Ενότητα 1

Εισαγωγή στην Ανάπτυξη Πληροφοριακών Συστημάτων

Μαθησιακοί στόχοι

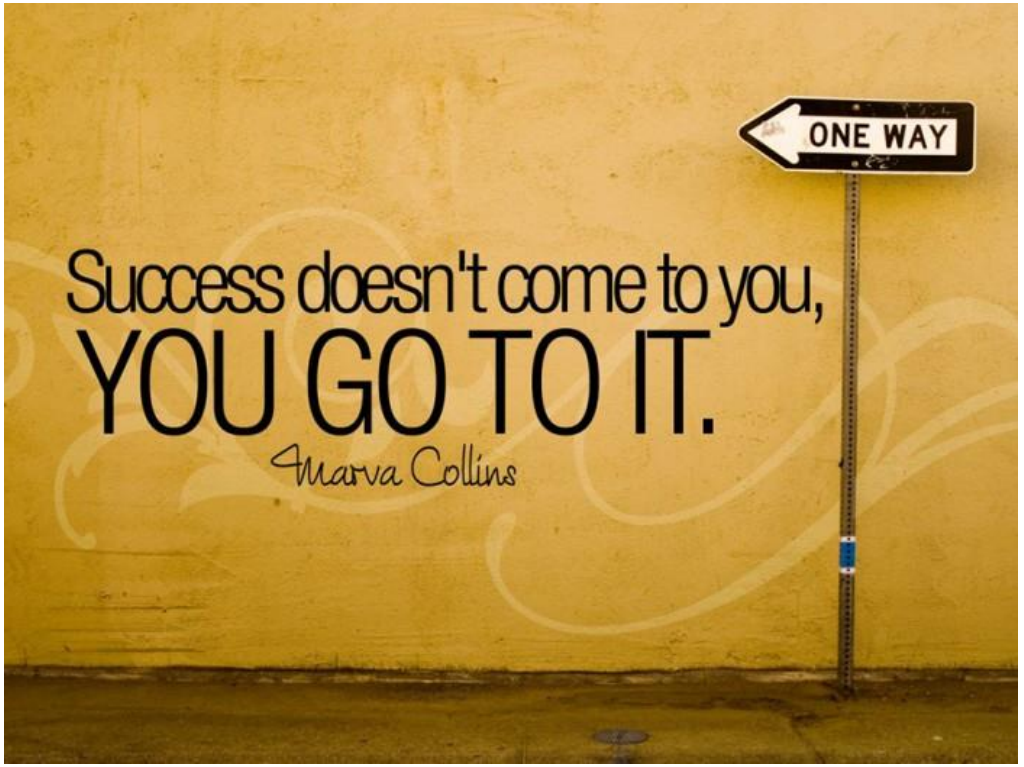
- Εισαγωγή στις έννοιες της ανάλυσης και σχεδίασης πληροφοριακών συστημάτων (ΠΣ)
- Αναφορά σε σύγχρονες προσεγγίσεις ανάλυσης και σχεδίασης ΠΣ
- Κατανόηση του ρόλου του Αναλυτή Συστημάτων
- Κατανόηση της έννοιας του Κύκλου Ζωής Ανάπτυξης ΠΣ
- Εξοικείωση με εναλλακτικές προσεγγίσεις του Κύκλου Ζωής Ανάπτυξης ΠΣ

Περιεχόμενα ενότητας

- Ο ρόλος του Αναλυτή Συστημάτων
- Ανάλυση και Σχεδίαση ΠΣ
- Κύκλος Ζωής Ανάπτυξης Πληροφοριακών Συστημάτων
- Μεθοδολογίες Ανάπτυξης Πληροφοριακών Συστημάτων
- CASE (Computer-Aided Software Engineering)

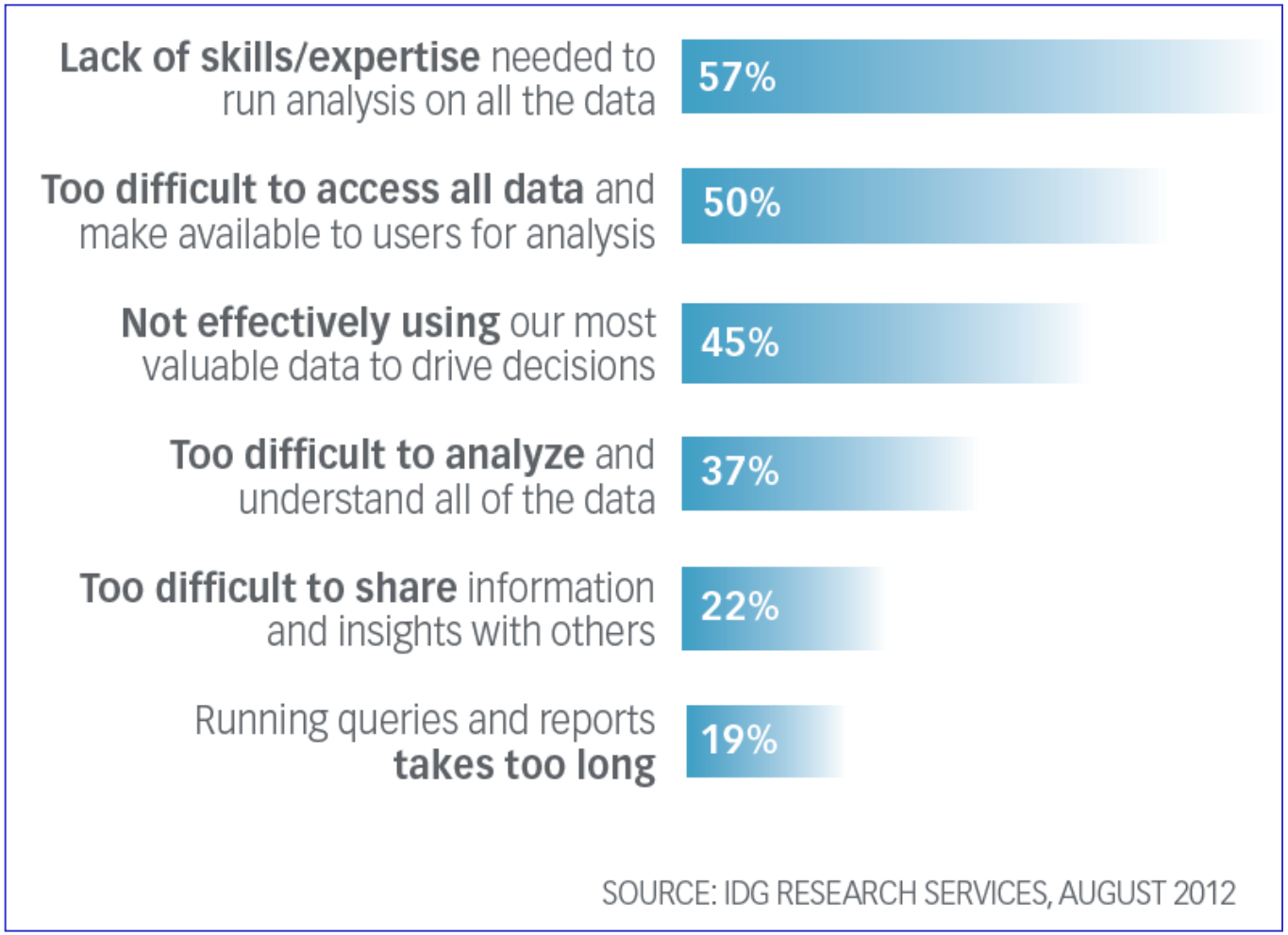
Συζήτηση - Γενικά

- Κλειδί επιτυχίας μιας επιχείρησης
  - Συλλογή δεδομένων
  - Επεξεργασία δεδομένων
  - Ερμηνεία δεδομένων
- Παραδείγματα
  - Εμπορική επιχείρηση
  - Πανεπιστήμιο Πατρών
    - Ψηφιακό Άλμα
    - E-class
    - Άλλα πληροφοριακά συστήματα



Πηγή: https://www.flickr.com/photos/98323572@N04/14926082682/

Συζήτηση – Σύγχρονες προκλήσεις



1.1: Ο Αναλυτής Συστημάτων

- Ο άνθρωπος-κλειδί για την ανάπτυξη ενός ΠΣ
  - Αναζήτηση και απόκτηση πληροφοριών
  - Ανάλυση απόδοσης συστήματος σε συνάρτηση με τους στόχους που είχε θέσει η διοίκηση
  - Ανάπτυξη και αξιολόγηση εναλλακτικών σεναρίων σχετικά με τη βελτίωση ή/και αναδιοργάνωση ενός συστήματος
  - Σχεδιασμός νέου συστήματος σύμφωνα με τις απαιτήσεις που καθορίστηκαν
  - Υλοποίηση νέου συστήματος

- Συνδυάζει παραδοσιακές ειδικότητες
  - Μηχανικού Μελέτης Εργασίας
  - Ειδικού Οργάνωσης & Μεθόδων
  - Επιχειρησιακού Ερευνητή

Ο ρόλος του Αναλυτή Συστημάτων (ΑΣ)

- Πολλαπλός ρόλος
  - Σύμβουλος, εκπαιδευτής, «μεταφορέας» απόψεων, επίλυση προβλημάτων, …

Προσόντα ενός Αναλυτή Συστημάτων

- Πρακτική γνώση τεχνικών επεξεργασίας πληροφοριών
  - Τεχνικές προγραμματισμού
  - Εργαλεία ανάπτυξης λογισμικού
  - Πακέτα εφαρμογών
  - Εξοπλισμός πληροφορικής
- Γενικές γνώσεις διοίκησης και διαχείρισης επιχειρήσεων (management)
  - Επικοινωνία με ειδικούς, κατανόηση προβλημάτων και αναγκών κάθε λειτουργίας
- Ικανότητες στην επίλυση προβλημάτων
  - Ανάλυση και επίλυση του προβλήματος
- Δημιουργικότητα, φαντασία, …

Αγγελία εργασίας για ΑΣ

Η Simon & Taylor, Inc., ένας κατασκευαστής κεριών, έχει μια άμεσα διαθέσιμη θέση εργασίας για έναν αναλυτή συστημάτων, στα γραφεία της στο Vermont.

Ο ιδανικός υποψήφιος θα έχει:

1. Πτυχίο στα πληροφοριακά συστήματα διοίκησης ή στην πληροφορική.
2. Δύο χρόνια εμπειρία σε UNIX/LINUX.
3. Εμπειρία σε C, Java, ή/και άλλες αντικειμενοστρεφείς γλώσσες προγραμματισμού, και σε περιβάλλοντα ανάπτυξης εφαρμογών, όπως τα Visual Studio ή IBM Rational Unified Process.
4. Δεξιότητες και εμπειρία με τοπικά δίκτυα (LAN).
5. Εξοικίωση με τις έννοιες της διανομής και παραγωγής (κατανομή, αναπλήρωση, διαχείριση καταστήματος, και προγραμματισμό παραγωγής).
6. Πρακτική εμπειρία διαχείρισης έργων και όλων των φάσεων του κύκλου ζωής ανάπτυξης συστημάτων.
7. Ισχυρές δεξιότητες επικοινωνίας.

Προσφέρουμε ανταγωνιστική αμοιβή, βοήθεια μετεγκατάστασης, και την πρόκληση να εργαστείτε σε ένα περιβάλλον αιχμής στην Τεχνολογία των Πληροφοριών.

Αποστείλετε το βιογραφικό σας στη διεύθυνση HR@simontaylor.com.

Η Simon & Taylor, Inc. είναι ένας εργοδότης ίσων ευκαιριών.

Πηγή: Valacich, George & Hoffer, "Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων (5η Έκδοση)", Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2015

Αναλυτές Συστημάτων vs. Προγραμματιστές

	ΑΣ	Προγραμματιστής
• Συμμετοχή		
– Μελέτη σκοπιμότητας	Πλήρης συμμετοχή	-
– Καθορισμός απαιτήσεων	Πλήρης συμμετοχή	-
– Σχεδιασμός	Ενεργός συμμετοχή	Μερική συμμετοχή
– Ανάπτυξη προγραμμάτων	Ενεργός συμμετοχή	Πλήρης συμμετοχή
– Εγκατάσταση	Ενεργός συμμετοχή	Μερική συμμετοχή
– Συντήρηση	Ενεργός συμμετοχή	Συμμετοχή (υλοποίηση)
• Σχέσεις με άλλους	Πολυάριθμες & πολύπλοκες	Περιορισμένες
• Συμμετοχή στις αποφάσεις	Καθοριστική	Χαμηλή έως ανύπαρκτη
• Προσόντα	Αναλυτική σκέψη Τεχνικές γνώσεις Επικοινωνία	Τεχνικές γνώσεις Πειθαρχία Υπομονή

Πηγή: Β. Λαοπόδη, "Ανάπτυξη Πληροφοριακών Συστημάτων - Ανάλυση και Σχεδιασμός Συστημάτων", Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1996.

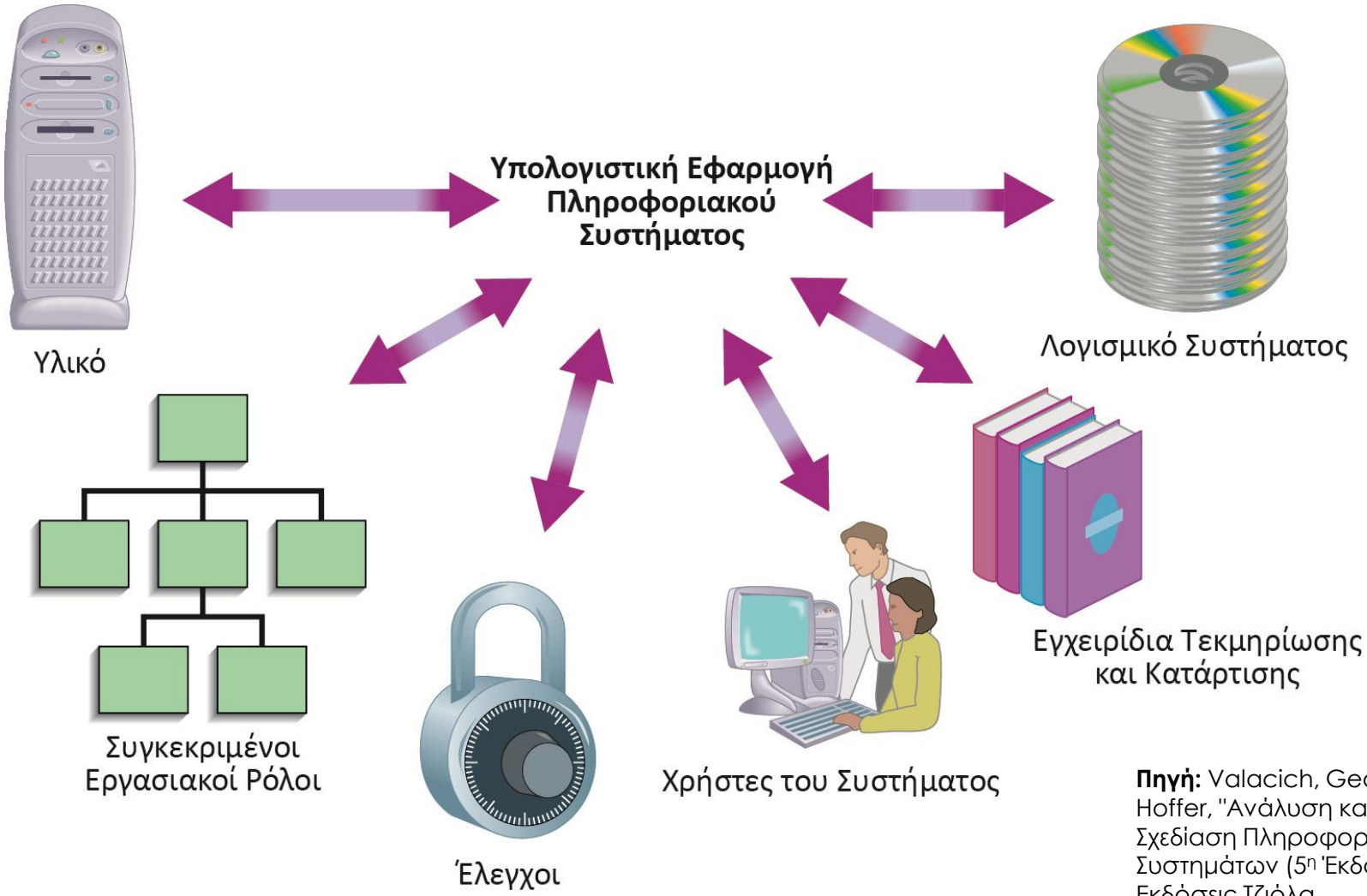
1.2: Ανάλυση και Σχεδίαση ΠΣ

- Διαδικασία ανάπτυξης και συντήρησης συστημάτων που εκτελούν βασικές επιχειρηματικές λειτουργίες

- Βασικός στόχος είναι η βελτίωση της απόδοσης της επιχείρησης με κατάλληλες εφαρμογές λογισμικού
  - Απαιτεί κατανόηση των στόχων, της δομής και των διεργασιών μιας επιχείρησης

- Δομημένη προσέγγιση
  - Ανάλυση
  - Σύνθεση

Συστατικά εφαρμογής ΠΣ

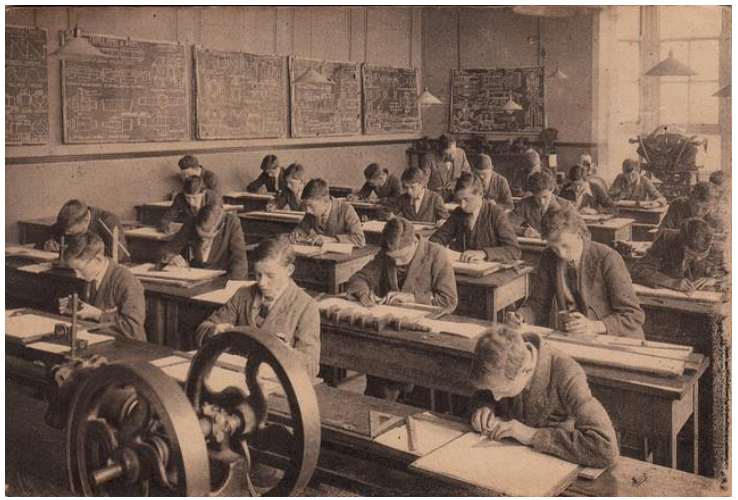


Πηγή: Valacich, George & Hoffer, "Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων (5η Έκδοση)", Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2015

Τεχνολογία Λογισμικού (Software Engineering)

- Η συστηματική εφαρμογή διαδικασιών, μεθοδολογιών, εργαλείων και τεχνικών για την επίτευξη των καθορισμένων απαιτήσεων ενός συστήματος λογισμικού

- Χτίσιμο μιας γέφυρας vs. «Χτίσιμο» ενός λειτουργικού συστήματος
  - Τι γίνεται σε περίπτωση κατάρρευσης (collapse);
  - Perfect vs. imperfect engineering
  - Πολυπλοκότητα
  - Συντήρηση



Industrious engineering students
Πηγή: https://www.flickr.com/photos/pellethepoet/12097798504/

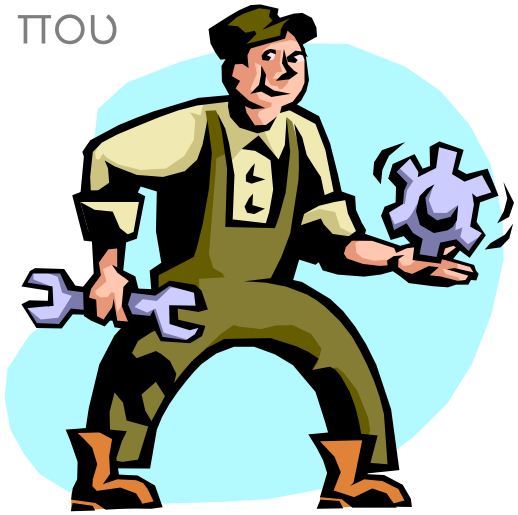
Δομή Τεχνολογίας Λογισμικού

Ανάπτυξη	Διαχείριση	Μέτρηση μεγεθών	Συντήρηση
• Ανάλυση • Σχεδίαση • Προγραμματισμός • Έλεγχος	• Σχεδίαση / Πρόβλεψη / Έλεγχος Έργου • Ποιοτική εξασφάλιση • Διαχείριση Σύνθεσης Συστήματος	• Αξιοπιστία • Επεκτασιμότητα • Ευχρηστία • Ευελιξία • Επαναχρησιμοποίηση	• Διόρθωση λαθών • Προσαρμογή • Τελειοποίηση

Πηγή: Β. Λαοπόδη: "Ανάπτυξη Πληροφοριακών Συστημάτων - Ανάλυση και Σχεδιασμός Συστημάτων", Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1996.

Τεχνολογία Λογισμικού (συν.)

- Παράδειγμα: Οικονομικά θέματα
  - Αξίζει να θέσουμε σε εφαρμογή μια νέα μέθοδο κωδικοποίησης ΜΕΚ2, η οποία είναι 10% γρηγορότερη από τη μέθοδο ΜΕΚ1 που εφαρμόζουμε ως σήμερα;



Κοινή λογική:

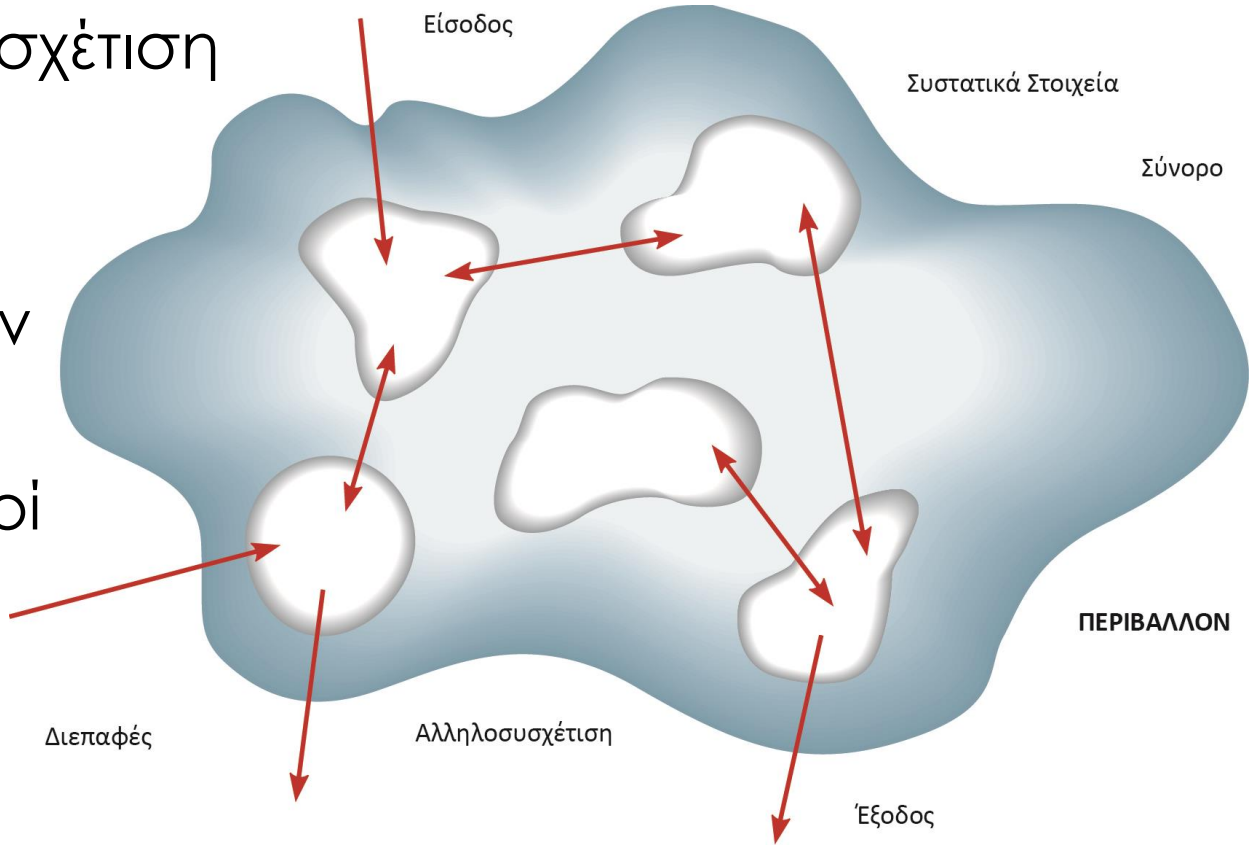
- Φυσικά!

Τεχνολόγος Λογισμικού:

- Τι επιπτώσεις θα έχει μια τέτοια απόφαση σε θέματα συντήρησης του λογισμικού;

Χαρακτηριστικά ΠΣ

- Συστατικά στοιχεία
- Αλληλοσυσχέτιση
- Σύνορο
- Σκοπός
- Περιβάλλον
- Διεπαφές
- Περιορισμοί
- Είσοδοι
- Έξοδοι



Πηγή: Valacich, George & Hoffer, "Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων (5η Έκδοση)", Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2015

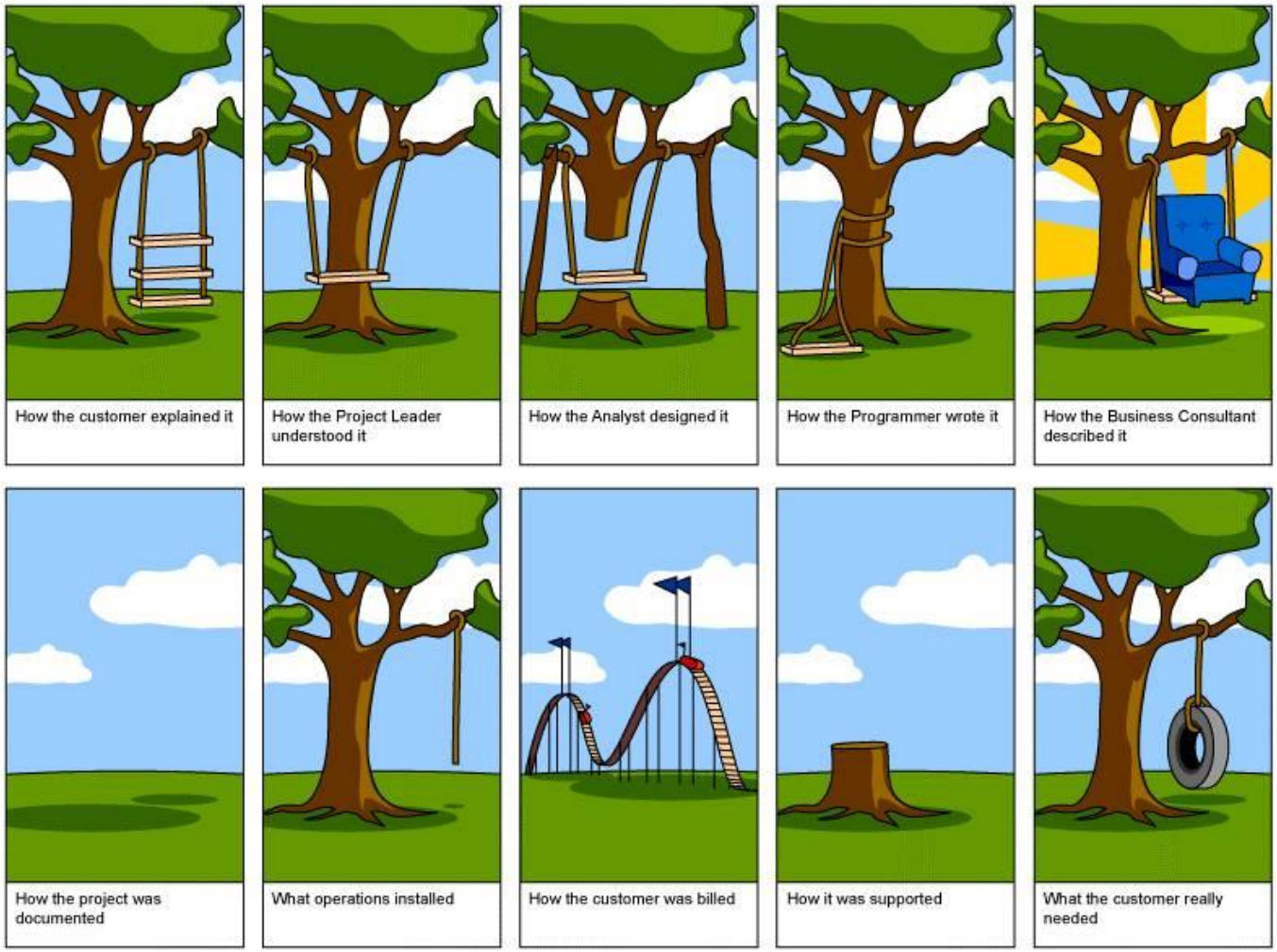
ΠΣ – Βασικές έννοιες

- Αποσύνθεση (decomposition)
  - Η διεργασία διάσπασης ενός συστήματος στα μικρότερα συστατικά του
    - Μικρότερα, καλύτερα διαχειρίσιμα και κατανοητά υποσυστήματα
    - Εστίαση σε συγκεκριμένη ομάδα χρηστών, σε συγκεκριμένο υποσύστημα, παράλληλη ανάπτυξη …
- Τμηματικότητα (modularity)
  - Η υποδιαίρεση ενός συστήματος σε τμήματα σχετικά ομοιόμορφου μεγέθους
    - The degree to which a system's components may be separated and recombined (wikipedia)
- Σύζευξη (coupling)
  - Ο βαθμός αλληλεξάρτησης μεταξύ υποσυστημάτων
- Συνεκτικότητα (cohesion)
  - Ο βαθμός συσχέτισης τμημάτων λειτουργικότητας σε ένα υποσύστημα

1.3: Κύκλος Ζωής Ανάπτυξης ΠΣ

- Η πορεία που ακολουθεί η ανάπτυξη ενός συστήματος από τη φάση του εντοπισμού του προβλήματος μέχρι τη λειτουργία του κατάλληλου ΠΣ
- Βασικές φάσεις
  - Ανάλυση, σχεδίαση, υλοποίηση και συντήρηση
- Κάθε φάση έχει συγκεκριμένο σκοπό και παραδοτέα
- Κάθε επιχείρηση τροποποιεί το μοντέλο του κύκλου ζωής σύμφωνα με τις ανάγκες της

Ανάπτυξη ΠΣ



Πηγή: http://dayjob.kimhooperwrites.com/2012/04/16/the-ad-agency-project-life-cycle/

Κύκλος Ζωής Λογισμικού (Software Life Cycle)

- Ο τρόπος που παράγουμε λογισμικό. Περιλαμβάνει:
  - μοντέλο κύκλου ζωής
  - managers, ειδικούς, προγραμματιστές, ...
  - εργαλεία ανάπτυξης λογισμικού (CASE)

- Φάσεις
  - Ανάλυσης Απαιτήσεων (Requirements)
  - Καθορισμού Προδιαγραφών (Specificati
  - Σχεδίασης (Design)
  - Υλοποίησης (Implementation)
  - Συνένωσης Κώδικα (Integration)
  - Συντήρησης (Maintenance)
  - Απόσυρσης (Retirement)

Κύκλος Ζωής Λογισμικού – Βασικές έννοιες

- Έλεγχος (testing)
  - Επαλήθευση (verification)
    - Στο τέλος κάθε φάσης
  - Επικύρωση (validation)
    - Πριν την παράδοση του προϊόντος στον πελάτη
- Τεκμηρίωση (documentation)
  - Πότε;

Ανάλυση Απαιτήσεων

- Προϋπόθεση
  - Η διαδικασία θεωρείται εύλογη οικονομικά
- Αναλυτική διερεύνηση του προβλήματος
  - Κατανόηση του τι χρειάζεται και όχι τι θέλει ο πελάτης
  - Καταγραφή περιορισμών (constraints)
- Περαιτέρω ανάλυση και "εκλέπτυνση" του προβλήματος
  - Είναι το προϊόν τεχνικά εφικτό;
  - Γίνεται στα πλαίσια του budget του πελάτη;
- Βασικός έλεγχος φάσης απαιτήσεων
  - Γρήγορο πρωτότυπο (prototype)
- Διασφάλιση Ποιότητας Λογισμικού (software quality)

Καθορισμός Προδιαγραφών

- Τα σχετικά έγγραφα έχουν νομική υπόσταση
  - Πρέπει να συμπληρωθούν έτσι ώστε να καθορίζουν επακριβώς το πρόβλημα
- Οι προδιαγραφές δεν πρέπει να είναι:
  - Διφορούμενες, ημιτελείς, αντιφατικές
- Μετά τη συμπλήρωση των σχετικών εγγράφων
  - Αναλυτικός σχεδιασμός παραγωγής και υπολογισμός κόστους
- Software product management plan (SPMP)
  - Παραδοτέα (deliverables), βασικά χρονικά σημεία (milestones), budget
- Έλεγχος φάσης
  - Σωστή δόμηση
  - Ανασκόπηση

Σχεδίαση

- Βασική διαφορά:
  - Προδιαγραφές → τι;
  - Σχεδίαση → πως;
- Καθορισμός δομικών ενοτήτων (modules) συστήματος και της μεταξύ τους επικοινωνίας (architectural/gross design)
- Επιλογή αλγορίθμων, δομών δεδομένων και ροών δεδομένων (detailed design)
- Καταγραφή των σχετικών αποφάσεων
  - Περιπτώσεις dead-end (backtrack and redesign)
  - Συντήρηση
  - Ιδανικά, η σχεδίαση θα πρέπει να είναι "open-ended"
- Έλεγχος φάσης
  - Αντιστοίχιση με προδιαγραφές
  - Αναζήτηση λαθών λογικής, διαπροσωπείας (interface) μεταξύ των ενοτήτων, απουσίας exception handling, ...

Υλοποίηση

- Υλοποίηση της αναλυτικής σχεδίασης
  - Γράψιμο κώδικα
- Τεκμηρίωση
  - Σχόλια (comments) στον κώδικα
  - Επιπλέον: ανάλυση περιπτώσεων, αποτελέσματα, ...
- Έλεγχος φάσης
  - Ανασκόπηση κώδικα (code review)
  - Test cases
  - Άτυπος έλεγχος (informal testing)
    - desk checking
  - Τυπικός έλεγχος (formal testing)
    - Τμήμα Διασφάλισης Ποιότητας Λογισμικού (Software Quality Assurance Group)

Συνένωση Κώδικα

- Συνένωση δομικών ενοτήτων λογισμικού και έλεγχος σωστής λειτουργίας του προϊόντος ως ενιαία οντότητα
- Υλοποίηση και συνένωση κώδικα πρέπει να γίνονται παράλληλα
- Έλεγχος φάσης
  - Λειτουργικότητα έναντι προδιαγραφών, περιορισμοί, ορθότητα, αξιοπιστία, "ευρωστία", συμβατότητα με άλλο λογισμικό, ...
  - Έλεγχος αποδοχής (acceptance testing)
    - στον πελάτη, με πραγματικά δεδομένα

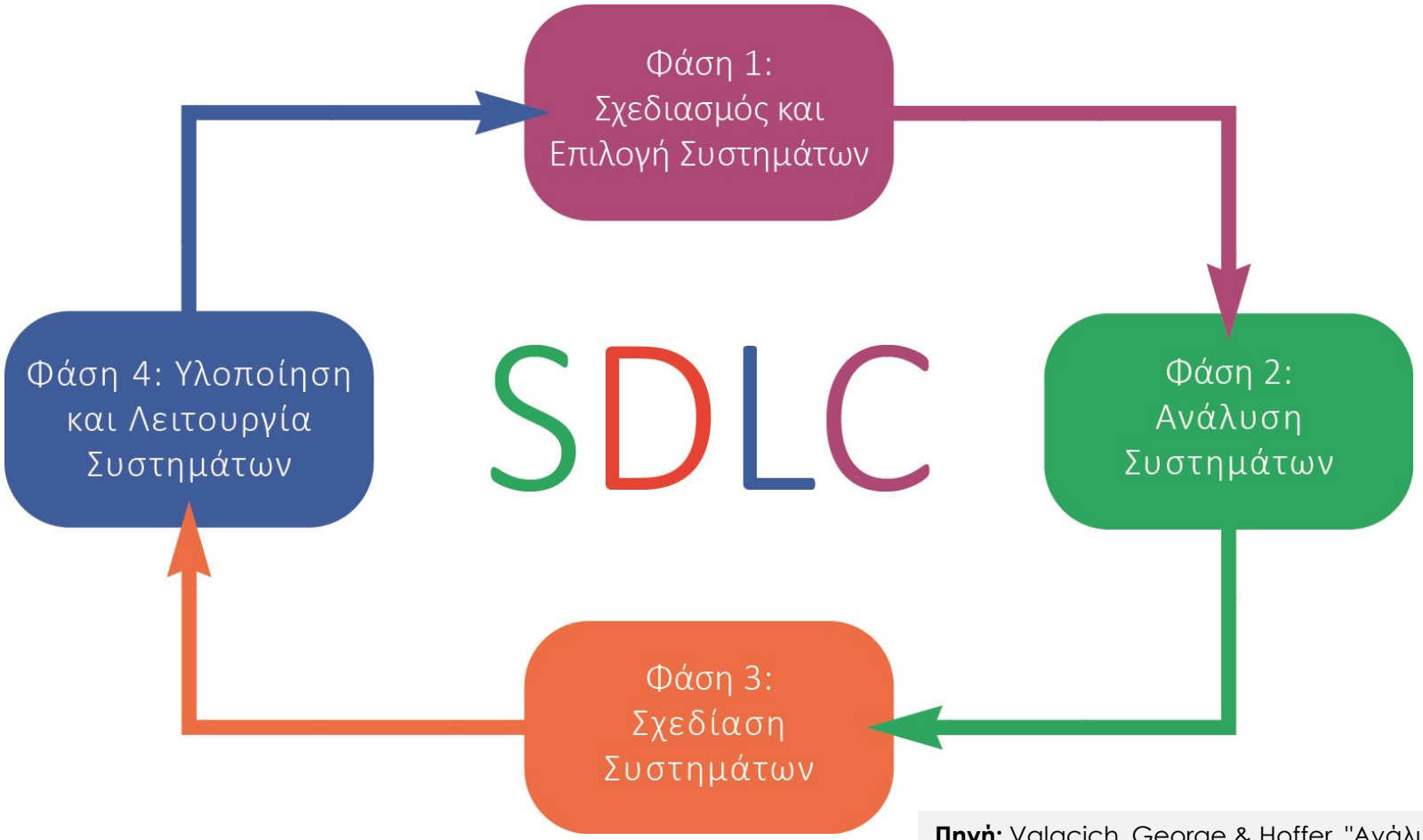
Συντήρηση

- Περιλαμβάνει οποιαδήποτε αλλαγή αφότου ο πελάτης έχει παραλάβει το προϊόν λογισμικού
- Πρέπει πάντα να θεωρείται ως αναπόσπαστο τμήμα της διαδικασίας ανάπτυξης λογισμικού
- Κοστίζει περισσότερο απ' ότι όλες οι άλλες φάσεις μαζί
- Βασικό πρόβλημα η έλλειψη (καλής) τεκμηρίωσης
- Έλεγχοι φάσης
  - Όχι μόνο αν έγιναν οι απαιτούμενες αλλαγές, αλλά και αν αυτές επέφεραν κάποιες άλλες, ανεπιθύμητες αλλαγές (regression testing)

Απόσυρση

- Το «καλό» λογισμικό συντηρείται
- Το λογισμικό γράφεται από την αρχή όταν:
  - απαιτούνται δραστικές αλλαγές στη σχεδίαση του τελικού προϊόντος
  - η συντήρηση καταντά αδύνατη ή ασύμφορη
  - αντικατασταθεί το hardware ή λειτουργικό σύστημα
  - η τεκμηρίωση είναι ανύπαρκτη, ελλιπής ή ανακριβής
- Πραγματική (ολική) απόσυρση ενός προϊόντος είναι σπάνιο φαινόμενο

Κύκλος Ζωής Ανάπτυξης Συστημάτων



Πηγή: Valacich, George & Hoffer, "Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων (5η Έκδοση)", Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2015

Κύκλος Ζωής Ανάπτυξης Συστημάτων (συν.)

- Προγραμματισμός και Επιλογή Συστημάτων
  - Σχεδιασμός → Προγραμματισμός (planning)
  - Καθορισμός της ανάγκης για ένα νέο ή βελτιωμένο ΠΣ
  - Διερεύνηση και προσδιορισμός του σκοπού του προτεινόμενου συστήματος

- Ανάλυση Συστημάτων
  - Μελέτη των υφιστάμενων διαδικασιών και πληροφοριακών συστημάτων
    - Προσδιορισμός των απαιτήσεων του ΠΣ
    - Παραγωγή εναλλακτικών σχεδίων που ικανοποιούν τις απαιτήσεις
    - Σύγκριση εναλλακτικών
    - Σύσταση της βέλτιστης εναλλακτικής

Κύκλος Ζωής Ανάπτυξης Συστημάτων (συν.)

- Σχεδίαση Συστήματος
  - Λογική Σχεδίαση
    - Ανεξάρτητη από συγκεκριμένη πλατφόρμα υλικού ή λογισμικού
    - Επικεντρώνεται στις επιχειρηματικές πτυχές του συστήματος (στο πως το ΠΣ θα επηρεάσει την επιχείρηση)
  - Φυσική Σχεδίαση
    - Τεχνικές Προδιαγραφές

- Υλοποίηση και Λειτουργία Συστημάτων
  - Συγγραφή κώδικα
  - Τεκμηρίωση
  - Εγκατάσταση υλικού και λογισμικού
  - Κατάρτιση χρηστών
  - Συντήρηση (αλλαγές, βελτιώσεις, προσθήκες …)

Έξοδοι φάσεων του SDLC

Φάση	Προϊόντα, έξοδοι, ή παραδοτέα
Σχεδιασμός και επιλογή συστημάτων	Προτεραιότητες για τα συστήματα και τα έργα.
	Αρχιτεκτονική για δεδομένα, δίκτυα, υλικό, και διαχείριση ΠΣ.
	Λεπτομερές πλάνο εργασιών για το επιλεγμένο έργο.
	Ορισμός του σκοπού του συστήματος.
	Αιτιολόγηση του συστήματος ή επιχειρηματική πρόταση.
Ανάλυση συστημάτων	Περιγραφή του υπάρχοντος συστήματος.
	Γενική πρόταση για τη διόρθωση, αναβάθμιση, ή αντικατάσταση του υπάρχοντος συστήματος.
	Εξήγηση των εναλλακτικών συστημάτων και αιτιολόγηση της επιλεγμένης εναλλακτικής λύσης.
	Πλάνο προμήθειας νέου τεχνολογικού εξοπλισμού.
Σχεδίαση συστημάτων	Λεπτομερείς προδιαγραφές όλων των στοιχείων του συστήματος (δεδομένα, διεργασίες, είσοδοι, και έξοδοι).
Υλοποίηση και λειτουργία συστημάτων	Κώδικας.
	Τεκμηρίωση.
	Διαδικασίες κατάρτισης και δυνατότητες υποστήριξης.
	Νέες εκδόσεις ή διανομές του λογισμικού με τις αντίστοιχες ενημερώσεις στην τεκμηρίωση, στην κατάρτιση, και στην υποστήριξη.

Πηγή: Valacich, George & Hoffer, "Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων (5η Έκδοση)", Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2015

1.4: Μεθοδολογίες Ανάπτυξης ΠΣ

- Μέθοδος
  - Κανονικός και συστηματικός τρόπος ή μέσο για να εκτελεσθεί μια εργασία ακολουθώντας μια σειρά διαδικασιών, σύμφωνα με ένα αναλυτικό και λογικά διαταγμένο σχέδιο
- Μεθοδολογία
  - Το σύστημα αρχών, πρακτικών και διαδικασιών που εφαρμόζονται σε ένα συγκεκριμένο κλάδο γνώσης
- ΑΣ μέσω μεθοδολογίας μπορεί να:
  - Διερευνήσει συστηματικά μια επιχείρηση
  - Τεκμηριώσει πλήρως και με ακρίβεια την παραπάνω διερεύνηση
  - Εξάγει τεκμηριωμένα συμπεράσματα και προβεί σε συστάσεις
  - Σχεδιάσει το κατάλληλο ΠΣ

Αντικειμενοστραφής vs. Δομημένη Προσέγγιση

- Η δομημένη προσέγγιση (structured paradigm) είχε αρχικά μεγάλη απήχηση
  - Επιτρεπόμενες δομές ελέγχου: συνέχειας (sequence), επιλογής (selection) και επανάληψης (repetition)

- Άρχισε να αποτυγχάνει όσο τα προϊόντα λογισμικού μεγάλωναν (> 50.000 γραμμές κώδικα)
  - Προβλήματα συντήρησης (φτάνουν το 80% της ολικής προσπάθειας σήμερα)

- Αιτία: οι δομημένες μέθοδοι είναι:
  - "action oriented", π.χ. διαγράμματα ροής δεδομένων, ή
  - "data oriented", π.χ. entity-relationship diagrams

  αλλά όχι και τα δύο μαζί

Αντικειμενοστραφής vs. Δομημένη Προσέγγιση (συν.)

- Αντικειμενοστραφής προσέγγιση
  - Data και actions είναι εξίσου σημαντικά
  - Αντικείμενο (object): Συστατικό του λογισμικού που "αντιμετωπίζει" ταυτόχρονα τα δεδομένα (data) και τις διεργασίες (actions) που λαμβάνουν χώρα πάνω σε αυτά τα δεδομένα

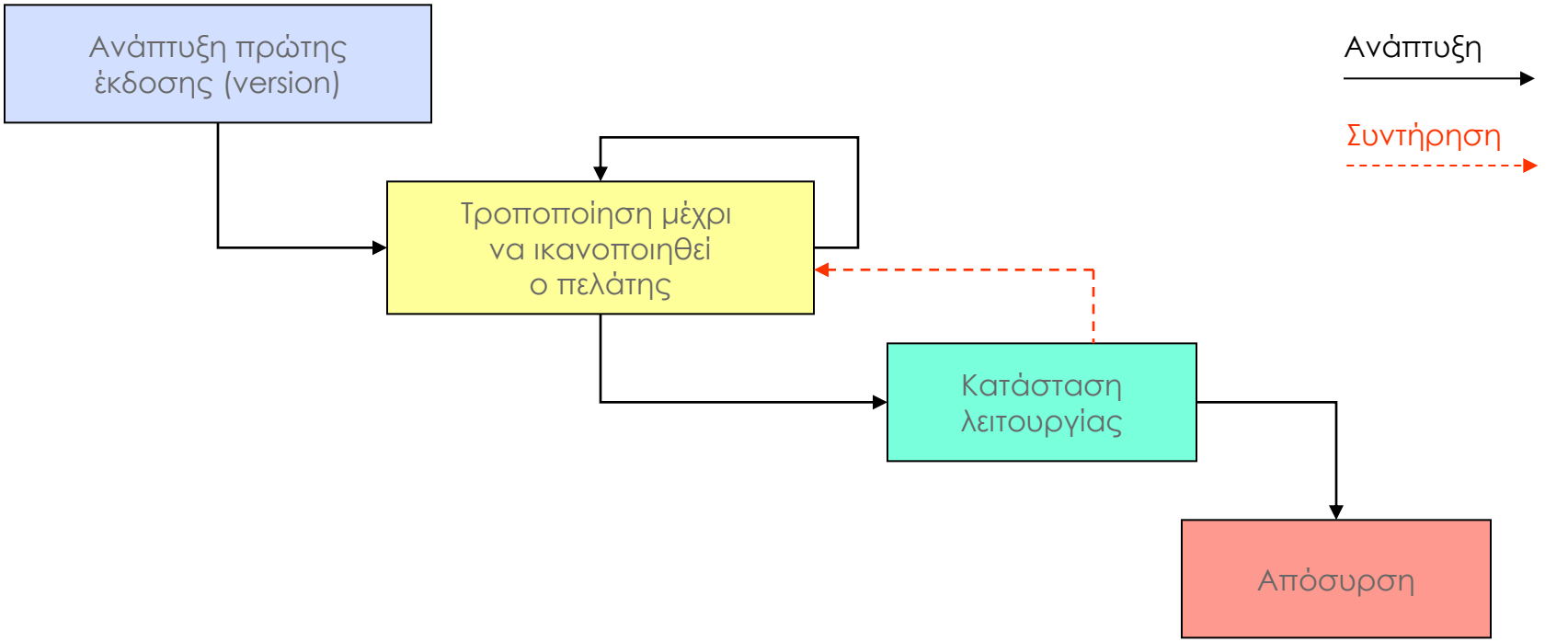
- Παράδειγμα: Τραπεζικός λογαριασμός
  - data: account balance
  - actions: deposit, withdraw, determine balance

- Δομημένη προσέγγιση:
  - απότομη μετάβαση ανάμεσα στην ανάλυση (τι θα φτιαχτεί) και το σχεδιασμό (πώς θα φτιαχτεί)
- Αντικειμενοστραφής προσέγγιση:
  - η έννοια του αντικειμένου εισέρχεται νωρίς, από τις πρώτες φάσεις του κύκλου ζωής του προϊόντος

Μοντέλα Κύκλου Ζωής Λογισμικού - Γενικά

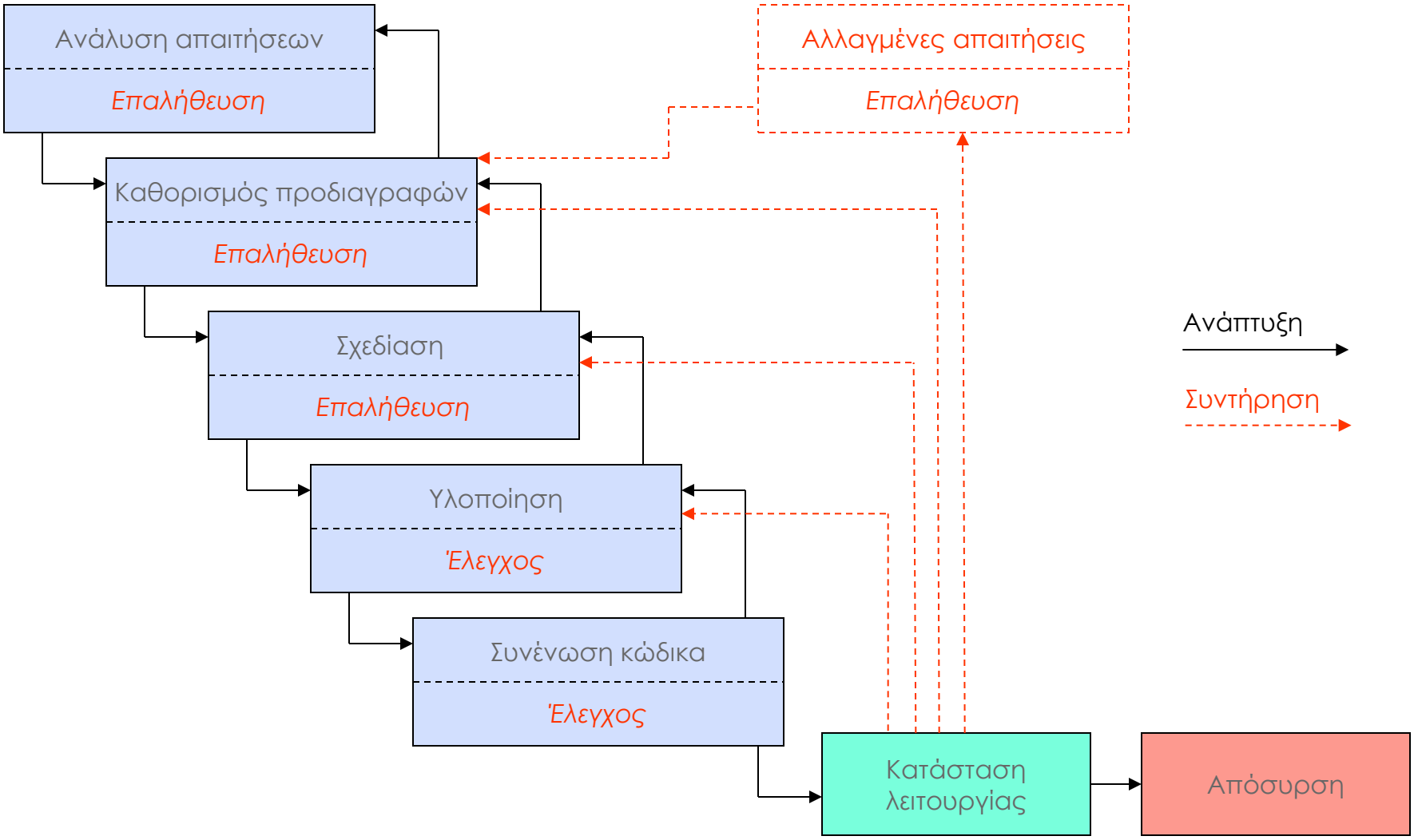
- Αφορούν τη σειρά των βημάτων μέσω των οποίων ένα προϊόν λογισμικού αναπτύσσεται
- Παρέχουν οδηγίες για τις εργασίες που πρέπει να γίνουν, τη χρονική τους σειρά και τα κριτήρια μετάβασης από τη μία στην άλλη
- Υπάρχει ποικιλία μοντέλων
  - Όλα έχουν δυνατά σημεία και αδύνατα σημεία
  - Κριτήρια επιλογής
    - οργάνωση
    - management
    - προσωπικό
    - προϊόν

Μοντέλο “build and fix”



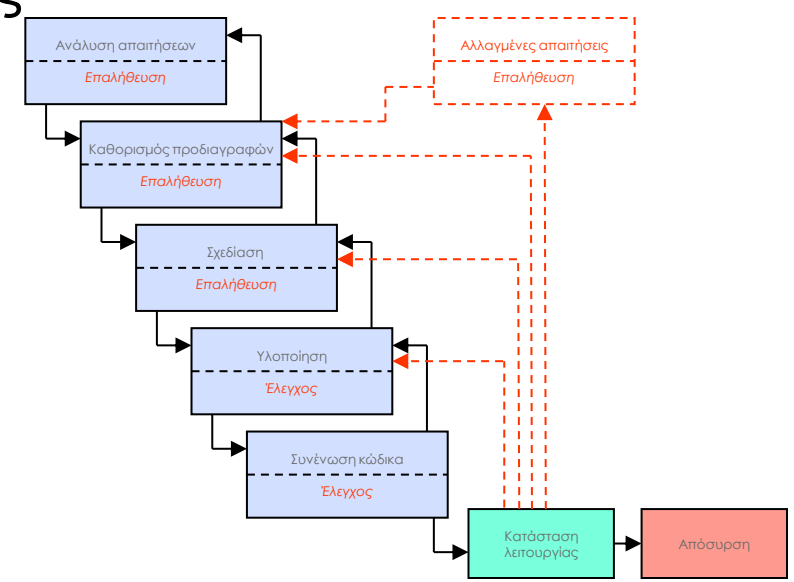
- Δεν γίνεται καθορισμός προδιαγραφών
- Δεν γίνεται σχεδίαση
- Δουλεύει καλά σε περιπτώσεις που το λογισμικό δεν ξεπερνά τις 100-200 γραμμές κώδικα

Μοντέλο καταρράκτη (waterfall model)

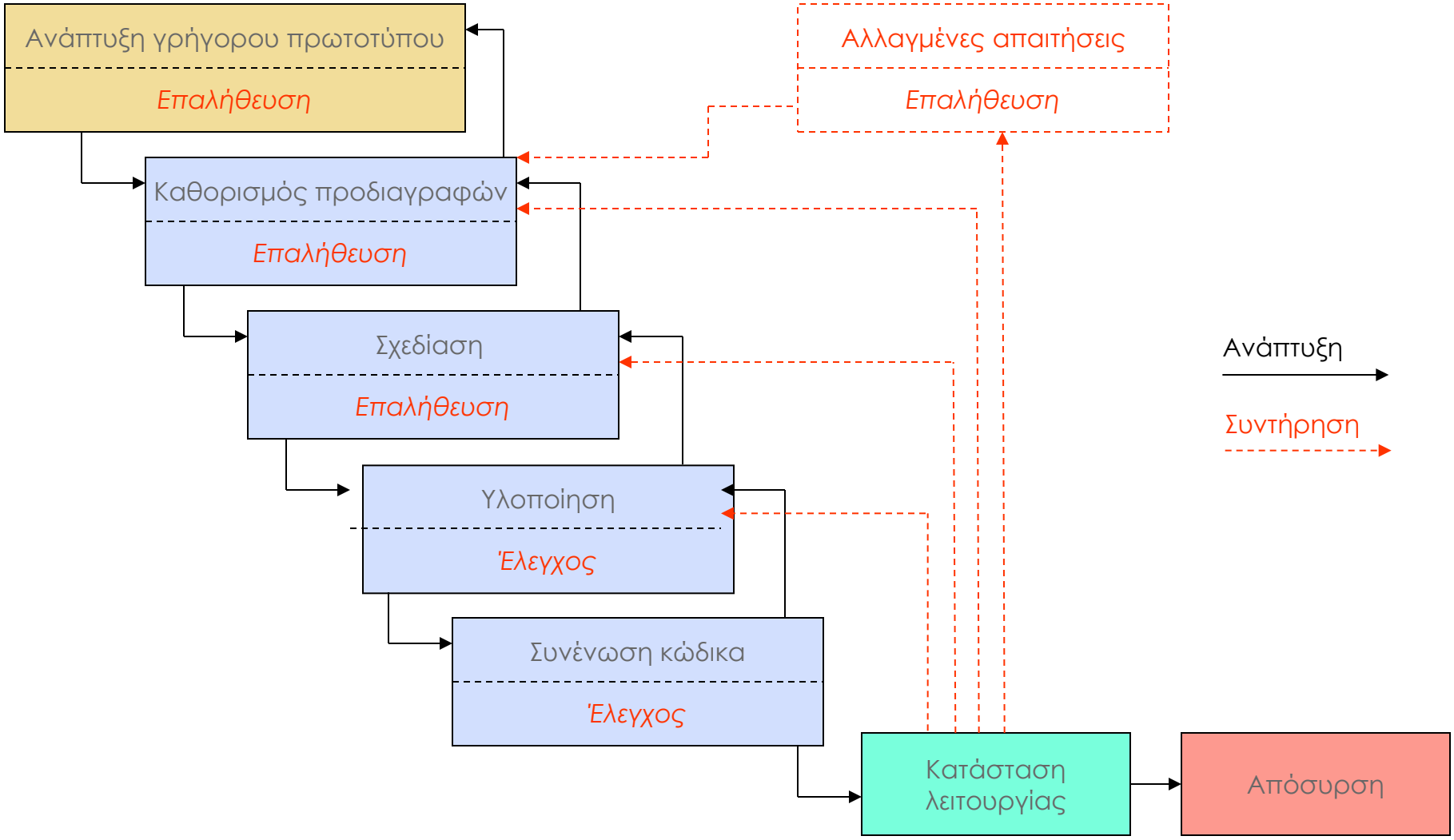


Μοντέλο καταρράκτη (συν.)

- Βρόχοι ανατροφοδότησης (feedback loops)
- «Οδηγείται» από την τεκμηρίωση (documentation-driven)
- Καμία φάση δεν θεωρείται τελειωμένη πριν γραφεί και εγκριθεί η τεκμηρίωσή της
- Πλεονεκτήματα
  - τεκμηρίωση
  - ευκολότερη συντήρηση
- Μειονεκτήματα
  - Ορισμοί Προδιαγραφών



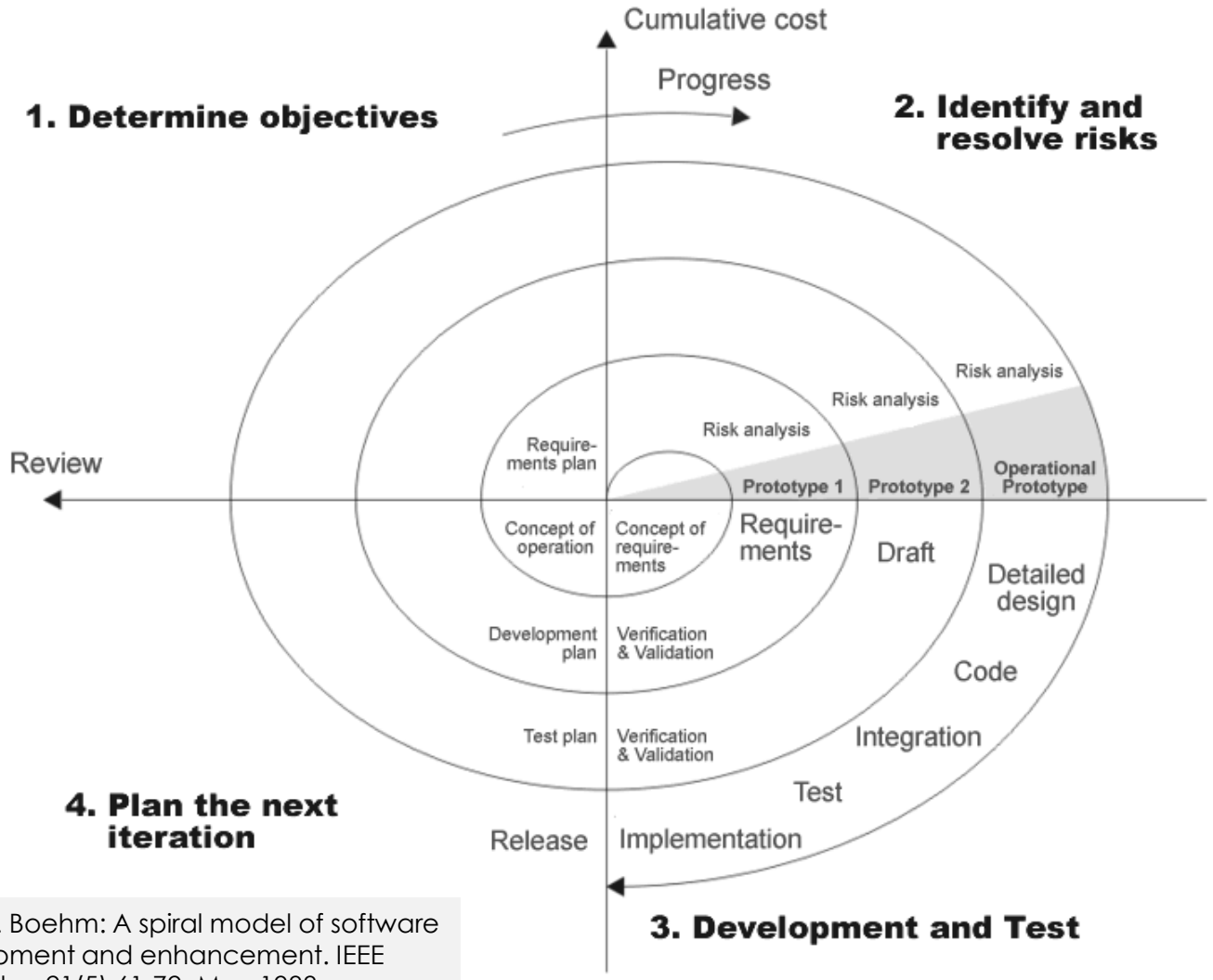
Μοντέλο ανάπτυξης γρήγορου πρωτοτύπου



Μοντέλο ανάπτυξης γρήγορου πρωτοτύπου (συν.)

- Οι χρήστες συμμετέχουν στη σχεδίαση
  - Η ανάπτυξη του προϊόντος είναι γραμμική
- Ανάπτυξη γρήγορου πρωτοτύπου (rapid prototype) → γρήγορη μετατροπή
- Βασικά σημεία
  - Δεν φτιάχνεται το προϊόν αυτό καθεαυτό
  - Η ανάπτυξη γρήγορου πρωτοτύπου μπορεί να αντικαταστήσει τη φάση του καθορισμού προδιαγραφών, αλλά ποτέ αυτή της σχεδίασης
  - Μοντέλο καταρράκτη → κάνε κάτι σωστά με την πρώτη
  - Γρήγορο πρωτότυπο → συχνές αλλαγές (και μετά απόσυρση)

Μοντέλο σπιράλ



Πηγή: B. Boehm: A spiral model of software development and enhancement. IEEE Computer, 21(5):61-72, May 1988.

- Έμφαση σε risk analysis → λιγότερο ρίσκο κατά την ανάπτυξη του ΠΣ
- Κατάλληλο για projects μεγάλης κλίμακας
- Συνήθως δαπανηρό μοντέλο

Ευέλικτη ανάπτυξη λογισμικού (agile development)

- Πρώτη προτεραιότητα είναι η ικανοποίηση του πελάτη μέσω της έγκαιρης και συνεχούς παράδοσης χρηστικού λογισμικού
- Οι αλλαγές στις απαιτήσεις είναι ευπρόσδεκτες, ακόμα και σε προχωρημένα στάδια της ανάπτυξης
- Παραδίδεται συχνά λογισμικό που λειτουργεί, σε διαστήματα μερικών εβδομάδων ή μηνών (με προτίμηση στη συντομότερη δυνατή χρονική κλίμακα)
- Το λογισμικό που λειτουργεί είναι το βασικό μέτρο προόδου
  - Όχι έμφαση στην τεκμηρίωση
- Συνομιλίες πρόσωπο-με-πρόσωπο μεταξύ των ατόμων της ομάδας ανάπτυξης λογισμικού
  - Διαρκής και άμεση επαφή με πελάτες

- Βλέπε http://agilemanifesto.org

Άλλες προσεγγίσεις

- Rapid Application Development – RAD
  - Γρήγορη Ανάπτυξη Εφαρμογών
  - Ξεκαθάρισμα απαιτήσεων μέσω πρωτοτυποποίησης
- Participatory Design
  - Συμμετοχική Σχεδίαση
  - Έμφαση στο ρόλο του χρήστη
- Joint application design – JAD
  - Συλλογική Σχεδίαση Εφαρμογών
- Rational Unified Process – RUP
- …

Συγκριτική θεώρηση μοντέλων

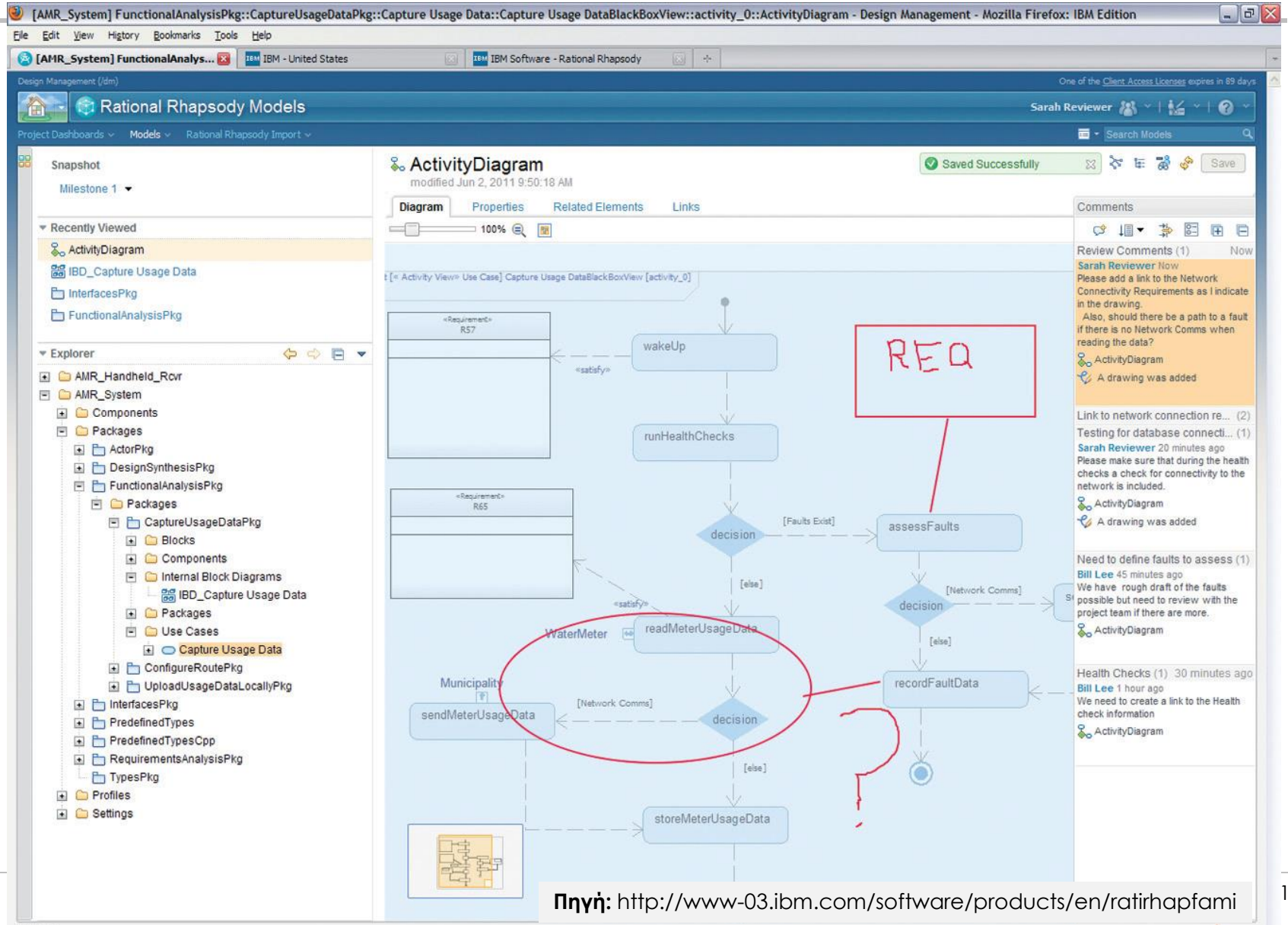
Καλύτερη λύση είναι (σχεδόν πάντα) ένα "mix-and-match" μοντέλο

Πηγή: https://flic.kr/p/8rH1UA

1.5: Computer-Aided Software Engineering

- Σκοπός
  - Υποστήριξη σε κάθε φάση του κύκλου ζωής
  - Είναι computer-aided και όχι computer-automated
- Τύποι εργαλείων
  - Εργαλεία κατασκευής διαγραμμάτων
  - Γεννήτριες απεικόνισης οθονών και αναφορών
  - Εργαλεία ανάλυσης και ελέγχου
  - Αποθετήρια (repositories)
  - Γεννήτριες υλικού τεκμηρίωσης
  - Γεννήτριες κώδικα
- Πλεονεκτήματα
  - Ταχύτερη ανάπτυξη προϊόντος
  - Λιγότερα λάθη
  - Ευκολότερη συντήρηση

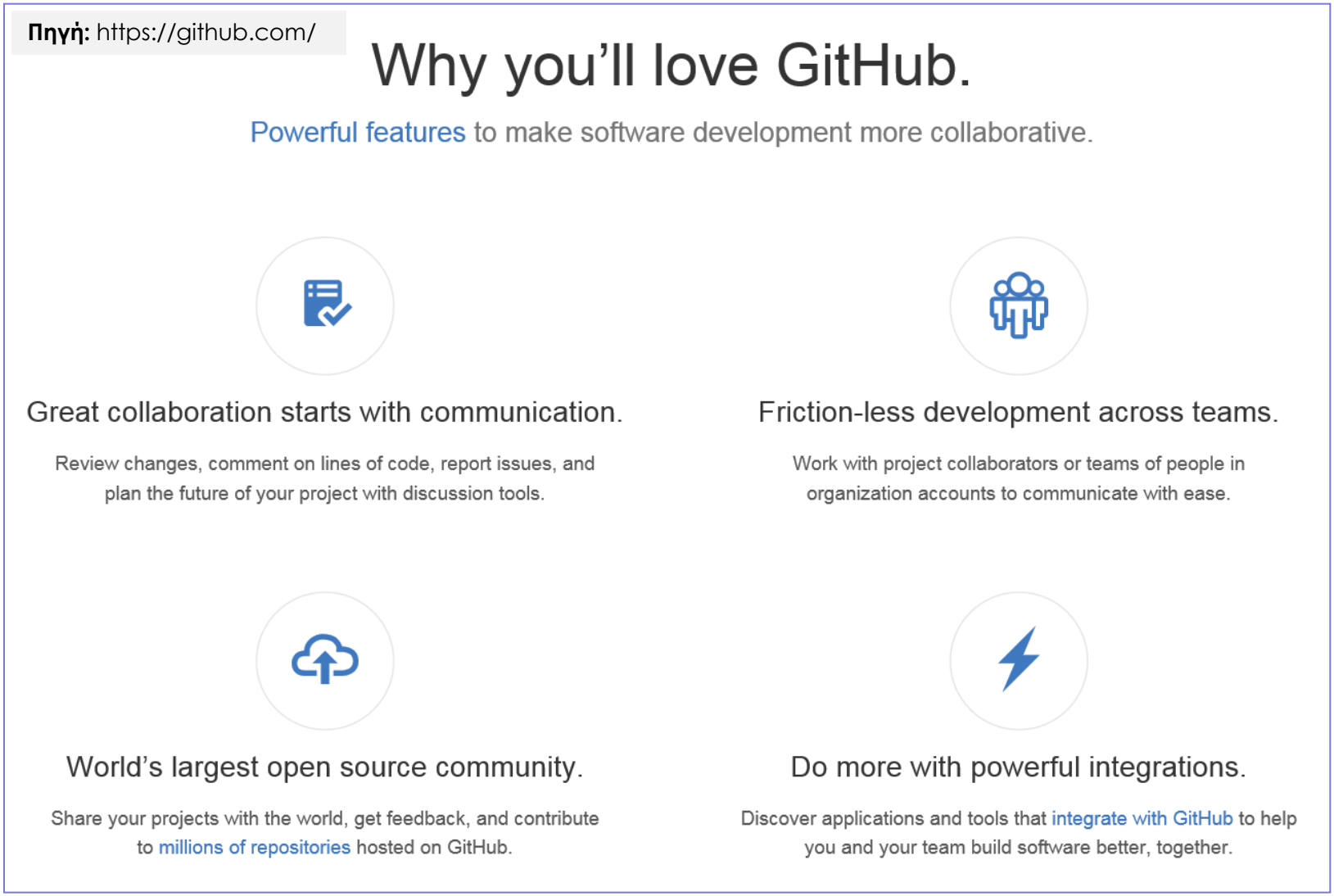
Μοντελοποίηση σε UML



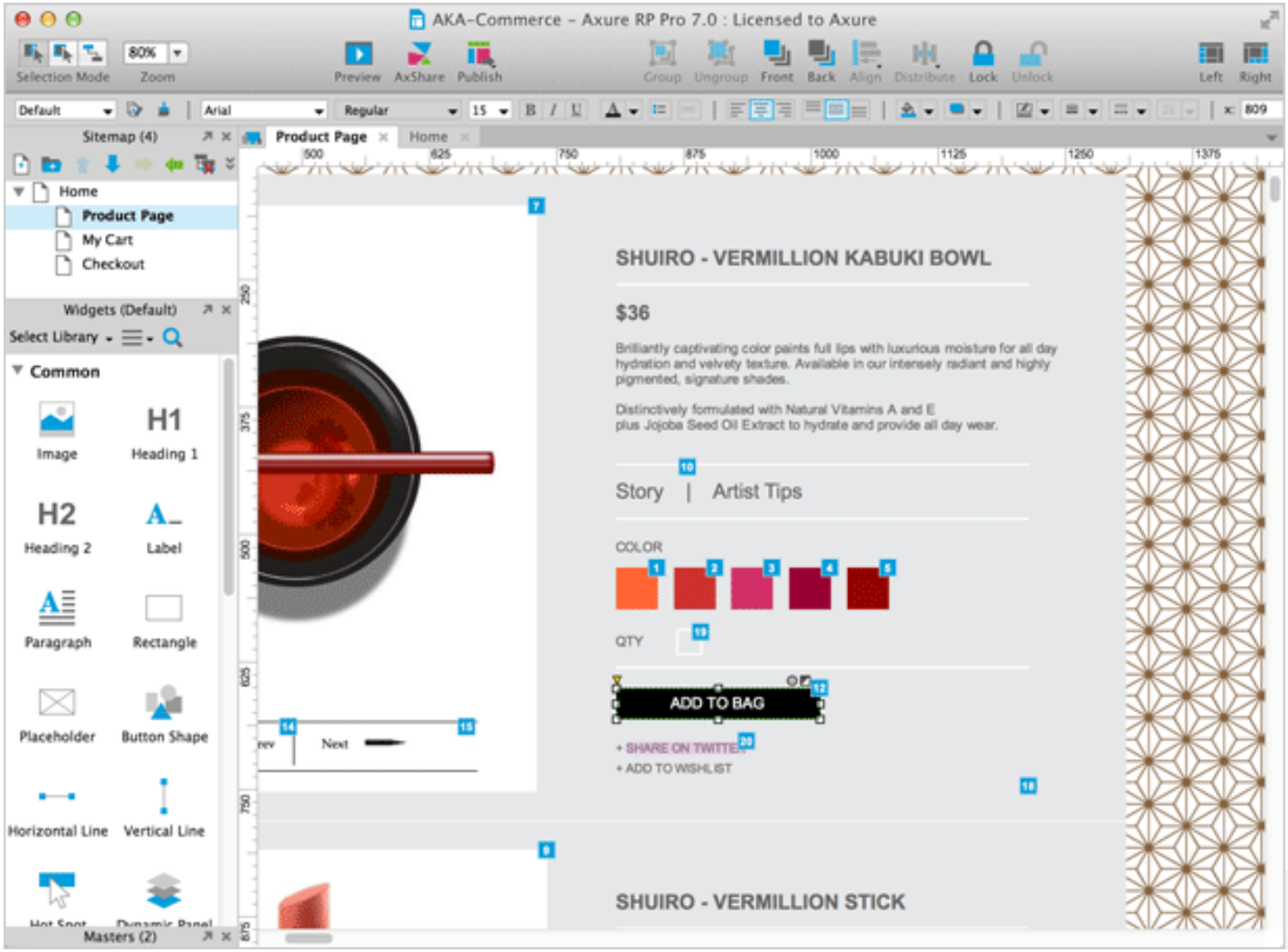
Πηγή: http://www-03.ibm.com/software/products/en/ratirhapfami

Αποθετήρια

Πηγή: https://github.com/



Advanced Prototyping



Πηγή:
http://www.axure.com/

Σημείωμα αδειοδότησης