3 Ορισμοί και συντομογραφίες

1.3.1 **Απαιτήσεις:** Η λειτουργία του λογισμικού...

1.3.2 **Διαδικασίες:** Η διαδικασία που ακολουθείται...

1.3.3 **Απαιτήσεις:** Η λειτουργία του λογισμικού...

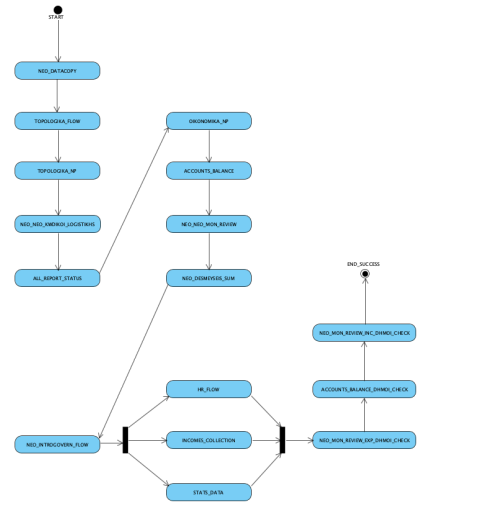
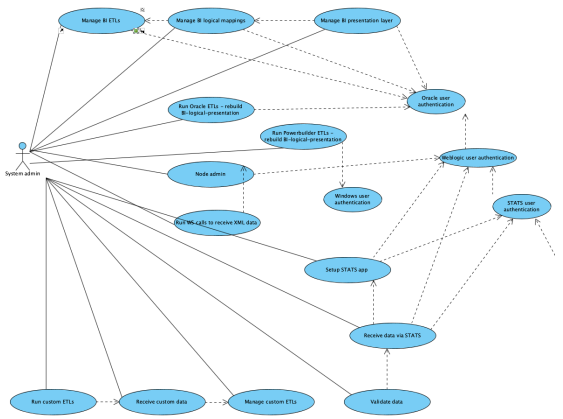
1.3.4 **Απαιτήσεις:** Η λειτουργία του λογισμικού...

1.4 Οφέλη

2. Αναφορές - πηγές πληροφοριών

ΕΠΙΣΤΑΣΗ ΟΜΟΛΟΓΗΣΗΣ: ΕΠΙΒΛΕΠΩΣΤΗΣ

ΕΠΙΣΤΑΣΗ ΟΜΟΛΟΓΗΣΗΣ: ΕΠΙΒΛΕΠΩΣΤΗΣ



Κύκλος ζωής λογισμικού

# Κύκλος ζωής Λογισμικού

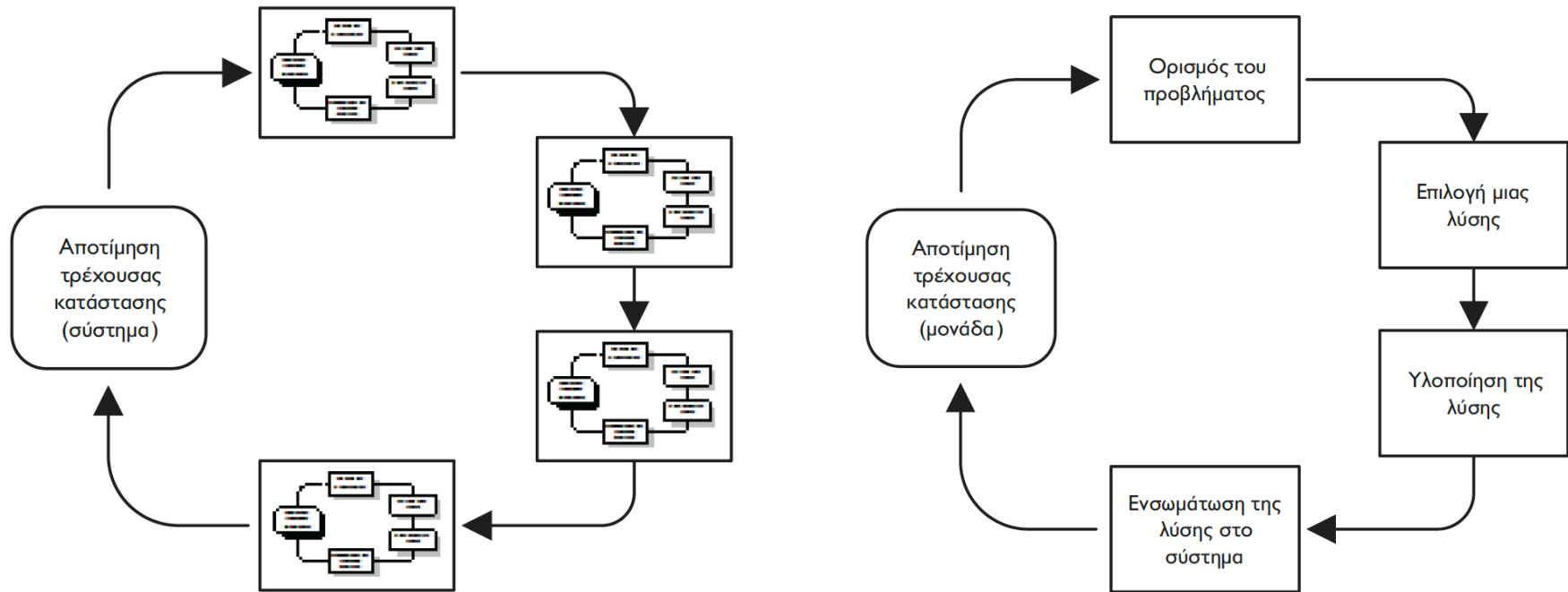
Οι φάσεις από τις οποίες διέρχεται μια εφαρμογή Λογισμικού

- από την σύλληψή της
- κατά την κατασκευή, χρήση, συντήρησή της
- μέχρι την απόσυρσή της

Ένα **μοντέλο κύκλου ζωής** περιγράφει

- Ποιες είναι αυτές οι φάσεις
- Ποιες εργασίες γίνονται και ποια προϊόντα παράγονται σε καθεμία
- Πώς διαδέχονται η μία την άλλη
- Σε ποια έκταση (του λογισμικού) εκτελούνται

# Problem solving, iterative, creative!...



# Παράσταση μοντέλων κύκλου ζωής

Με τη βοήθεια ενός διαγράμματος δραστηριοτήτων

- Μέσα σε ένα παραλληλόγραμμο περιέχεται ο τίτλος μιας εργασίας
- Με βελάκια υποδηλώνεται η μετάβαση σε κάποια άλλη εργασία

Με λεπτομερή δομημένη τεκμηρίωση σε μορφή κειμένου

Με **πρότυπα** για την περιγραφή των συστατικών λογισμικού



# Χαρακτηριστικά μοντέλα κύκλου ζωής λογισμικού

Το μοντέλο του καταρράκτη

Το μοντέλο πρωτοτυποποίησης

Το μοντέλο λειτουργικής επαύξησης

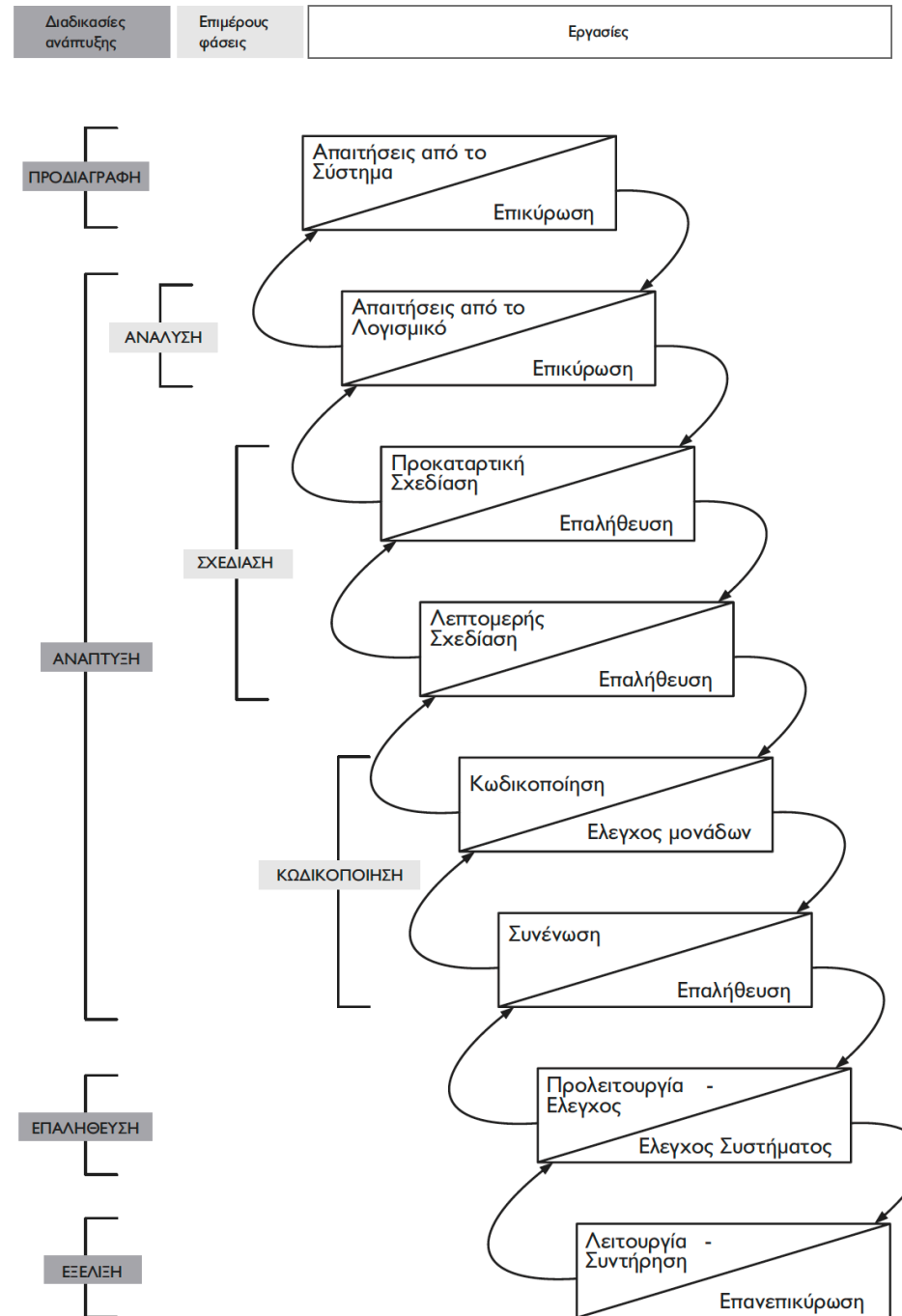
Το σπειροειδές μοντέλο

Το μοντέλο του πίδακα

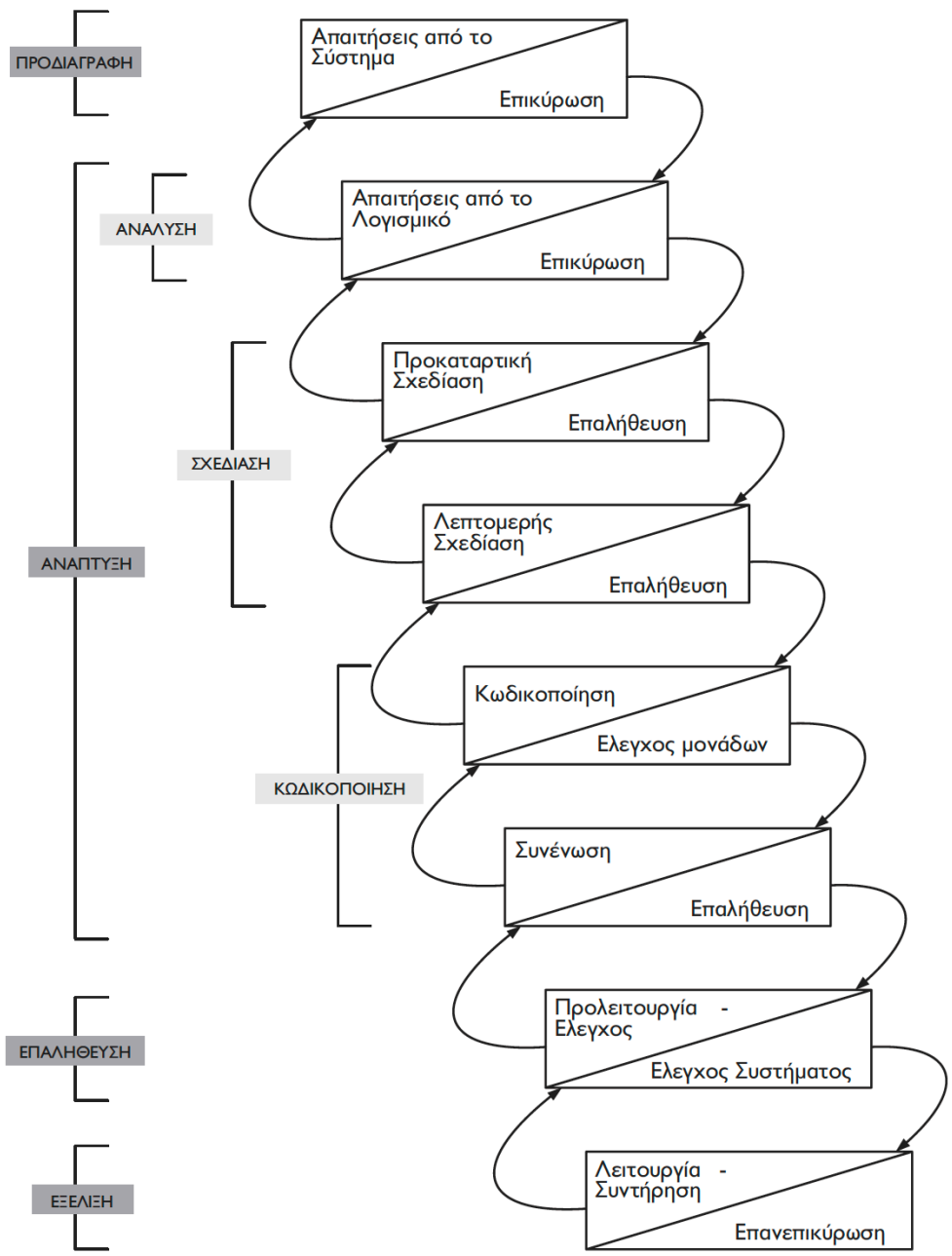
Ένα γενικό μοντέλο κύκλου ζωής

# Το μοντέλο του καταρράκτη

Ολόκληρη η εφαρμογή λογισμικού διέρχεται από διακριτές φάσεις ανάπτυξης.

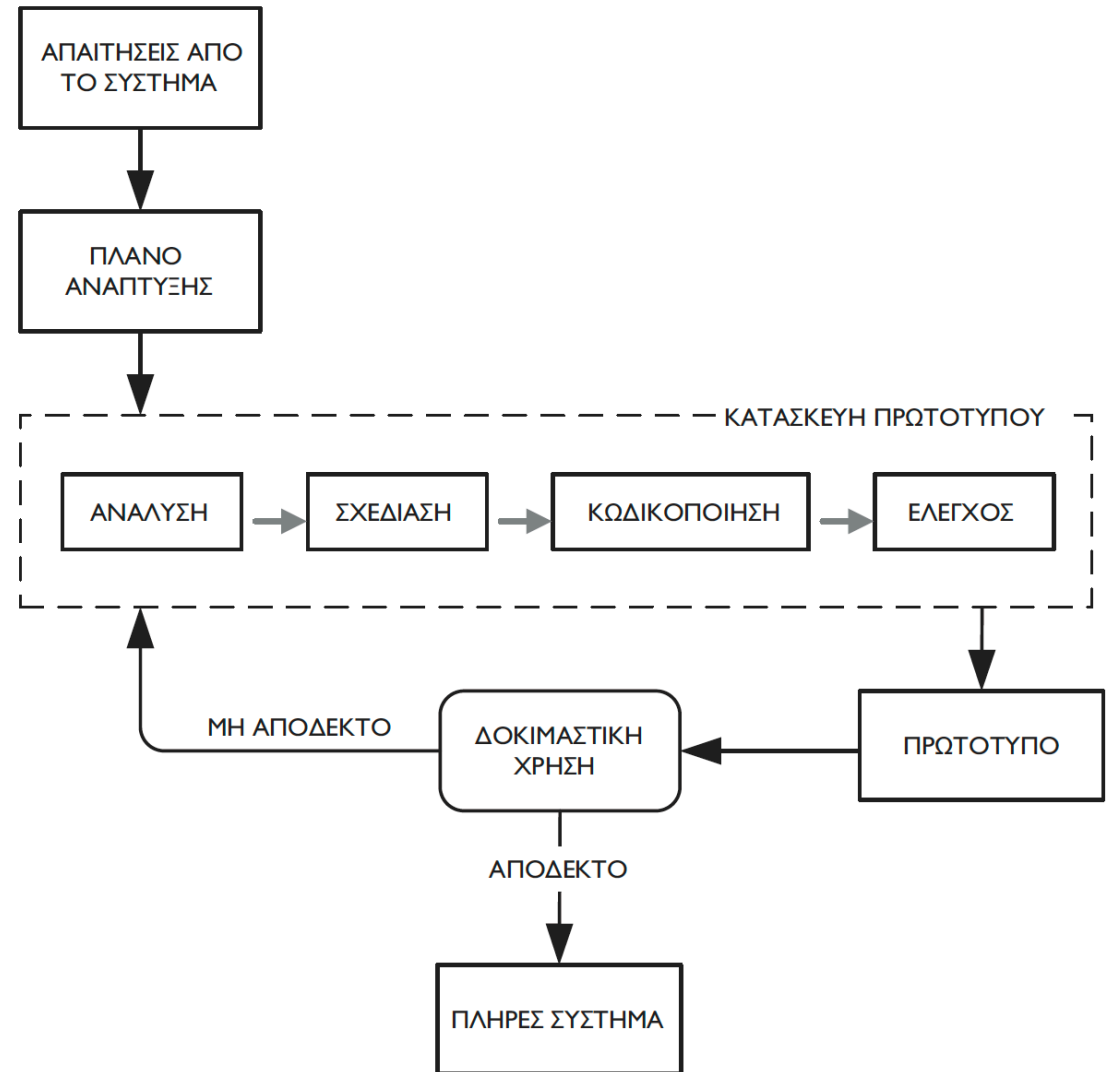


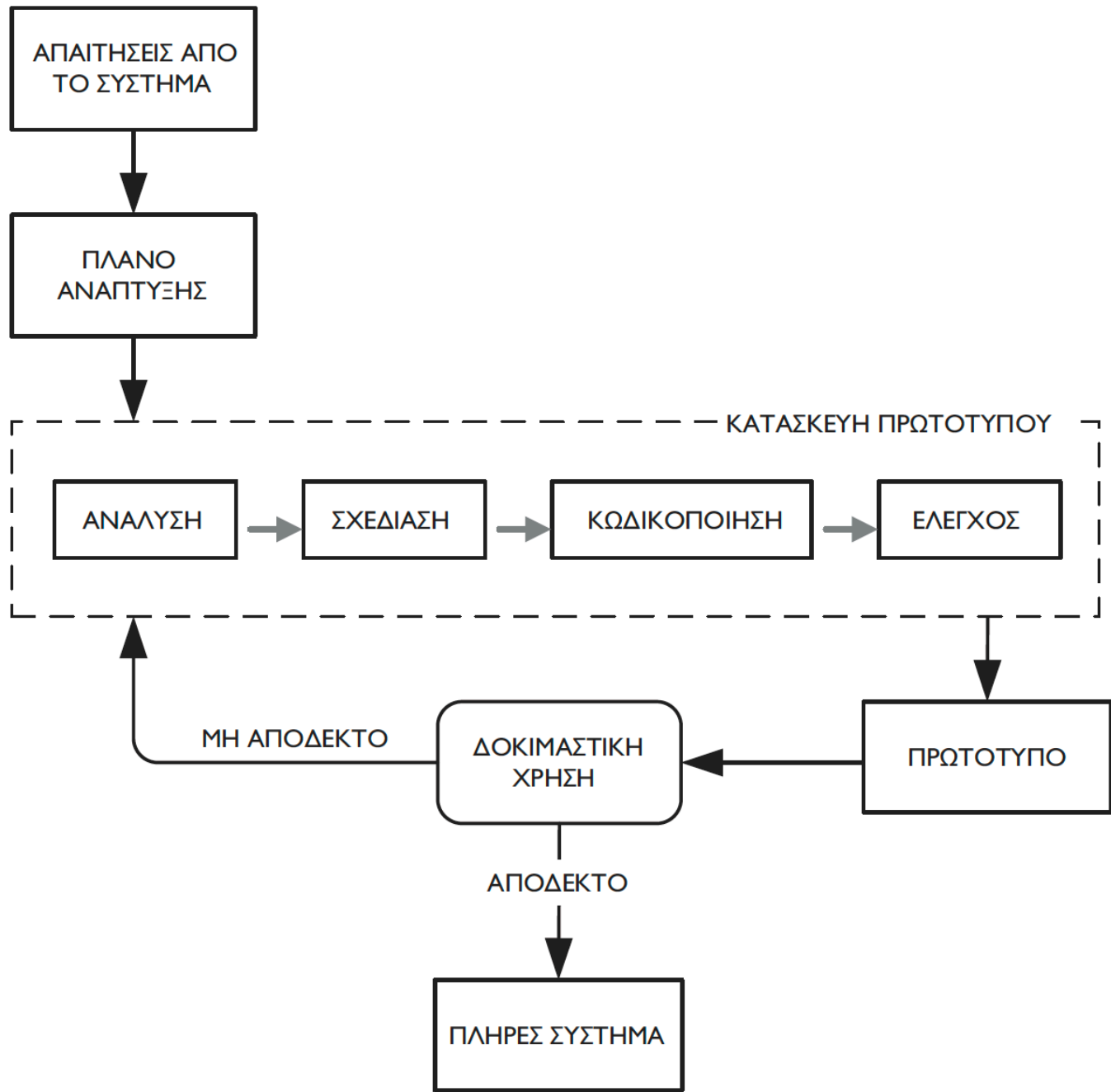




# Το μοντέλο πρωτοτυποποίησης

Κατασκευάζονται διαδοχικά πρωτότυπα με ολοένα και περισσότερα χαρακτηριστικά.

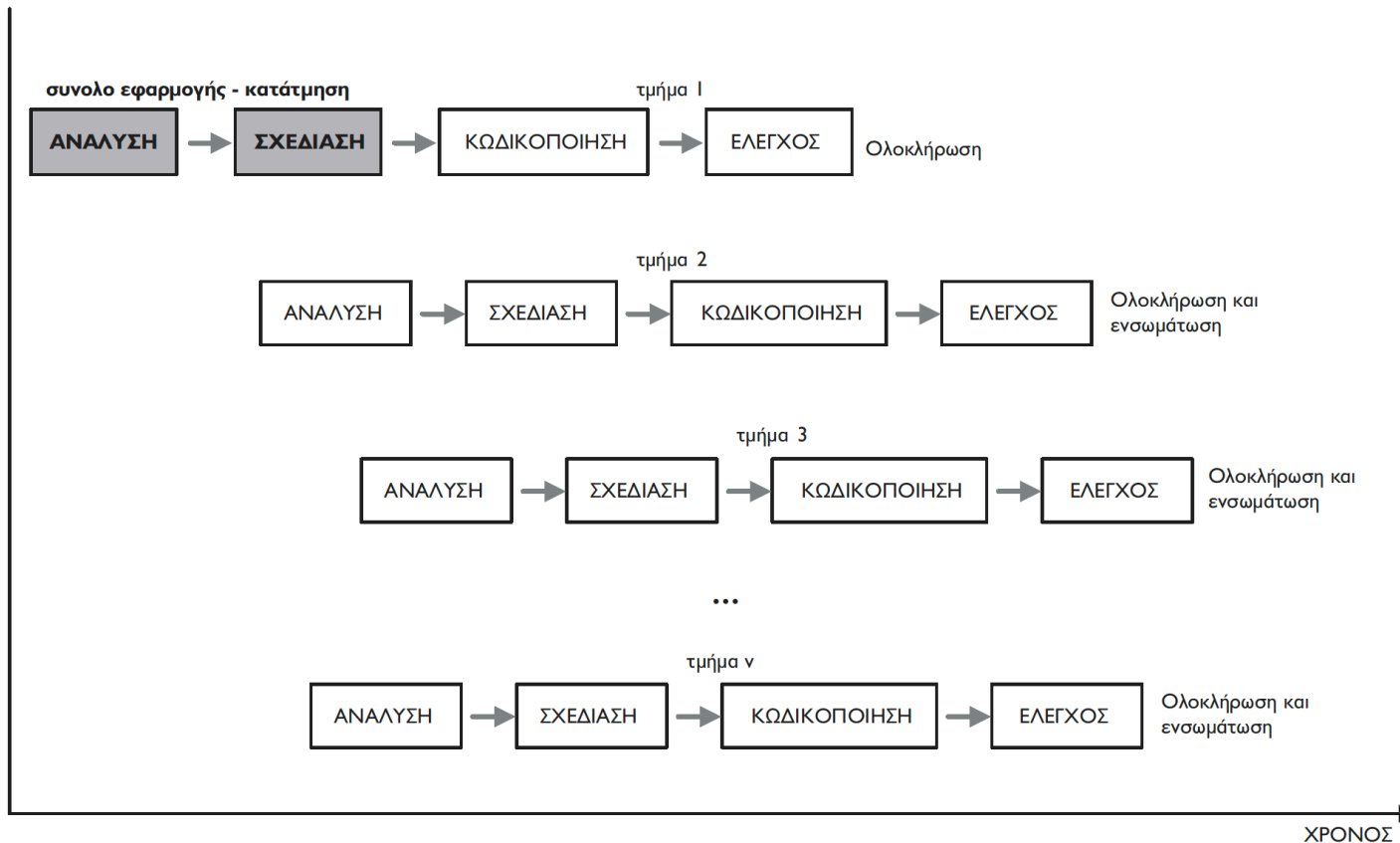


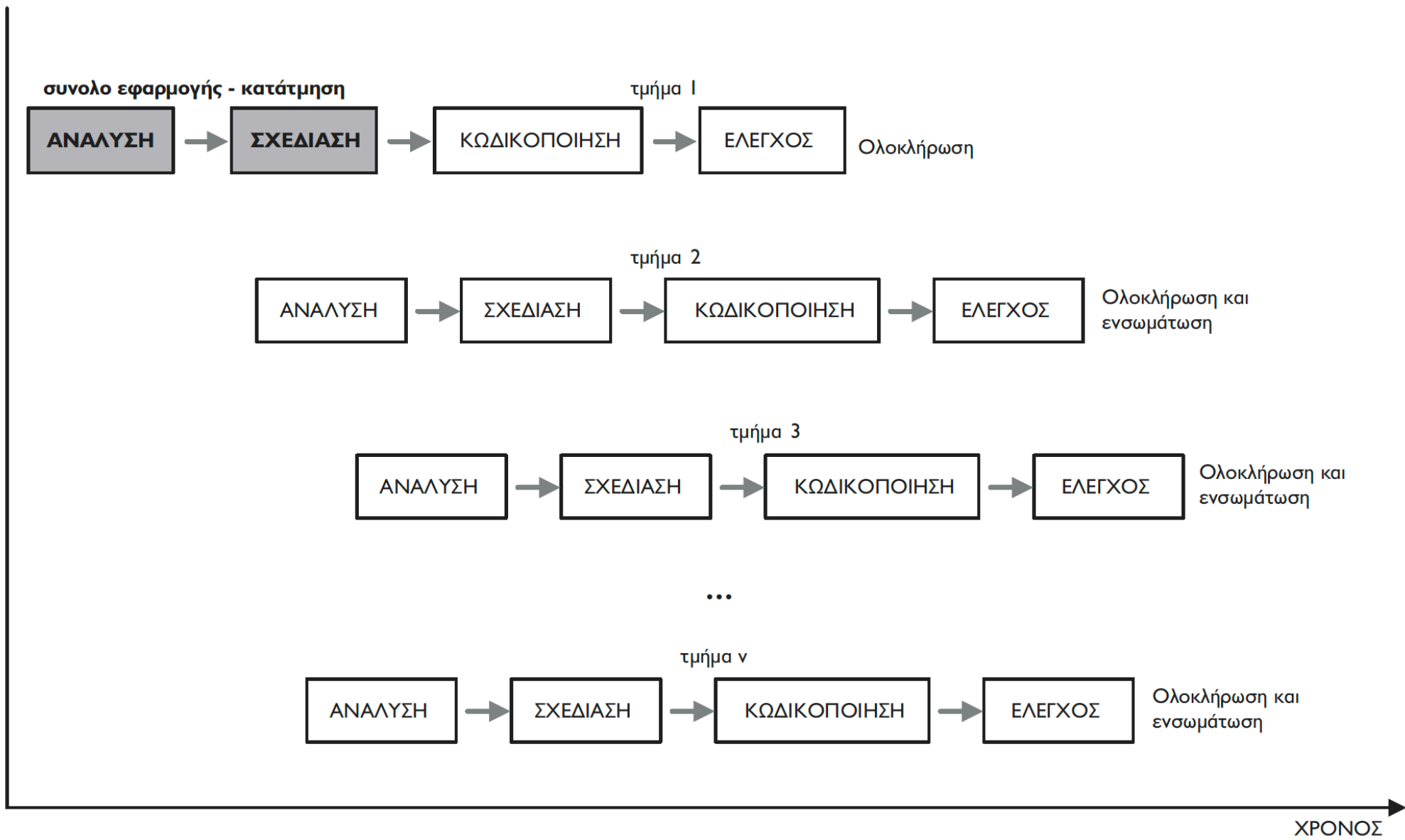


# Το μοντέλο λειτουργικής επαύξησης

Κατάτμηση του λογισμικού και εφαρμογή του μοντέλου του καταρράκτη σε κάθε τμήμα.

Συνένωση στο τέλος

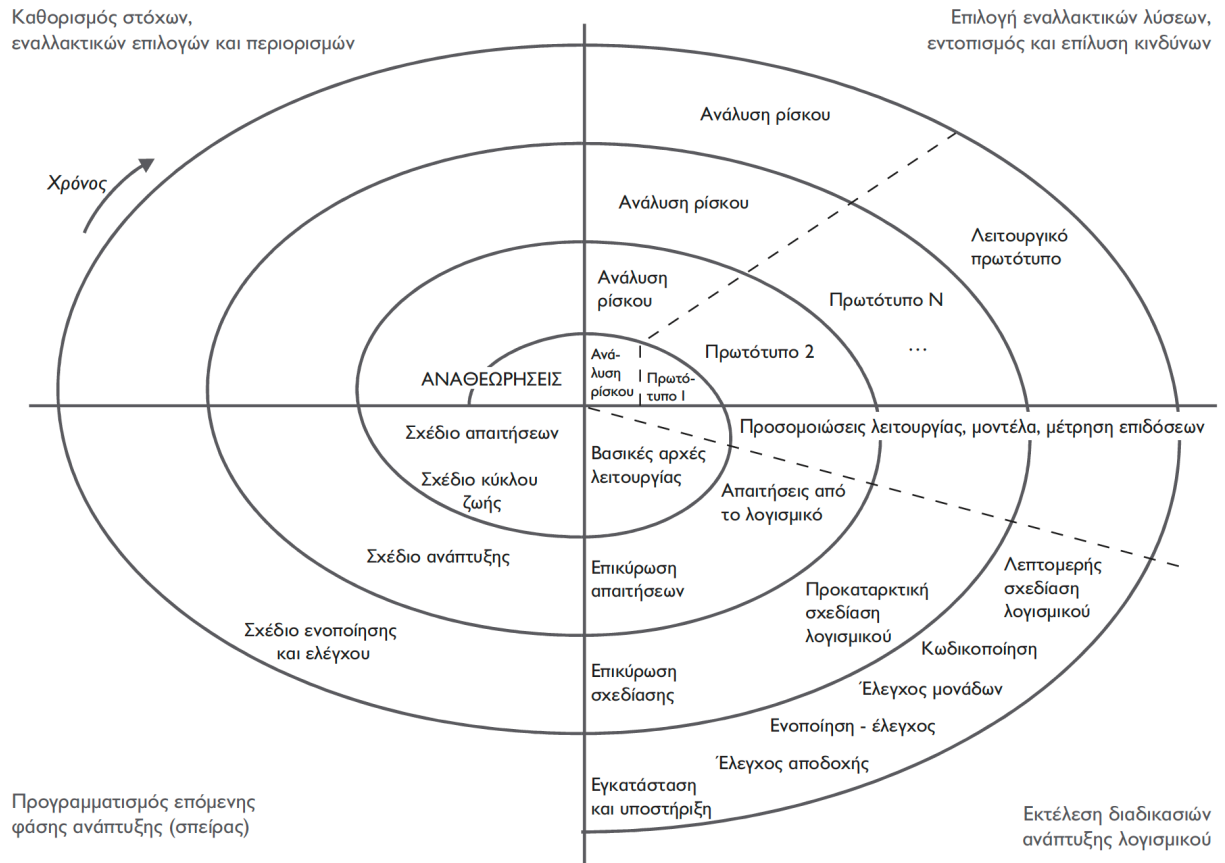




# Το σπειροειδές μοντέλο

Κύκλοι εργασιών με σταδιακή επέκταση των λειτουργικών χαρακτηριστικών της εφαρμογής.

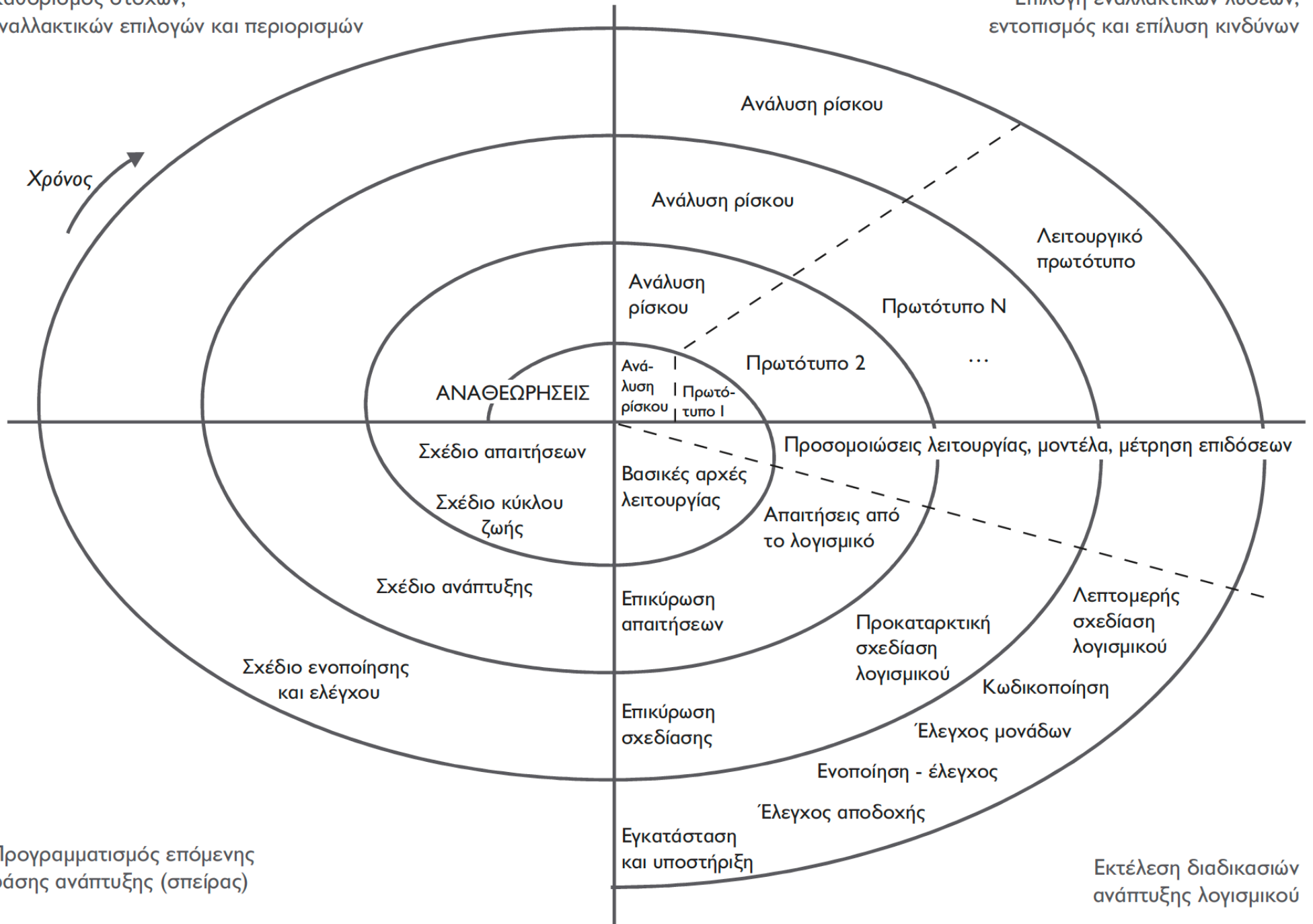
Εκτίμηση του ρίσκου σε κάθε κύκλο



Καθορισμός στόχων,  
εναλλακτικών επιλογών και περιορισμών

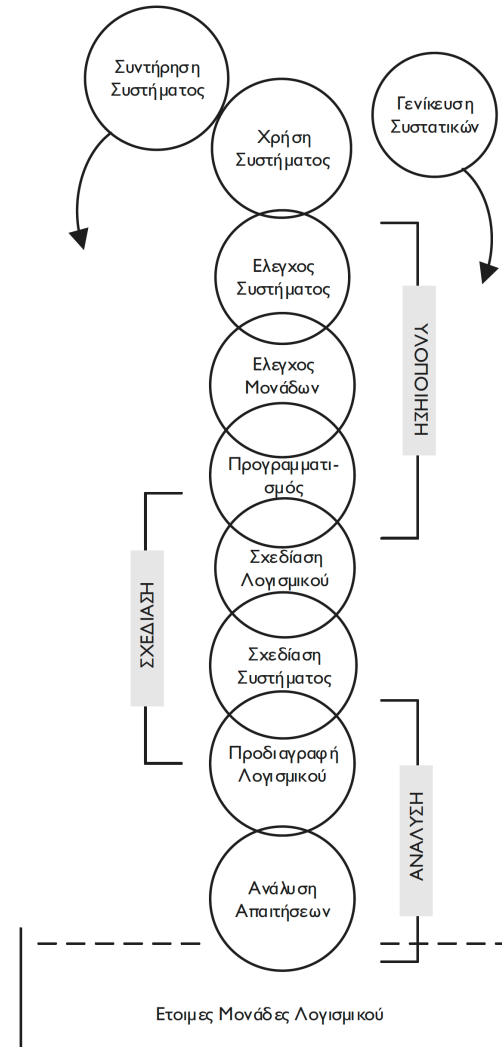
Επιλογή εναλλακτικών λύσεων,  
εντοπισμός και επίλυση κινδύνων

Χρόνος

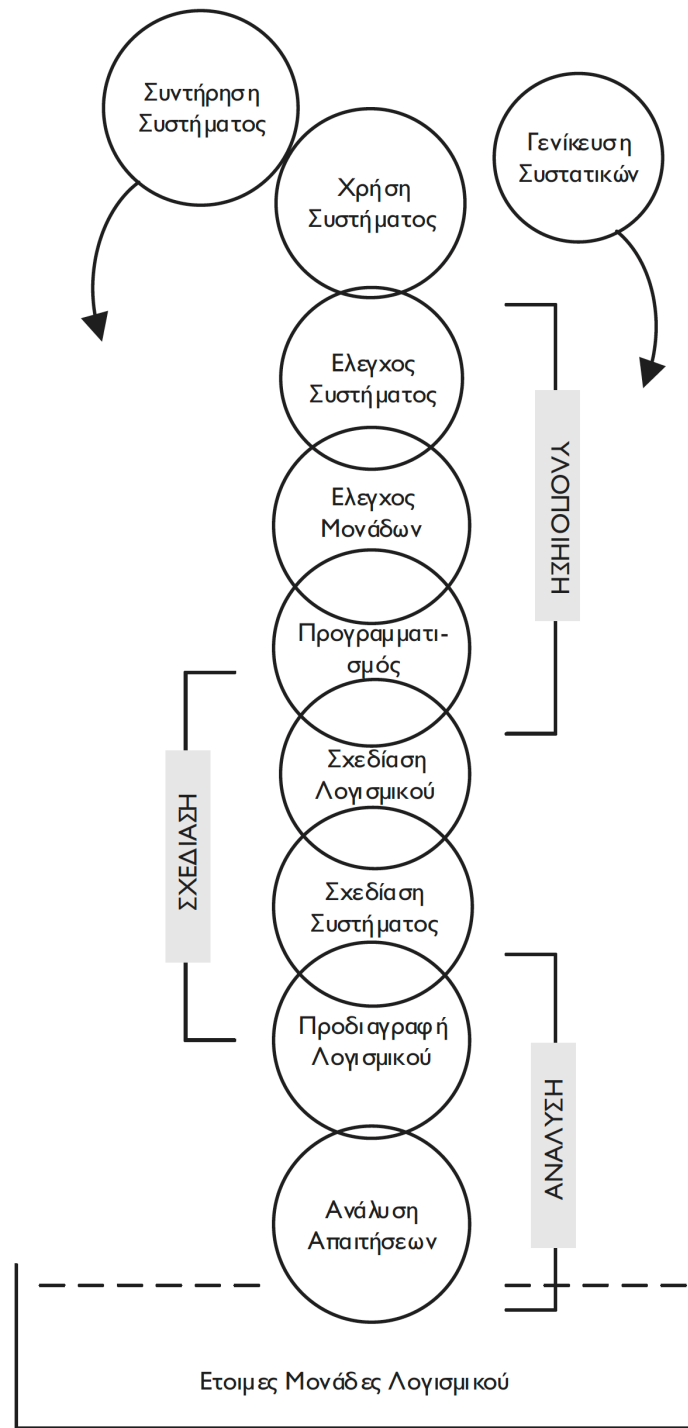


# Το μοντέλο του πίδακα

Ανάπτυξη με αντικειμενοστρεφή φιλοσοφία και επαναχρησιμοποίηση έτοιμων συστατικών



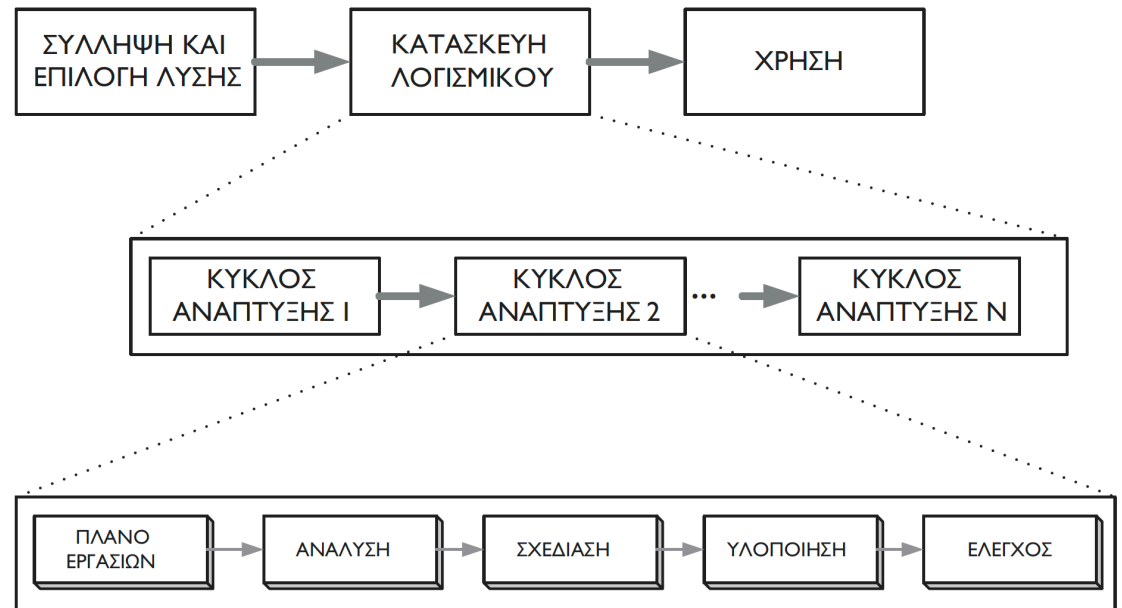


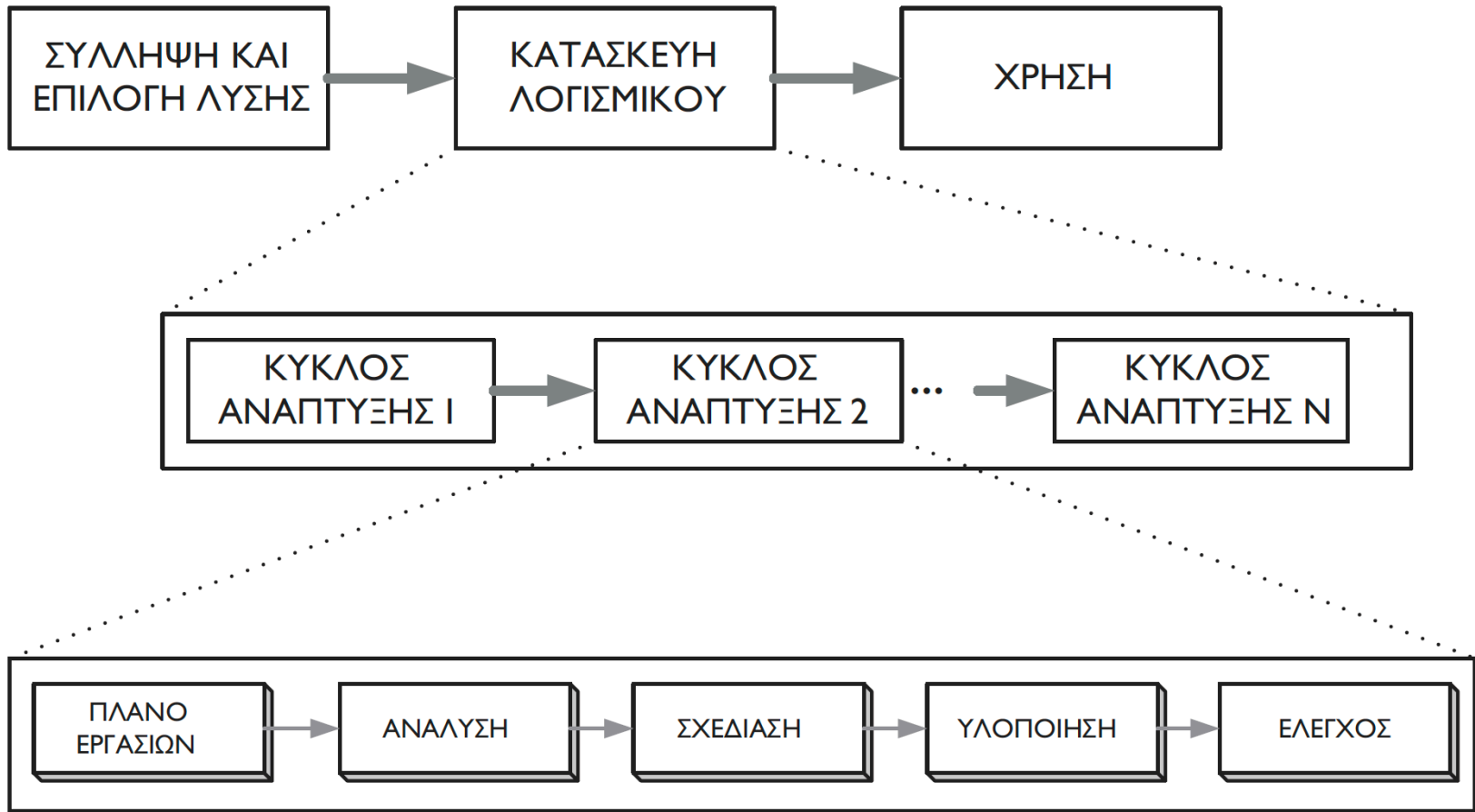


# Ένα γενικό μοντέλο κύκλου ζωής

Ανάπτυξη σε κύκλους σύμφωνα με τα χαρακτηριστικά και τις δυνατότητες του κατασκευαστή.

Γενικευμένη μορφή των προηγούμενων μοντέλων κύκλου ζωής.





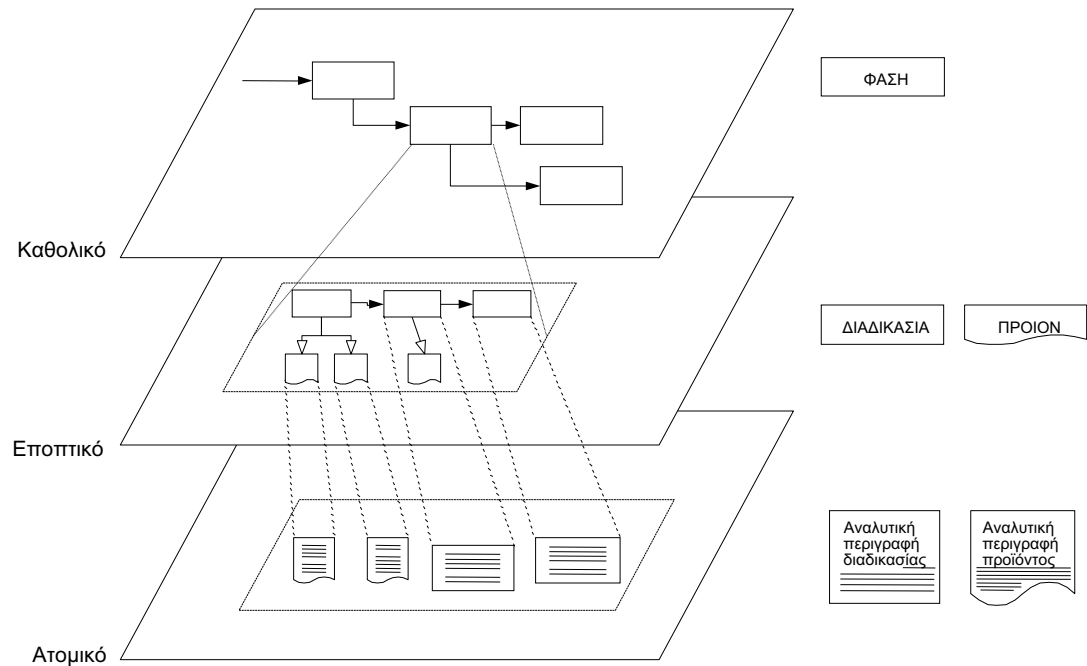
# Σύγκριση μοντέλων κύκλου ζωής λογισμικού

(κάπως υποκειμενική)

<b>Μοντέλο</b>	<b>Μέγεθος εφαρμογών</b>	<b>Μεταβολές στις απαιτήσεις</b>	<b>Προσαρμοστικότητα στον κατασκευαστή</b>	<b>Διάδοση</b>
Καταρράκτη	Μικρό έως μεσαίο	Ανεπιθύμητες	Καμία	Μεγάλη με τάση μείωσης
Πρωτοτυποποίησης	Μικρό ως μεσαίο	Δεκτές	Μικρή	Μικρή με τάση αύξησης
Λειτουργικής επαύξησης	Μεσαίο ως μεγάλο	Ανεπιθύμητες	Καμία	Μικρή με τάση μείωσης
Σπειροειδές	Μεσαίο ως μεγάλο	Δεκτές	Αρκετή	Μικρή με τάση μείωσης
Πίδακα	Οποιοδήποτε	Δεκτές	Αρκετή	Μικρή
Γενικό	Οποιοδήποτε	Δεκτές	Μεγάλη	Μικρή με ισχυρές τάσεις αύξησης

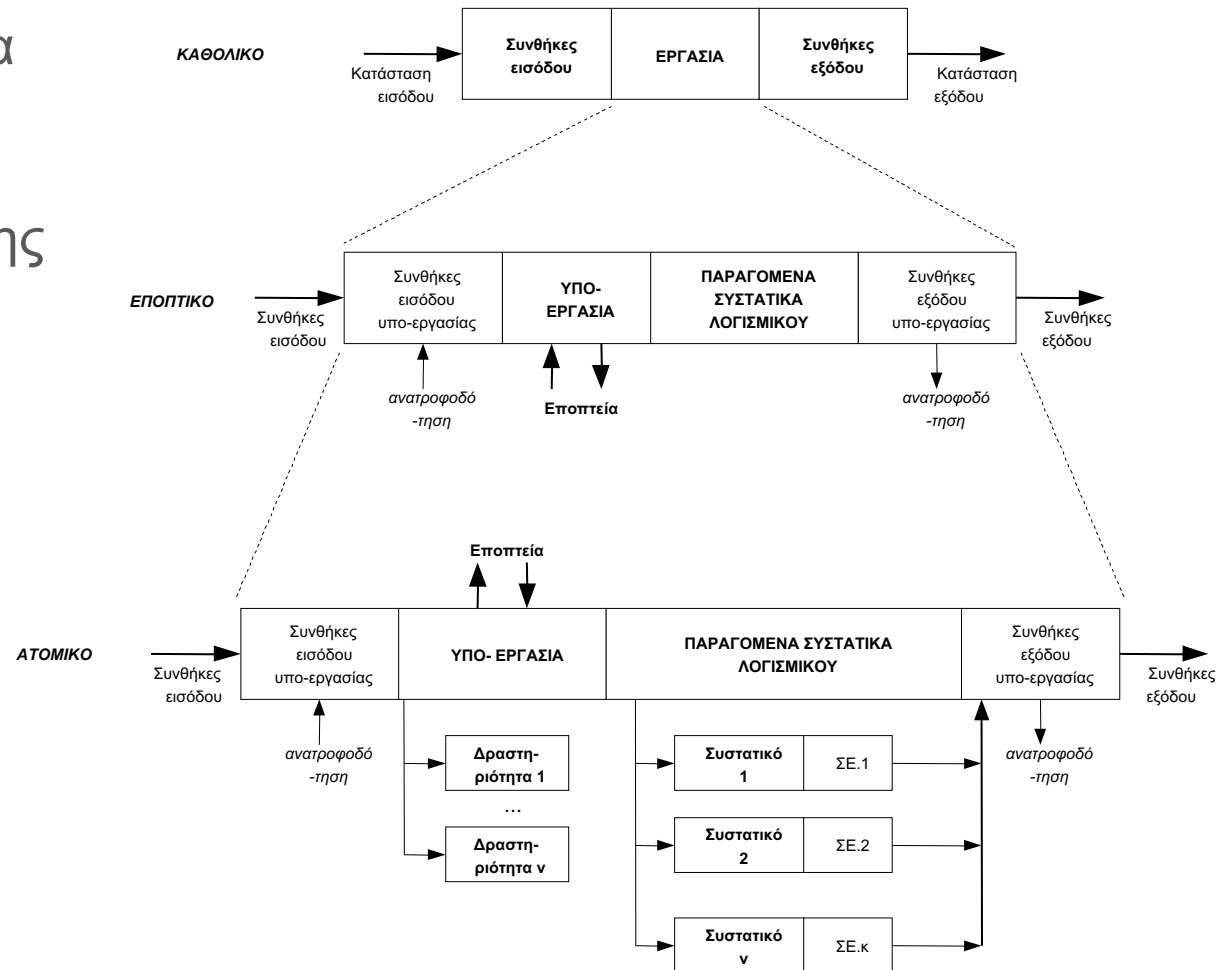
# Περιγραφή εργασιών και συστατικών στοιχείων λογισμικού

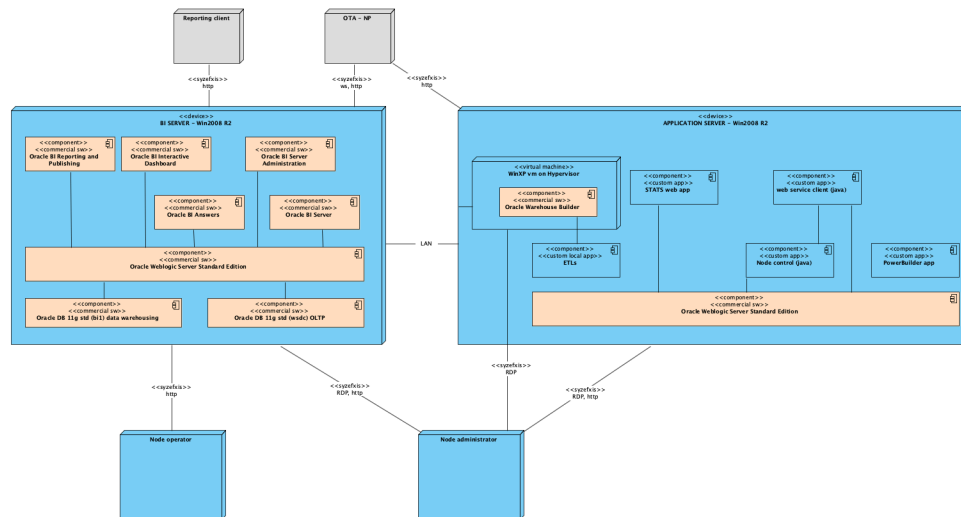
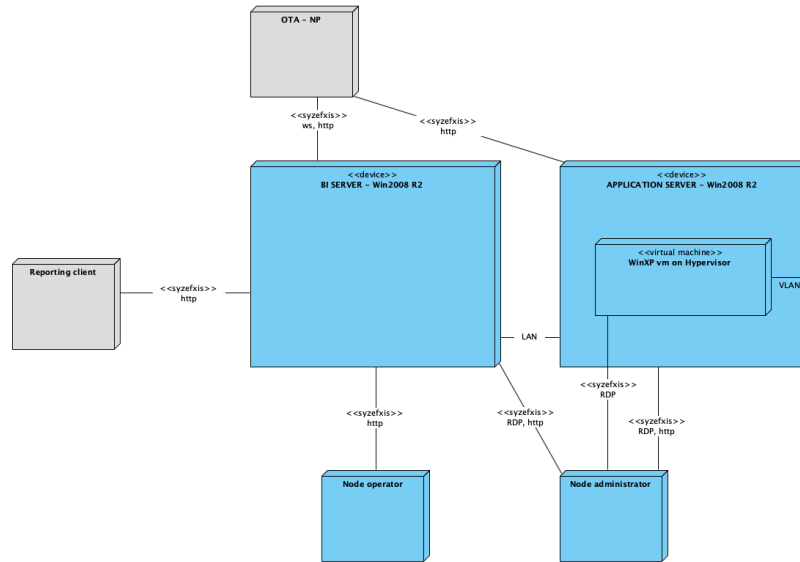
- Διαδοχικά επίπεδα λεπτομέρειας, ανάλογα με το σημείο εστίασης της προσοχής του παρατηρητή.



# Περιγραφή εργασιών και συστατικών στοιχείων λογισμικού

- Διαδοχικά επίπεδα λεπτομέρειας, ανάλογα με το σημείο εστίασης της προσοχής του παρατηρητή.





abstraction



# Περιγραφή συστατικών στοιχείων λογισμικού

**Μοντέλα**, δηλαδή αναπαραστάσεις κάποιων χαρακτηριστικών του λογισμικού

- Αφαιρετικά, δηλαδή δεν περιέχουν όλες τις λεπτομέρειες
- Συμπληρωματικά, δηλαδή πολλά μαζί περιγράφουν πλήρως το λογισμικό
- Υλοποιήσιμα, δηλαδή μπορούν να καθοδηγούν την κατασκευή λογισμικού

Δομημένο κείμενο (πρότυπα)

Συμβολισμοί

- Διαγράμματα
- Άλλοι τρόποι (π.χ. ψευδοκώδικας)



# Προβλήματα στην περιγραφή στοιχείων λογισμικού

Οι συμβολισμοί και οι ορισμοί είναι...

- Μη καθολικά αποδεκτοί
- Αντιληπτοί με διαφορετικά
- Με πλουραλισμό διατυπώσεων

Ανάγκη για Πρότυπα

- Παρέχουν τη δομή της περιγραφής του Λογισμικού
- Εξασφαλίζουν μια ελάχιστη πειθαρχία
- Αποτελούν ένα **κοινό σημείο αναφοράς** των κατασκευαστών

Το πρόβλημα των οργανισμών τυποποίησης

- IEEE, ACM, OMG, κ.ά.

# Συζήτηση

