



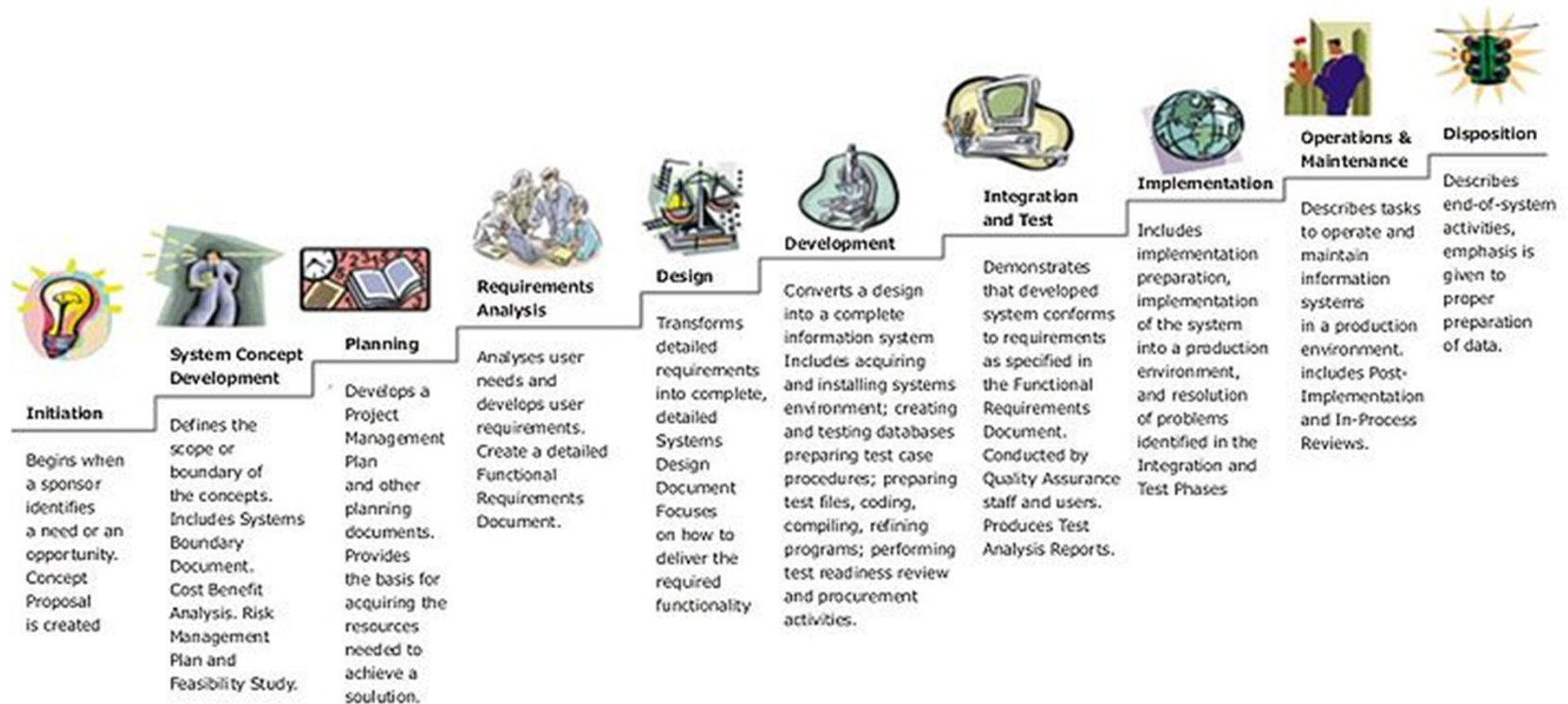
Ανάπτυξη Πληροφοριακών Συστημάτων

Επικ. Καθ. Κωνσταντίνος Μεταξιώτης

kmetax@unipi.gr

Κύκλος Ζωής Ανάπτυξης ΠΣ

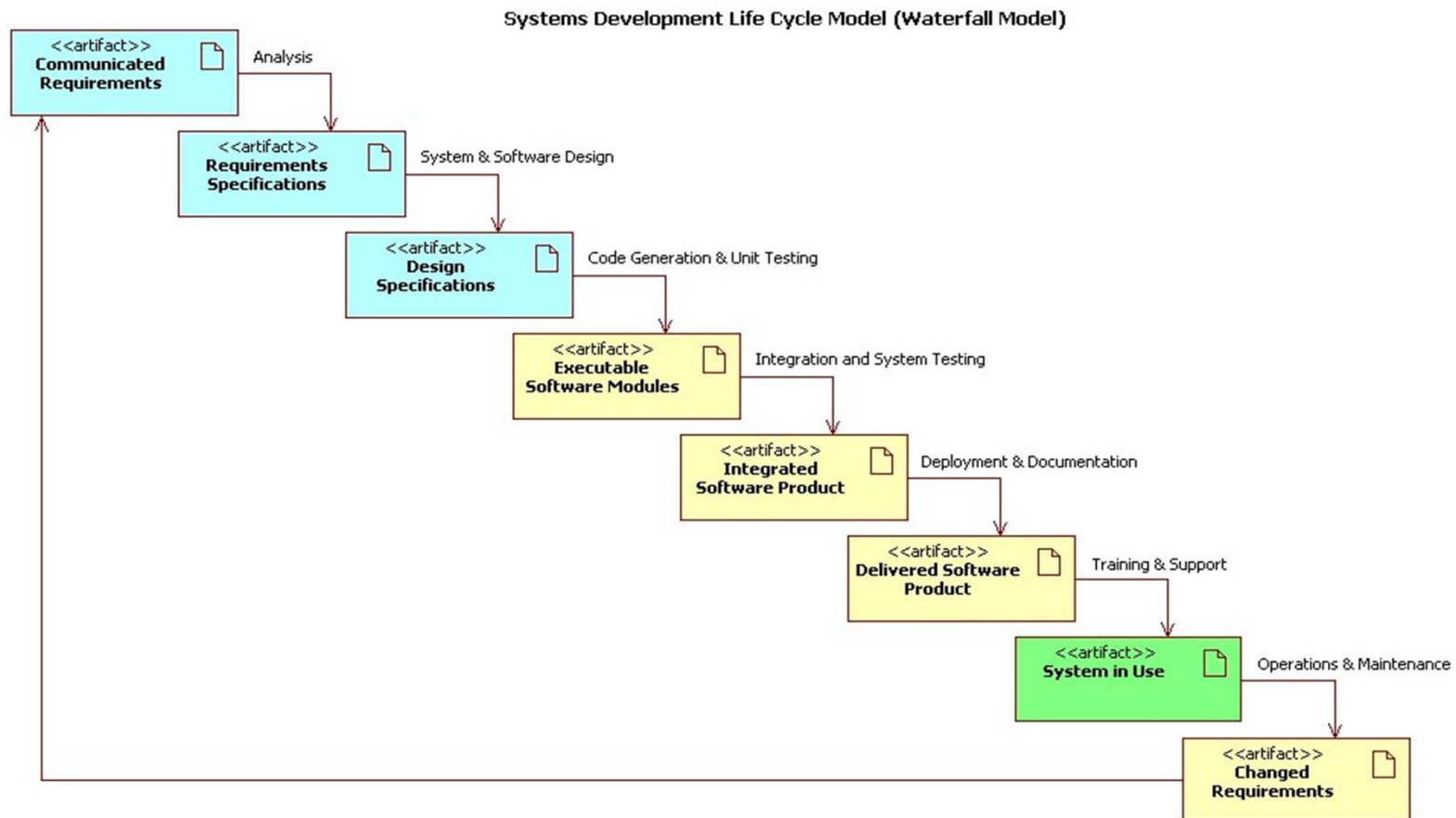
Systems Development Life Cycle (SDLC) Life-Cycle Phases



Αποτελεί τη λογική διαδικασία ανάπτυξης ΠΣ, η οποία χωρίζεται σε τυπικά στάδια, και περιλαμβάνει διάφορα μοντέλα...

A Ανάπτυξη Πληροφοριακών Συστημάτων

Το μοντέλο «Waterfall» (1/2)



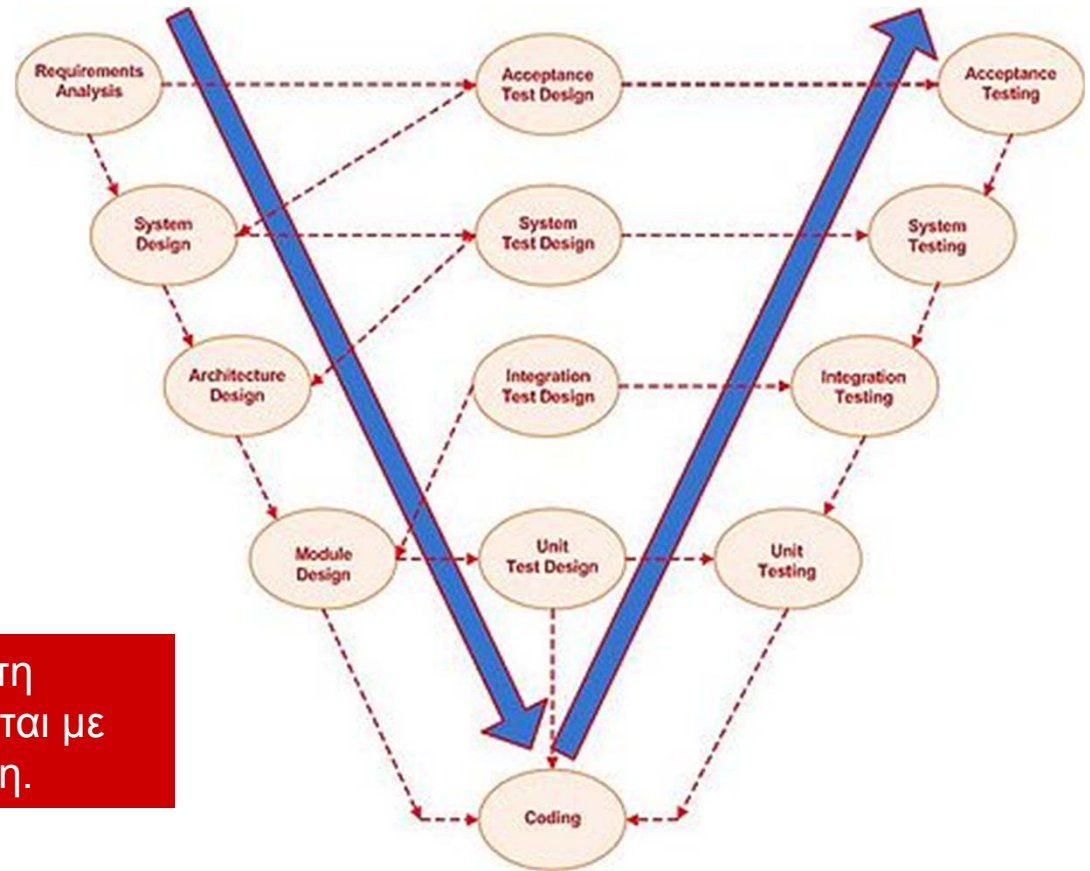
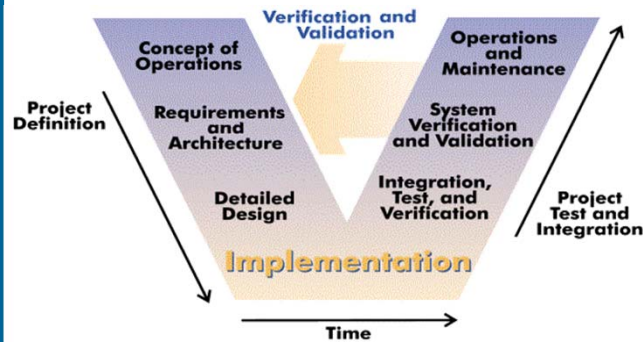
Σύμφωνα με το μοντέλο αυτό η πρόοδος ανάπτυξης ρέει ολοκληρωμένα ως ακολουθία προς τα κάτω σαν “καταρράκτης” (sequential model)...

Το μοντέλο «Waterfall» (2/2)

- **Χαρακτηρίζεται από**
 - Σειριακά βήματα (phases)
 - Βασίζεται στην δημιουργία προδιαγραφών σε κάθε βήμα
- **Προτερήματα**
 - Παραγωγή προδιαγραφών
 - Διευκολύνει την συντήρηση
- **Μειονεκτήματα**
 - Προδιαγραφές που δεν μπορούν να αλλάξουν στη πορεία δεν είναι ρεαλιστική παραδοχή
 - Ο Χρήστης συμμετέχει μόνο στην αρχή
 - Ό χρήστης βλέπει το προϊόν πολύ αργά στη διάρκεια της διαδικασίας

A Ανάπτυξη Πληροφοριακών Συστημάτων

Το μοντέλο «V-Model» (1/2)



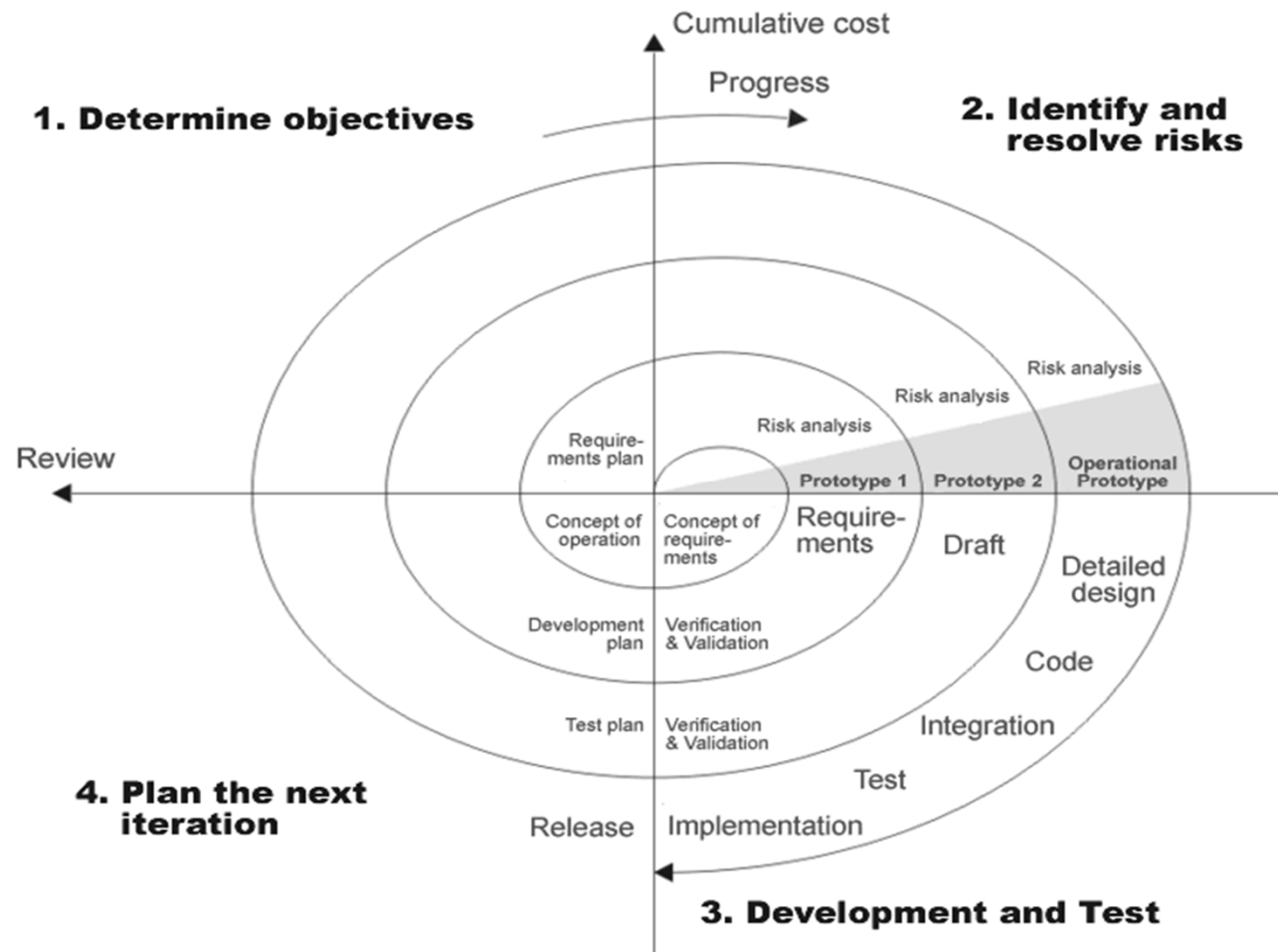
Το μοντέλο αυτό δίνει έμφαση στη σχεδίαση η οποία πάντα συνδέεται με δοκιμές (testing) και πιστοποίηση.

Το μοντέλο «V-Model» (2/2)

- Παραλλαγή γραμμικού μοντέλου
- Αριστερό σκέλος: ανάλυση, σχεδίαση
- Δεξιό σκέλος: δοκιμές, συντήρηση
- Σύνδεση αριστερού με δεξιού σκέλους για την επαλήθευση και επικύρωση.
- Έλεγχοι αποδοχής από πελάτη
- Εισάγει επαναληπτικές και αναθεωρητικές εργασίες
- Εστίαση στις δραστηριότητες και ορθότητα

A Ανάπτυξη Πληροφοριακών Συστημάτων

Το μοντέλο «Spiral» (1/2)



Το μοντέλο αυτό χρησιμοποιείται για την ανάπτυξη μεγάλων ΠΣ και δίνει έμφαση στην ανάπτυξη πρωτοτύπων και εφαρμογή επαναληπτικών δοκιμών (iterative model)...

Το μοντέλο «Spiral» (2/2)

- Συνδυασμός ανάπτυξης και διαχείρισης κινδύνων
 - Νέο βήμα: η αξιολόγηση κινδύνων
- Ουσιαστικά είναι το μοντέλο πρωτοτυποποίησης όπου στο τέλος κάθε βήματος κάνουμε έλεγχο σκοπιμότητας και ανάλυση ρίσκου
- 4 φάσεις επανάληψης
Σε κάθε φάση γίνονται τα εξής:
 - Καθορισμός στόχων → αξιολόγηση κινδύνων → ανάπτυξη και δοκιμές

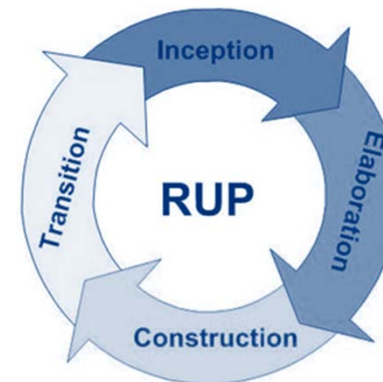
IBM Rational Unified Process (RUP) (1/3)

- Ενοποιημένη διεργασία ανάπτυξης ΠΣ, η οποία υποστηρίζει την **ανάπτυξη λογισμικού βάσει-συστατικών** (component-based software development)
- Ως προϊόν αναπτύχθηκε αρχικά από την Rational Software, και εν συνεχεία αποκτήθηκε από την IBM τον Φεβρουάριο 2003
- Προσαρμόσιμη μεθοδολογία ανάλογα με τον τύπο και το μέγεθος του ΠΣ που αναπτύσσεται
- Επαναληπτική και αυξητική (Iterative and incremental)
- Υποστηρίζεται από **εργαλεία (tools)**, τα οποία αυτοματοποιούν μεγάλα μέρη της διεργασίας

A Ανάπτυξη Πληροφοριακών Συστημάτων

IBM Rational Unified Process (RUP) (2/3)

- Καθοδηγείται από μελέτες χρήσης (**use cases**)
- Χρησιμοποιεί την **Unified Modeling Language (UML)**, η οποία είναι το βιομηχανικό standard που μας επιτρέπει να επικοινωνούμε καθαρά τις απαιτήσεις, τις αρχιτεκτονικές και τα σχέδια
- 2 διαστάσεις: **Δυναμική** (χρόνος) και **Στατική** (ροές εργασίας)



RUP's Four Phases:

1. Inception
2. Elaboration
3. Construction
4. Transition

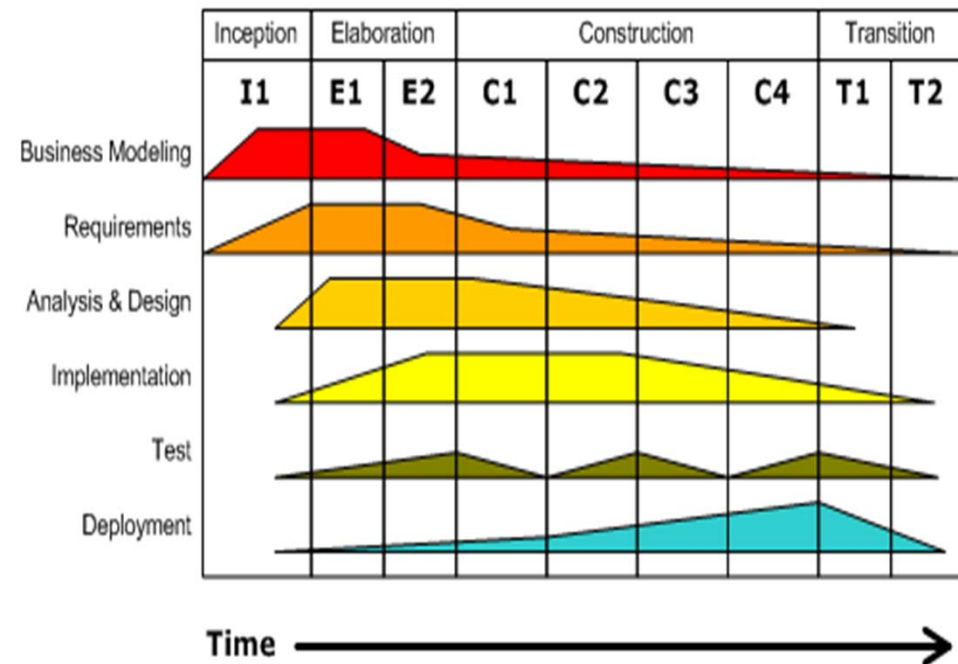
Τέσσερις φάσεις στο χρόνο:

- **Σύλληψη (Inception)** – Ορισμός του έργου και της έκτασής του, εκτίμηση ρίσκου
- **Επεξεργασία (Elaboration)** – Κατάστρωση μεθόδου υλοποίησης, μοντελοποίηση χαρακτηριστικών του ΠΣ, περιγραφή της αρχιτεκτονικής του συστήματος, πρωτότυπο
- **Κατασκευή (Construction)** – Υλοποίηση του ΠΣ, εγκατάσταση ΠΣ, εγχειρίδια χρήσης
- **Μετάβαση (Transition)** – Ανάπτυξη του συστήματος στο περιβάλλον χρήσης του, “beta testing” για την επικύρωση του νέου συστήματος ως προς τις προσδοκίες των χρηστών, εκπαίδευση

- Βασικές ροές εργασίες:
 - Μοντελοποίηση επιχειρησιακού περιβάλλοντος (Business Modeling)
 - Συγγραφή προδιαγραφών (Requirements)
 - Ανάλυση και σχεδίαση (Analysis and Design)
 - Υλοποίηση (Implementation)
 - Έλεγχος (Test)
 - Εγκατάσταση (Deployment)

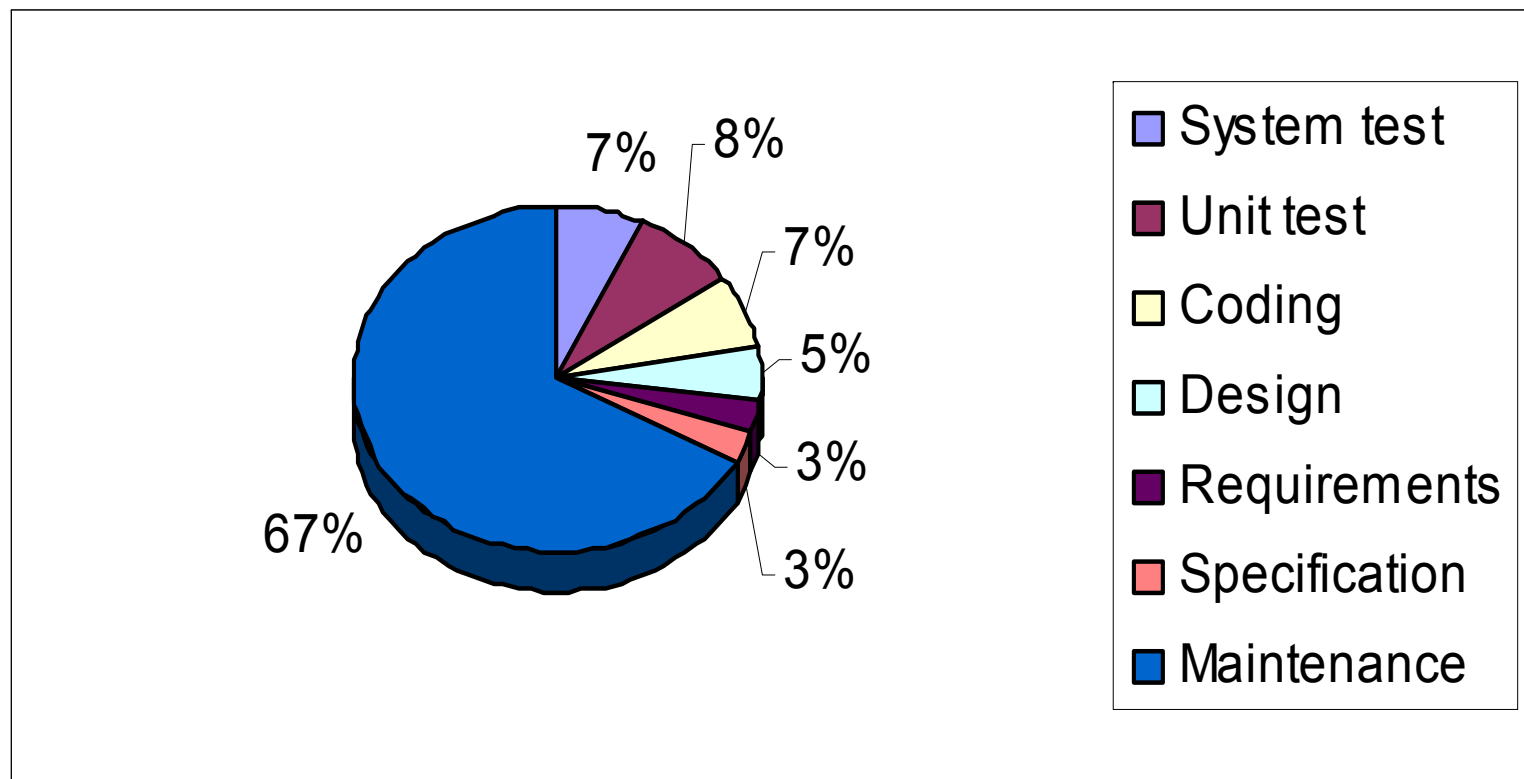
Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



A Ανάπτυξη Πληροφοριακών Συστημάτων

Κατανομή Χρόνου σε ένα τυπικό IS έργο



ΕΡΩΤΗΜΑΤΑ

- Ποιες είναι οι βασικές λειτουργίες του ΠΣ;
- Ποιες οι ειδικές απαιτήσεις;
- Ποια τα κριτήρια επικύρωσης/αποδοχής;

ΠΑΡΑΤΗΡΗΣΕΙΣ

Περιγραφή του τι πρέπει να κάνει το ΠΣ,
ανεξάρτητα από τεχνολογία υλοποίησης

Η ανάλυση απαιτήσεων απαιτεί την κατανόηση του πεδίου, την συλλογή των απαιτήσεων, την κατηγοριοποίηση, δόμηση, την τοποθέτηση προτεραιοτήτων και την αξιολόγηση

- Οι απαιτήσεις διακρίνονται σε δυο κατηγορίες: **Λειτουργικές** και **Μη Λειτουργικές**.
- **Οι Λειτουργικές Απαιτήσεις (Functional Requirements)** : Περιγράφουν τι ακριβώς πρέπει να κάνει το ΠΣ, προσδιορίζοντας τις απαραίτητες λειτουργίες
- **Οι Μη Λειτουργικές Απαιτήσεις (Non Functional Requirements)** : Περιγράφουν περιορισμούς που θα πρέπει να ικανοποιούνται από τις υπηρεσίες και λειτουργίες που προσφέρει το σύστημα.

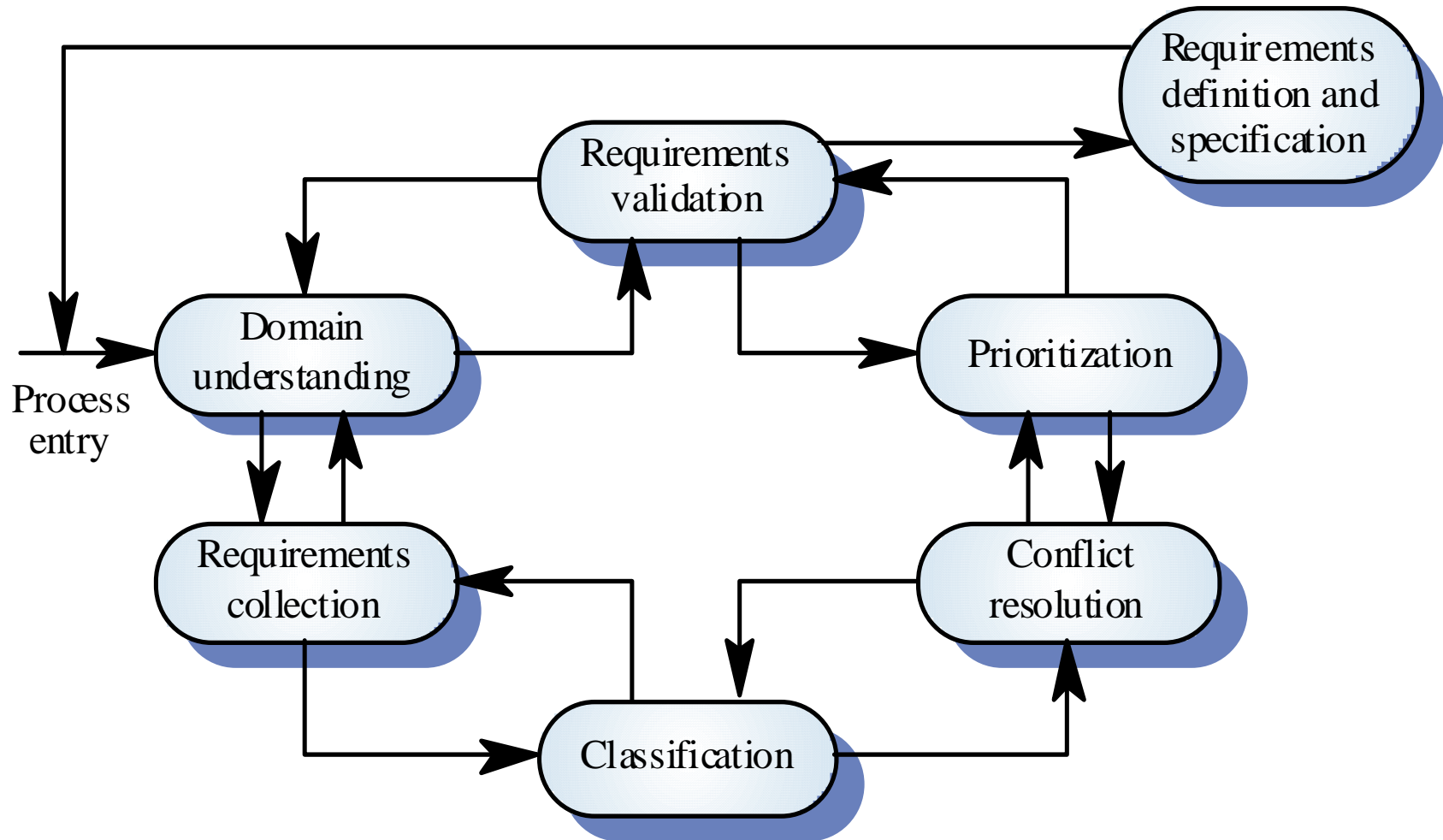
Η καταγραφή των λειτουργικών απαιτήσεων γίνεται κυρίως με μελέτες χρήσης (use cases)

A Ανάπτυξη Πληροφοριακών Συστημάτων

Ανάλυση Απαιτήσεων (3/4)

Η εξέταση της εγκυρότητας των απαιτήσεων είναι μια κρίσιμη δραστηριότητα της μηχανικής των απαιτήσεων. Έλεγχος θα πρέπει να γίνει για τις παρακάτω ιδιότητες :

- **Εγκυρότητα (validity)** : Πιστοποίηση για το ορθό σύστημα.
- **Συνέπεια (Consistency)** : Νέες απαιτήσεις δεν θα πρέπει να έρχονται σε αντίθεση με άλλες.
- **Πληρότητα (Completeness)** : Πρέπει να περιλαμβάνονται όλες οι λειτουργίες και οι προδιαγραφές.
- **Ρεαλισμός (Realism)** : Καθορισμός μόνο υλοποιήσιμων απαιτήσεων.
- **Ελεγκσιμότητα (Verifiability)** : Πλήρης ελεγκσιμότητα των απαιτήσεων.
- **Αντιληπτότητα (Comprehensibility)** : Απόλυτα κατανοήσιμες απαιτήσεις.
- **Δυνατότητα Ιχνηλάτησης (Traceability)** : Η προέλευση των προδιαγραφών θα πρέπει να είναι απόλυτα γνωστή.
- **Προσαρμοστικότητα (Adaptability)** : Οι απαιτήσεις θα πρέπει να τροποποιούνται χωρίς απρόβλεπτες συνέπειες και αλλαγές.



ΕΡΩΤΗΜΑΤΑ

- Πώς θα είναι η δομή του ΠΣ;
- Ποιος ο εξοπλισμός (hardware/software);
- Ποιες διαδικασίες απαιτούνται;
- Πώς θα γίνουν οι δοκιμές ελέγχου;

ΠΑΡΑΤΗΡΗΣΕΙΣ

- Αναλυτική περιγραφή του πώς θα είναι το σύστημα
- Τεχνικές προδιαγραφές υλικού/λογισμικού

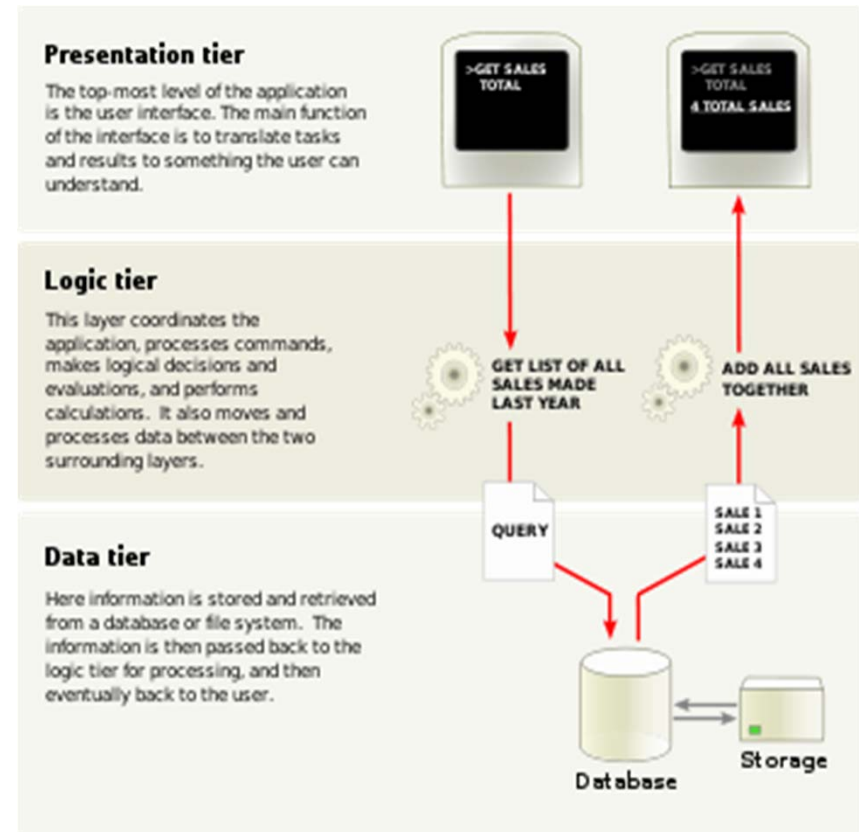
Γνωστές μέθοδοι που χρησιμοποιούνται για σχεδίαση ΠΣ είναι: Business Process Modeling Notation (BPMN), Unified Modeling Language (UML), Extended Enterprise Modeling Language (EEML)

Η σχεδίαση θα πρέπει να γίνεται με τέτοιο τρόπο, ώστε να εξασφαλίζονται οι παρακάτω ιδιότητες του συστήματος :

- **Compatibility** : Συμβατότητα του συστήματος με άλλα προϊόντα.
- **Extensibility** : Επεκτασιμότητα του συστήματος
- **Modularity** : το σύστημα πρέπει να αποτελείται από μονάδες (modules)
- **Reliability** : Αξιοπιστία του συστήματος
- **Reusability** : Ικανότητα επαναχρησιμοποίησης των σχετικών μονάδων σε άλλες, παρόμοιες εφαρμογές
- **Robustness** : Ευρωστία του ΠΣ, να μπορεί να λειτουργεί και κάτω από απρόβλεπτες συνθήκες (π.χ. Invalid inputs, low memory conditions)
- **Security** : Ασφάλεια του συστήματος
- **Usability** : Ο βαθμός ευχρηστίας και φιλικότητας του συστήματος.

ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ ΣΤΥΛ

- Client-Server (2-tier, n-tier)
- Service-oriented Architecture (SOA)
- 3-tier model (Presentation, Business Logic, Database Tiers)



ΕΡΩΤΗΜΑΤΑ

- Πώς κατασκευάζουμε το «κατάλληλο» λογισμικό με το χαμηλότερο κόστος;
- Πώς κατασκευάζουμε ή προμηθευόμαστε το «κατάλληλο» υλικό;
- Πώς θα είναι οι διαδικασίες που θα πρέπει να ακολουθούνται;

ΠΑΡΑΤΗΡΗΣΕΙΣ

- Υλοποίηση κώδικα
- Τεκμηρίωση υλικού/λογισμικού/διαδικασιών

Δοκιμή (testing) (1/2)

- Είναι η διαδικασία μέσω της οποίας γίνονται δοκιμές είτε από τον υλοποιητή του ΠΣ είτε από τον τελικό χρήστη ή πελάτη
- Ειδικότερα, η διαδικασία που ακολουθείται από τον τελικό χρήστη ή πελάτη λέγεται **User Acceptance Testing**
- Ενδεικτικά αναφέρουμε ότι οι **Alpha testing** και **Beta testing** διαδικασίες εφαρμόζονται πολύ συχνά στην πράξη



Δοκιμή (testing) (2/2)

A Typical Cycle for Testing

Test planning: [Test strategy](#), [test plan](#), [testbed](#) creation. Since many activities will be carried out during testing, a plan is needed.

•**Test development:** Test procedures, [test scenarios](#), [test cases](#), test datasets, test scripts to use in testing software.

•**Test execution:** Testers execute the software based on the plans and tests and report any errors found to the development team.

•**Test reporting:** Once testing is completed, testers generate metrics and make final reports on their [test effort](#) and whether or not the software tested is ready for release.

•**Test result analysis:** Or Defect Analysis, is done by the development team usually along with the client, in order to decide what defects should be treated, fixed, rejected (i.e. found software working properly) or deferred to be dealt with later.

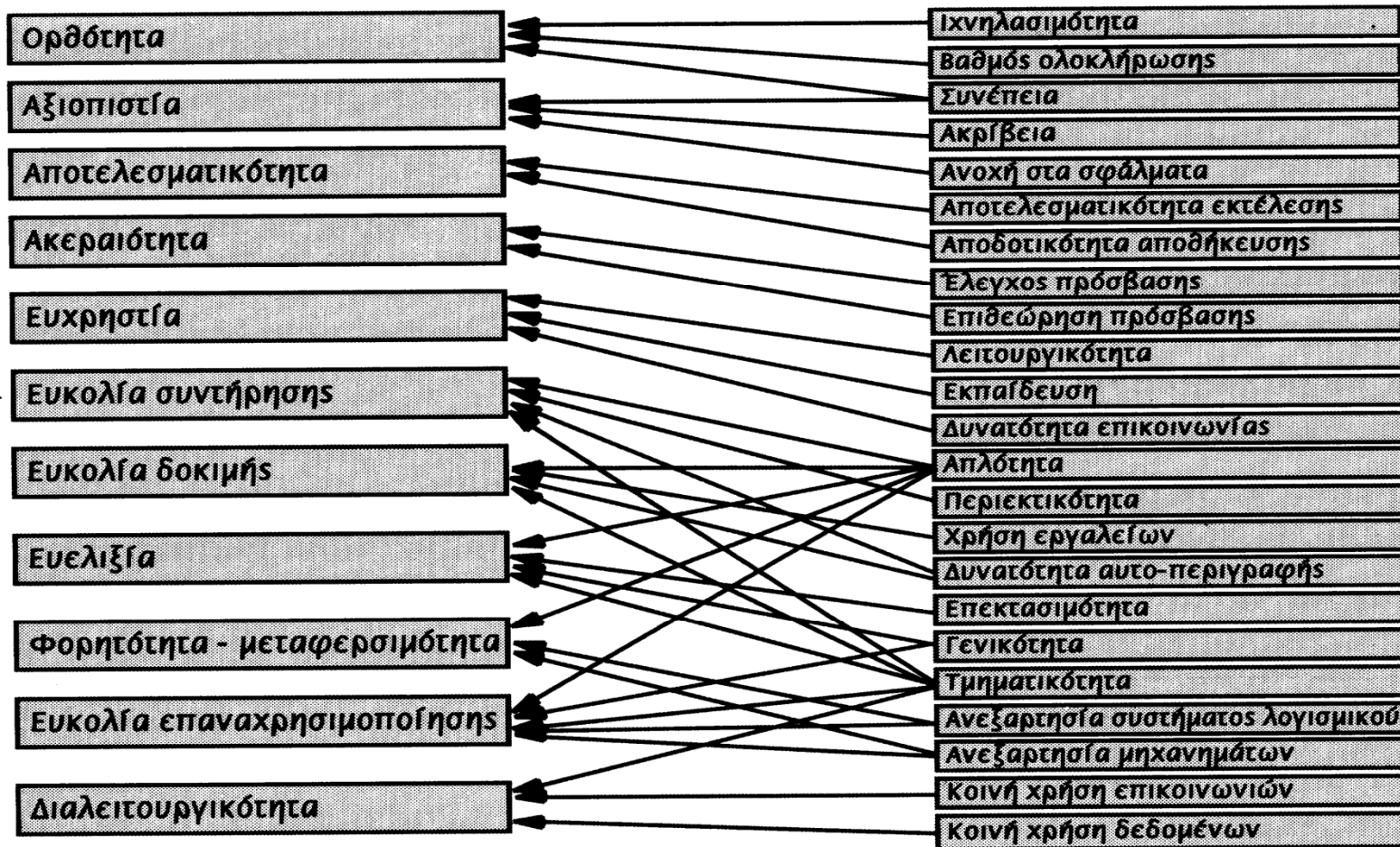
•**Defect Retesting:** Once a defect has been dealt with by the development team, it is retested by the testing team.

•**Regression testing:** It is common to have a small test program built of a subset of tests, for each integration of new, modified, or fixed software, in order to ensure that the latest delivery has not ruined anything, and that the software product as a whole is still working correctly.

•**Test Closure:** Once the test meets the exit criteria, the activities such as capturing the key outputs, lessons learned, results, logs, documents related to the project are archived and used as a reference for future projects.

Α Ανάπτυξη Πληροφοριακών Συστημάτων

Αξιολόγηση Ποιότητας ΠΣ



Εικόνα 1.5. Το μοντέλο αξιολόγησης ποιότητας του McCall.

ΕΡΩΤΗΜΑΤΑ

- Δουλεύει το σύστημα ικανοποιητικά;
- Πώς θα γίνει η μετάβαση από το παλαιό στο νέο;

ΠΑΡΑΤΗΡΗΣΕΙΣ

- Εγχειρίδια με οδηγίες
- Παραλαβή του συστήματος

ΕΡΩΤΗΜΑΤΑ

- Απαιτούνται προσθήκες, αλλαγές, μελλοντικές τροποποιήσεις, βελτιώσεις

ΠΑΡΑΤΗΡΗΣΕΙΣ

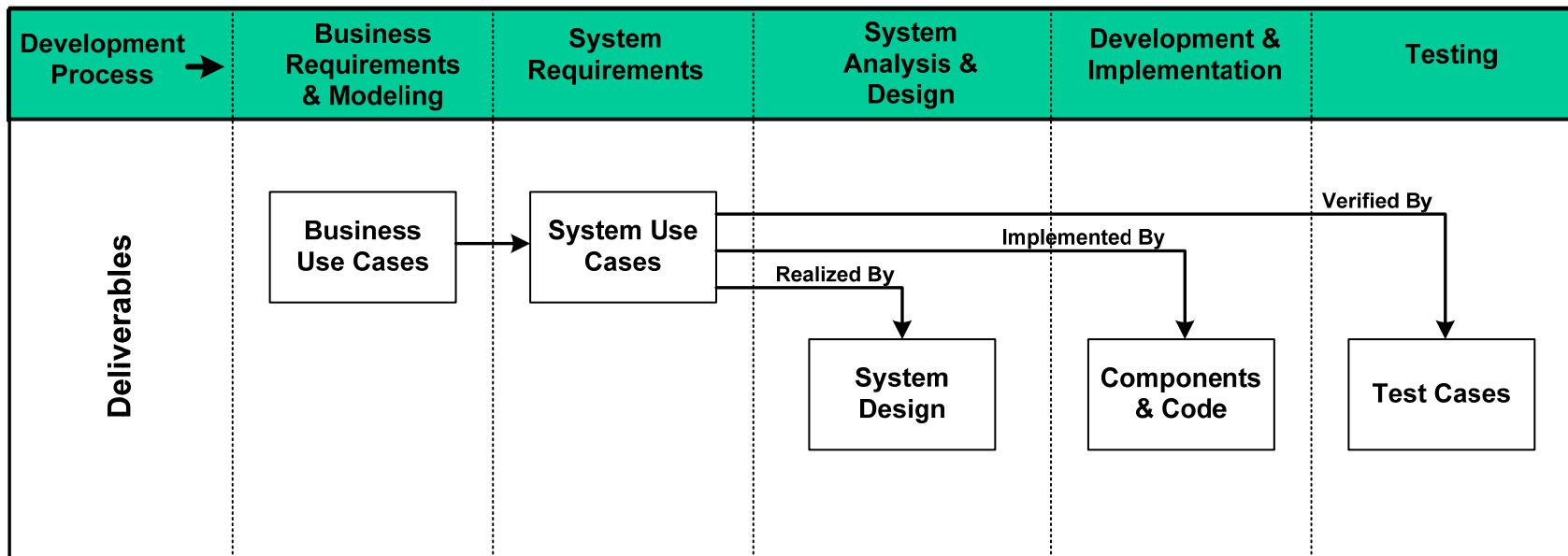
- Πλάνο Συντήρησης
- Καταγραφή μελλοντικών αναγκών

ΠΑΡΑΔΕΙΓΜΑΤΑ

- Service Level Agreements (SLAs): Εξειδικευμένα συμβόλαια συντήρησης
- Help Desk

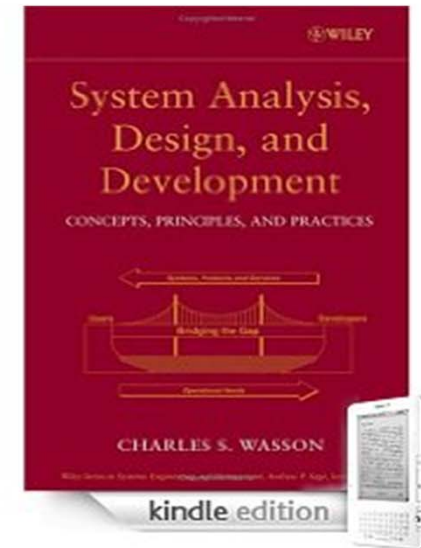
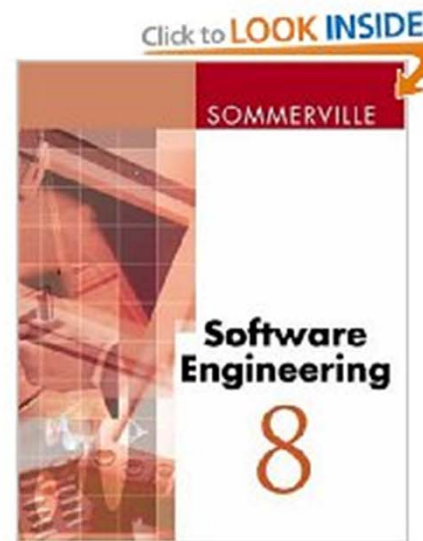
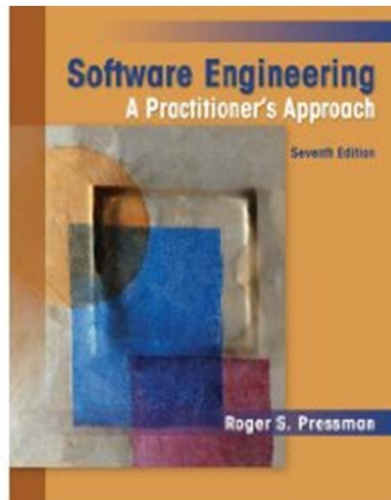
A Ανάπτυξη Πληροφοριακών Συστημάτων

Σύνοψη



Α Ανάπτυξη Πληροφοριακών Συστημάτων

Προτεινόμενη Βιβλιογραφία



Α Ανάπτυξη Πληροφοριακών Συστημάτων

Προτεινόμενη Βιβλιογραφία

