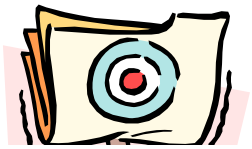
HY351:
Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων
Information Systems Analysis and Design

Καθορισμός των Απαιτήσεων (Requirements Determination)

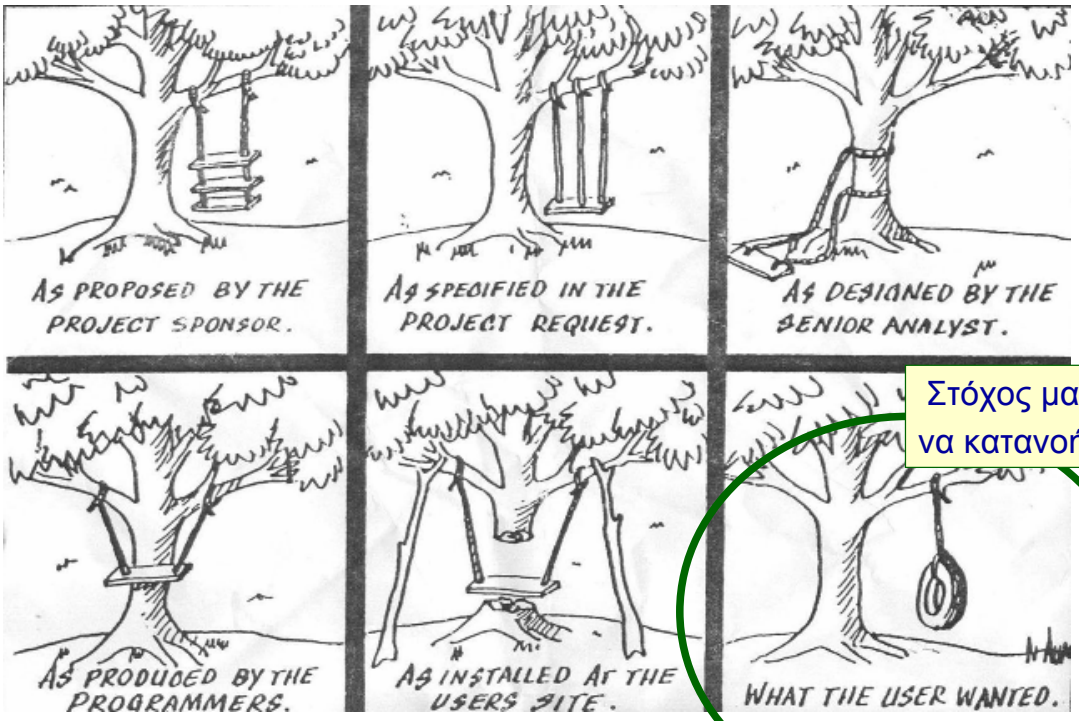


Γιάννης Τζίτζικας

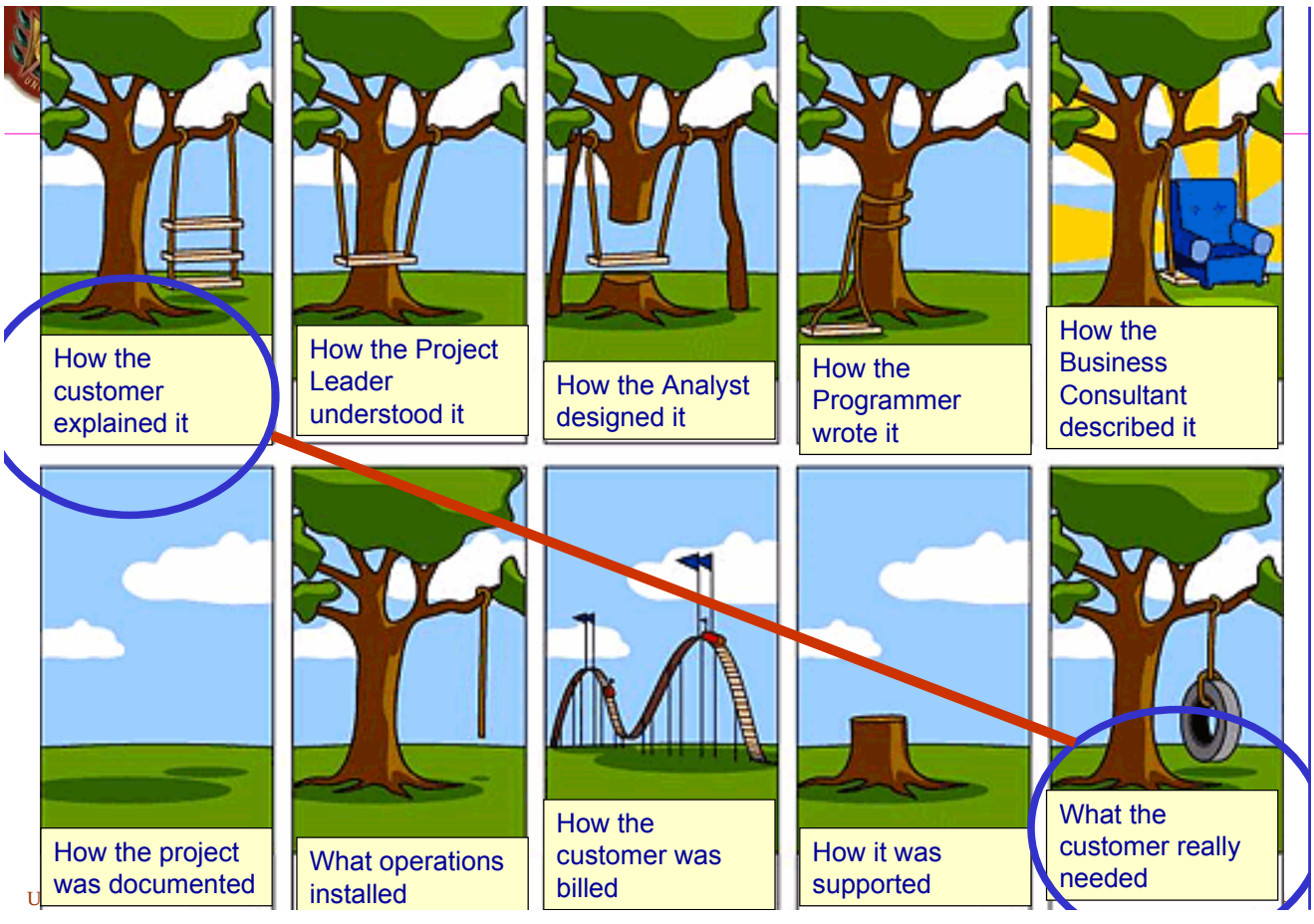
Διάλεξη : 6α
Ημερομηνία :
Θέμα :

Διάρθρωση

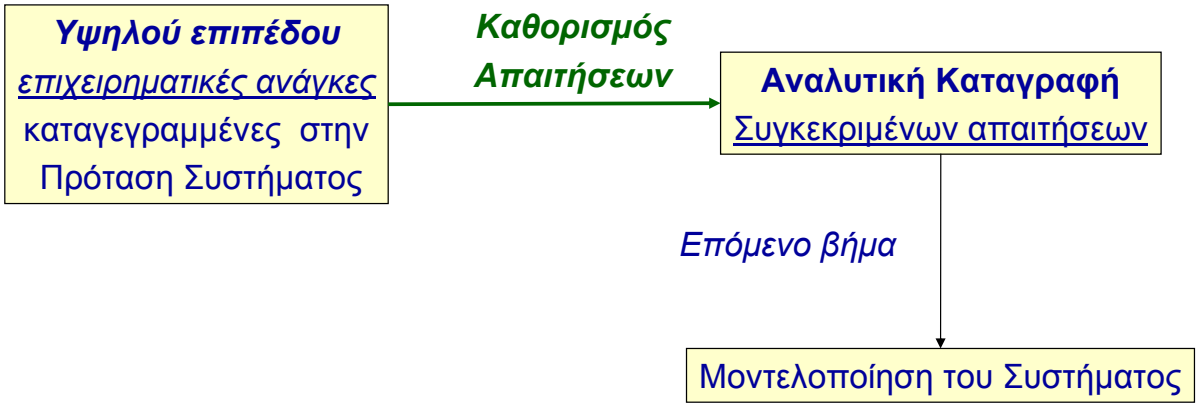
- Τι είναι ο Καθορισμός Απαιτήσεων;
- Τι είναι «Απαίτηση»;
- Λειτουργικές και Μη-Λειτουργικές Απαιτήσεις
  - Functional and Nonfunctional Requirements (FR and NFR)
- Ποιος, πως και πότε καθορίζει τις απαιτήσεις?
- Το Έγγραφο Περιγραφής Απαιτήσεων
  - The Requirements Specification Document
- Τις μας συστήνει να κάνουμε η Αντικειμενοστρεφής Ανάλυση και Σχεδίαση;
- Διάφορες Διαγραμματικές Τεχνικές



Στόχος μας είναι να κατανοήσουμε



Καθορισμός Απαιτήσεων



Σχόλια

- Δεν υπάρχει ξεκάθαρη διαχωριστική γραμμή μεταξύ ανάλυσης και σχεδίασης
- Ανάλυση ~ το πρώτο βήμα της Σχεδίασης

Τι είναι η «Απαίτηση»;

- Απαίτηση είναι μια δήλωση του τι το σύστημα πρέπει να κάνει
  - Requirement is a statement of what a system must do.
- Οι απαιτήσεις αρχικά περιγράφονται από τη σκοπιά της επιχείρησης (και όχι την τεχνική)
- Κατόπιν προσθέτονται και τεχνικές απαιτήσεις (που αλλιώς λέγονται «απαιτήσεις συστήματος»)
  - “system requirements”
- Οι απαιτήσεις συχνά αποτελούν μια μορφή **συμβολαίου** μεταξύ πελατών και κατασκευαστών
  - Εξαίρεση: εύκαμπτες μεθοδολογίες ανάπτυξης.

Λειτουργικές και Μη-Λειτουργικές Απαιτήσεις (ΛΑ & ΜΛΑ) (Functional and Non-Functional Requirements)

- **Λειτουργικές Απαιτήσεις (ΛΑ) Functional Requirements (FR)**
  - Περιγράφουν **τι πρέπει να κάνει** το σύστημα (π.χ. ως συναρτήσεις που λαμβάνουν είσοδο και δίδουν έξοδο)
- **Μη-Λειτουργικές Απαιτήσεις (ΜΛΑ)**
  - Περιγράφουν **ιδιότητες** του συστήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής:
    - Απόδοση (performance)
    - Χρηστικότητα (usability)
    - Ασφάλεια (security)
    - Νομιμότητα (legislative)
    - Ιδιωτικότητα (privacy)
  - Με άλλα λόγια: περιγράφουν το **πώς** (ή το **πόσο καλά**) το σύστημα θα υποστηρίξει τις λειτουργικές απαιτήσεις
  - Μπορούμε να τις θεωρήσουμε ως «**περιορισμούς»** που περιορίζουν τους τρόπους με τους οποίους θα μπορούσαμε να πραγματώσουμε τις λειτουργικές απαιτήσεις.

Χωρίστε τις παρακάτω Απαιτήσεις σε ΛΑ και ΜΛΑ

- *Ο χρόνος απόκρισης του συστήματος δεν πρέπει να υπερβαίνει τα 3 δευτερόλεπτα*
- *Το σύστημα πρέπει να μπορεί να ενοποιηθεί με το υπάρχον*
- *Τα προσωπικά στοιχεία των πελατών πρέπει να προστατεύονται.*
- *Να αποθηκεύει τα στοιχεία των πελατών*
- *Να τυπώνει συγκεντρωτικές αναφορές*
- *Το σύστημα πρέπει να λειτουργεί αδιάλειπτα (όλο το χρόνο)*
- *Να τυπώνει αποδείξεις*
- *Μόνο οι διευθυντές πρέπει να έχουν πρόσβαση τους μισθούς*
- *Το σύστημα πρέπει συμμορφώνεται με τα πρότυπα της βιομηχανίας*
- *Το σύστημα πρέπει να μπορεί να υποστηρίξει πολλές φυσικές γλώσσες*

ΛΑ

- *Να αποθηκεύει τα στοιχεία των πελατών*
- *Να τυπώνει συγκεντρωτικές αναφορές*
- *Να τυπώνει αποδείξεις*

ΜΛΑ

- *Ο χρόνο απόκρισης του συστήματος δεν πρέπει να υπερβαίνει τα 3 δευτερόλεπτα*
- *Το σύστημα πρέπει να λειτουργεί αδιάλειπτα (όλο το χρόνο)*
- *Μόνο οι διευθυντές πρέπει να έχουν πρόσβαση τους μισθούς*
- *Το σύστημα πρέπει συμμορφώνεται με τα πρότυπα της βιομηχανίας*
- *Το σύστημα πρέπει να μπορεί να ενοποιηθεί με το υπάρχον*
- *Το σύστημα πρέπει να μπορεί να υποστηρίξει πολλές φυσικές γλώσσες*
- *Τα προσωπικά στοιχεία των πελατών πρέπει να προστατεύονται.*

Οι Μη Λειτουργικές απαιτήσεις ~ Χαρακτηριστικά Λογισμικού

- Ορθότητα (Correctness)
  - Ένα πρόγραμμα είναι *λειτουργικά ορθό* όταν συμπεριφέρεται σύμφωνα με τις καταγεγραμμένες λειτουργικές απαιτήσεις.
- Αξιοπιστία (Reliability)
  - Το λογισμικό θα πρέπει να μην προκαλεί φυσική ή οικονομική καταστροφή στην περίπτωση λάθους. (Η πιθανότητα το λογισμικό να συμπεριφέρεται σωστά σε ένα συγκεκριμένο χρονικό διάστημα)
- Αποδοτικότητα (Performance)
  - Το πρόγραμμα δεν θα πρέπει να κάνει αλόγιστη χρήση των πόρων του συστήματος
- Ευχρηστία (Usability)
  - Το λογισμικό πρέπει να επικοινωνεί καλά με το χρήση.

Οι Μη Λειτουργικές απαιτήσεις ~ Χαρακτηριστικά Λογισμικού

- Ευελιξία – Δυνατότητα Συντήρησης (Maintainability)
  - Εύκολη εξέλιξη του συστήματος σε περίπτωση αλλαγής των απαιτήσεων
- Επαληθευσιμότητα (Verifiability)
  - Εύκολη επαλήθευση της ορθής λειτουργίας του συστήματος (π.χ. η λειτουργική ορθότητα, ή η απόδοση πρέπει να μπορούν να ελεγχθούν με χρήση προσομοίωσης, ή μέσω τυπικών μεθόδων)
- Δυνατότητα Επαναχρησιμοποίησης (Reusability)
  - Δυνατότητα χρήσης του για την ανάπτυξη άλλων εφαρμογών.
- Φορητότητα (Portability)
  - Δυνατότητα εκτέλεσης του προγράμματος σε διαφορετικά περιβάλλοντα (λειτουργικά συστήματα, βάσεις δεδομένων).

Μια άλλη κατηγοριοποίηση των ΜΛΑ (Another categorization of NFR)

- Επιχειρησιακές (Operational)
  - Σχετίζονται με το φυσικό και τεχνικό περιβάλλον στο οποίο θα λειτουργήσει το σύστημα
- Απόδοσης (Performance)
  - Σχετίζονται με την Ταχύτητα, χωρητικότητα/αξιότητα (capacity), αξιοπιστία (reliability)
- Ασφάλειας
- Πολιτιστικές και Πολιτικες (Cultural and Political)
  - Παράγοντες κουλτούρας, πολιτικής και νομοθεσίας που επηρεάζουν το σύστημα

Η Σπουδαιότητα και η Δυσκολία των ΜΛΑ

Έχουν αντίκτυπο στις σχεδιαστικές αποφάσεις που αφορούν κυρίως την σχεδίαση του φυσικού επιπέδου

Παραδείγματα

- Η επιλογή του Συστήματος Διαχείρισης Βάσης Δεδομένων
- Αρχιτεκτονική για Ασφάλεια

- Δύσκολα μοντελοποιούνται
- Συχνά διατυπώνονται άτυπα και ασαφώς
- Δύσκολα μπορούμε να ελέγξουμε την ικανοποίηση τους πριν την παράδοση του συστήματος στον πελάτη

Επιθυμητές Ιδιότητες Περιγραφής Απαιτήσεων (αμφότερων ΛΑ και ΜΛΑ)

- Ορθότητα
  - Πρέπει να επικυρώνονται από τον πελάτη και την ομάδα έργου
- Συνέπεια
  - Δεν πρέπει να υπάρχουν αντιφάσεις (χρήστες <10, χρήστες <100)
- Πληρότητα
- Δυνατότητα Πραγμάτωσης (Επιτευξιμότητα)
- Δυνατότητα Ελέγχου Επίτευξης (Επαληθευσιμότητα)
  - Πρέπει να μπορούμε να ελέγξουμε την επίτευξη μιας απαίτησης
- Δυνατότητα Εξιχνίασης (Ιχνηλασιμότητα) (traceability)
  - Πρέπει να μπορούμε να εντοπίζονται εύκολα τις επιχειρηματικές ανάγκες που οδήγησαν στον προσδιορισμό της κάθε απαίτησης

Οι ΜΛΑ πρέπει να είναι μετρήσιμες!
(NFRs should be measurable!)

Πρέπει να μπορούμε να μετρήσουμε το βαθμό ικανοποίησης κάθε ΜΛΑ

Παραδείγματα διατύπωσης μετρήσιμων μη-λειτουργικών απαιτήσεων (1/6)

- Το σύστημα πρέπει να παρέχει απόκριση *πραγματικού χρόνου*
- Το σύστημα πρέπει να κάνει *καλή διαχείριση του αποθηκευτικού χώρου*
- Το σύστημα πρέπει είναι μπορεί να διεκπεραιώσει *πολλές δοσοληψίες* ταυτόχρονα
- Το σύστημα πρέπει να μπορεί να εξυπηρετεί αποδοτικά *πολλούς χρήστες ταυτόχρονα*

- Το σύστημα πρέπει να αποκρίνεται σε 2 δευτερόλεπτα το πολύ
  - hardware = …, συνθήκες χρήσης=..
- Ο χώρος στο δίσκο για έναν πελάτη δεν πρέπει να υπερβαίνει τα 200 bytes
- Το σύστημα πρέπει είναι μπορεί να διεκπεραιώσει τουλάχιστον 100 δοσοληψίες ταυτόχρονα
- Ο χρόνος απόκρισης δεν πρέπει να υπερβαίνει τα 2 δεύτερα ακόμα και αν έχουμε 20 ταυτόχρονους χρήστες

Παραδείγματα διατύπωσης μετρήσιμων μη-λειτουργικών απαιτήσεων (2/6)

- Η εκμάθηση του τρόπου χειρισμού του συστήματος από τους εργαζομένους της επιχείρησης πρέπει να είναι εφικτή και γρήγορη.
- Το σύστημα πρέπει να είναι φιλικό στη χρήση

- Η εκμάθηση του συστήματος δεν πρέπει να απαιτήσει πάνω από 4 ώρες εκπαίδευση
- Κάθε οθόνη πρέπει να έχει παράθυρο βοήθειας
- κατά την παραγγελιοληψία ο χρήστης αντί να πληκτρολογεί πρέπει να μπορεί να επιλέξει τον τύπο του προϊόντος, καθώς και την χώρα/πόλη αποστολής από προκαθορισμένες λίστες.
- Το σύστημα δεν πρέπει να επιτρέπει την εισαγωγή στοιχείων που δεν ικανοποιούν τον τύπο των αντίστοιχων πεδίων

Παραδείγματα διατύπωσης μετρήσιμων μη-λειτουργικών απαιτήσεων (3/6)

- Το σύστημα πρέπει να είναι *εύρωστο* (robust)

- Το ποσοστό των συμβάντων που έχουν σαν αποτέλεσμα την πτώση (διακοπή ομαλής λειτουργίας) του συστήματος δεν πρέπει να υπερβαίνει το 2%
- Ο χρόνος επανεκκίνησης του συστήματος μετά από οποιαδήποτε διακοπή δεν πρέπει να υπερβαίνει τα 3 λεπτά
- Το σύστημα πρέπει να κάνει αυτόματη επανεκκίνηση μετά από πτώση

Παραδείγματα διατύπωσης μετρήσιμων μη-λειτουργικών απαιτήσεων (4/6)

- Το σύστημα πρέπει να είναι *αξιόπιστο*
- Το σύστημα πρέπει να είναι *μεταφέρσιμο* (portable) σε άλλες πλατφόρμες

- Το σύστημα δεν πρέπει να καταρρέει πάνω από 2 φορές το χρόνο
- Ο μέσος χρόνος μεταξύ 2 καταρρεύσεων πρέπει να είναι τουλάχιστον 4 μήνες
- Το σύστημα πρέπει να κρατά αντίγραφα ασφαλείας
- Ο ποσοστό των γραμμών κώδικα που εξαρτώνται από την πλατφόρμα υλοποίησης δεν πρέπει να υπερβαίνει το 4%

Παραδείγματα διατύπωσης μετρήσιμων μη-λειτουργικών απαιτήσεων (5/6)

- Το σύστημα πρέπει να είναι ασφαλές

- Κάθε χρήστης πρέπει να έχει όνομα εισόδου και κωδικό πρόσβασης
- Η επικοινωνία πρέπει να είναι κρυπτογραφημένη (RSA)
- Αν το (ATM) σύστημα δεν μπορεί να επικοινωνήσει με τον υπολογιστή της τράπεζας, τότε πρέπει να διακόπτεται αμέσως η λειτουργία του

Παραδείγματα διατύπωσης μετρήσιμων μη-λειτουργικών απαιτήσεων σχετικές με Χρηστικότητα (Usability) (6/6)

- The product shall be easy for 11 year-old children to use.
- The product shall help the user to avoid making mistakes.
- The product shall make the users want to use it.
- The product shall be used by people with no training, and possibly no understanding of English.

- [An agreed percentage, say 90%] of a test panel of 11 year olds shall be able to successfully complete [list of tasks] within [specified time]
- One month's use of the product shall result in a total error rate of less than [an agreed percentage, say 2%]
- An anonymous survey shall show that [an agreed percentage, say 75%] of the users are regularly using the product after [an agreed time] familiarization period.

Taken fro Volere Specification Template

(σχετικά με ευρωστία και αξιοπιστία) Τύποι Σφαλμάτων

- Μόνιμα (permanent)
  - Επέρχονται (εμφανίζονται) σε κάθε είσοδο (with all inputs)
- Transient
  - Συμβαίνουν με συγκεκριμένες εισόδους
- Μη-Ανανήψιμα (Unrecoverable)
  - Η ανθρώπινη παρέμβαση είναι απαραίτητη για την ανάνηψη (recovery) του συστήματος
- Ανανήψιμα (Recoverable)
  - Το σύστημα μπορεί να ανανήψει από μόνο του
- Φθοροποιά (Corrupting)
  - Τα δεδομένα μπορούν να φθαρούν
- Μη-Φθοροποιά
  - Η ακεραιότητα (integrity) των δεδομένων διατηρείται

Άλλες συμβουλές για τη γλωσσική διατύπωσή των ΜΛΑ

- Αποφυγή λέξεων και φράσεων όπως:
  - **αρκετά, πολλά, γρήγορα, επαρκές, εφικτό, όσο γίνεται, αποτελεσματικό, φιλικό προς το χρήστη, αξιόπιστο, εύρωστο, μεταφέρσιμο.**
- Συντομία προτάσεων
- Ομοιομορφία προτάσεων
  - Ο γραμματέας θα μπορεί να …
  - Ο πελάτης θα μπορεί να ..
  - Ο υπεύθυνος παραγωγής θα μπορεί να ...

Ποιος ορίζει τις απαιτήσεις;

Πελάτες και Κατασκευαστές (αναλυτές) αμφότεροι είναι υπεύθυνοι για αυτό

Πως και πότε ορίζονται οι απαιτήσεις;

- *Είναι μια **επαναληπτική (iterative)** και **συνεχής (ongoing)** διαδικασία*
- *Αρχικά χρησιμοποιούμε **τεχνικές συλλογής απαιτήσεων** (requirements-gathering techniques)*
  - *(τι οποίες θα περιγράψουμε αργότερα)*
- *Εν συνεχεία, τις **επικυρώνουμε (verify)**, τις **εκλεπτύνουμε (refine), τροποποιούμε (modify), συμπληρώνουμε (complete),** και **ιεραρχούμε (prioritize).***
- *Σε κάθε χρονική στιγμή, το **Έγγραφο Απαιτήσεων** πρέπει να αντανακλά την τρέχουσα κατάσταση*

Προσοχή

- **Οι αλλαγές** πρέπει να γίνονται με προσοχή (δεν πρέπει να ξεφύγουμε πέραν τις εμβέλειας του συστήματος)

Σε ποια μορφή τις εκφράζουμε;

Υπάρχουν πολλές μορφές:

- Φυσική Γλώσσα
- Διαγράμματα Απαιτήσεων (Requirement Diagrams)
- Διαγράμματα Ροής Δεδομένων
- Διαγράμματα Warnier
- SADT
- Διαγράμματα UML
- Τυπικές μέθοδοι
- ..

που τις καταγράφουμε;

Έγγραφο Περιγραφής Απαιτήσεων
(Requirements Definition (Specification) Document)

Επίσης τα σύγχρονα εργαλεία CASE παρέχουν αρκετούς τρόπους για να τις εκφράσουμε και να τις οργανώσουμε.

Έγγραφο Περιγραφής Απαιτήσεων
(Requirements Definition (Specification) Document)

Έγγραφο Απαιτήσεων

Μια αναφορά που

- καταγράφει όλες τις ΛΑ και ΜΛΑ
- τις αριθμεί και τις ομαδοποιεί σε ΛΑ και ΜΛΑ
- ενδεχομένως τις ομαδοποιεί και ανάλογα με τη λειτουργία ή τον τύπο των ΜΛΑ
- τις διαβαθμίζει ανάλογα με την προτεραιότητά της (Υψηλή, Μέτρια, Χαμηλή)
- ενδεχομένως τις μαρκάρει με τον αριθμό έκδοσης (*release number)* που (σύμφωνα με το πρόγραμμα του έργου) θα τις πραγματώσει

Θυμηθείτε τους **MoSCoW rules** (**Must Should Could Want**) (μάθημα 5ο)

Παράδειγμα: *Κειμενογράφος*

Γ. Λειτουργικές Απαιτήσεις (ΛΑ)

1. Εκτύπωση

1.1. Ο χρήστης θα μπορεί να επιλέγει τις προς εκτύπωση σελίδες

1.2. Ο χρήστης θα μπορεί να βλέπει μια προεπισκόπηση πριν εκτυπώσει

1.3. Ο χρήστης θα μπορεί να αλλάζει τα περιθώρια, τον τύπο χαρτιού και τον προσανατολισμό της σελίδας

2. Ορθογραφικός έλεγχος

2.1. Το σύστημα πρέπει να έχει μια κατάσταση λειτουργίας στην οποία να ελέγχεται η ορθογραφία

2.1.1. Κατάσταση 1 (χειροκίνητη): Ο χρήστη θα ενεργοποιεί τον ορθογραφικό έλεγχο και το σύστημα θα μεταβαίνει στην πρώτη λανθασμένη λέξη

2.1.2. Κατάσταση 2 (αυτόματη): Ο ορθογραφικός έλεγχος θα γίνεται καθώς ο χρήστης πληκτρολογεί. Στην περίπτωση λάθους το σφάλμα θα υπογραμμίζεται.

2.2. Ο χρήστης θα μπορεί να προσθέτει νέες λέξεις στο λεξικό

2.3. Ο χρήστης μπορεί να μαρκάρει μια λανθασμένη λέξη ως αποδεκτή χωρίς να είναι υποχρεωμένος να την προσθέσει στο λεξικό.

Παράδειγμα: *Κειμενογράφος*

Δ. Μη Λειτουργικές Απαιτήσεις (ΜΛΑ)

1. Επιχειρησιακές

1.1. Το σύστημα θα μπορεί λειτουργεί σε περιβάλλον Windows και Macintosh

1.2. Το σύστημα θα μπορεί να αναγνώσει και να εγγράψει έγγραφα τύπου (.doc, .rtf, .html)

1.3. Το σύστημα θα επιτρέπει την εισαγωγή εικόνων (.gif, .jpg, .bmp) σε ένα έγγραφο

2. Επιδόσεων

2.1. Ο χρόνος απόκρισης δεν πρέπει ποτέ να υπερβαίνει το 1 δευτερόλεπτο

2.2 Το μέγεθος του αποθηκευτικού χώρου στο δίσκο για έναν έγγραφο πρέπει να είναι μικρότερο απ' ότι στο Microsoft Word

3. Ασφάλειας

3.1. Καμία ιδιαίτερη απαίτηση ασφάλειας δεν προβλέπεται.

Το έγγραφο μπορεί να περιέχει και ένα Γλωσσάρι

Για αποφυγή παερμηνειών και επίτευξη σαφήνειας και λιτότητας στο κείμενο (συνήθως έχει τη μορφή πίνακα)

Παράδειγμα Γλωσσαρίου για μια εφαρμογή τηλεαγοράς

Term	definition
bonus campaign	A special serious of activities, conducted within a *campaign*, to additionally entice *supporters* to buy the campaign *tickets*. Typical examples are giving free tickets for bulk or early buying or for attracting new supporters. A particular kind of bonus campaign can be used in many campaigns.
campaign	A government approved and carefully planned series of activities which are intended to achieve a *lottery* objective.
draw	An act of randomly choosing a particular *lottery ticket* as a winning ticket.
lottery	A funds raising game of chance, organized by the charity in order to make money, in which people (*supporters*) buy numbered *tickets* to have a chance of winning a *prize* if their number is chosen in a *draw*.
placement	Acquisition of one or more *lottery tickets* by a *supporter* during *telemarketing*. The placement is paid by a supporter with a credit card.

Πρότυπα (templates) για το Έγγραφο Απαιτήσεων

Υπάρχουν πολλά. Μερικά παραδείγματα:

- Volere Requirements Specification Template
  - http://www.systemsguild.com/GuildSite/Robs/Template.html
- Adaptable Process Model Software Requirements Specification
  - http://www.rspa.com/docs/Reqmspec.html
- IEEE Standard for SRS

Άλλα:

- Η πρότυπο αναφοράς που θα σας δοθεί.
- FASTAXON requirements document

Τι μας προτείνει η Αντικειμενοστρεφής Μεθοδολογία Ανάλυσης και Σχεδίασης;

- Έναρξη με ***Περιπτώσεις Χρήσης*** (***Use Cases)***
  - ~ Σενάρια από τα οποία μπορούμε να συνάγουμε τις ΛΑ και τις ΜΛΑ
    - Τα σενάρια αυτά μπορεί να περιγράφουν επιθυμητές και ανεπιθύμητες ακολουθίες συμβάντων (γεγονότων)
- Εν συνεχεία, μοντελοποίηση με διαγράμματα της UML

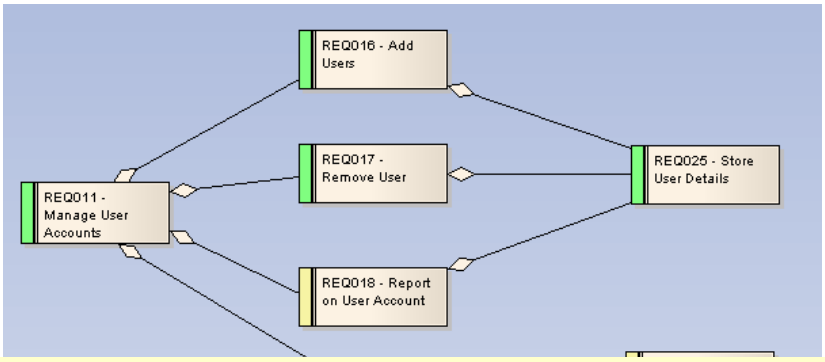
Παραδείγματα Διαγραμματικών Τεχνικών που υποστηρίζονται από εργαλεία CASE

Γενικά Σχόλια

- Δεν ανήκουν στην UML αλλά πολλοί τα χρησιμοποιούν.
- Επίσης αρκετά εργαλεία CASE τα υποστηρίζουν

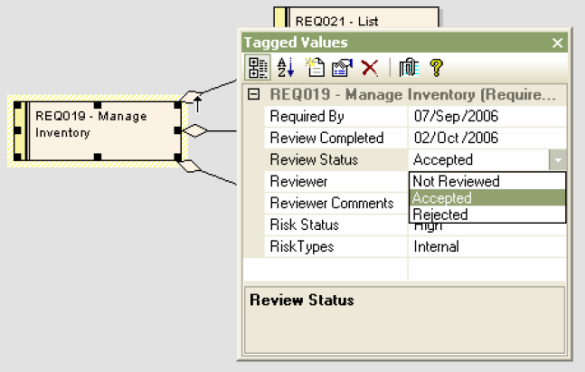
Requirements Management in EA (Enterprise Architect)

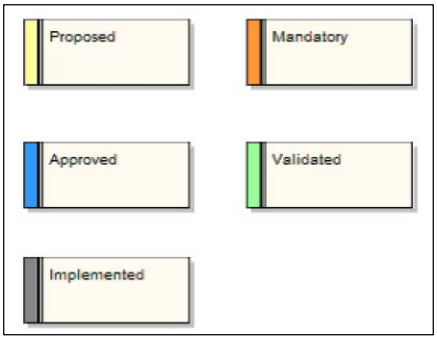
- A ***Requirements* diagram** is a custom diagram used to describe a system's requirements or features as a visual model.
- Requirements are defined using *Requirement elements* (*Custom* elements of type *Requirement*).. Requirement elements can be linked back to *Use Cases* and *Components* in the system to illustrate how a particular system requirement is met.
- Requirements models provide extensions to the UML model and enable traceability between specifications and design requirements, and the model elements that realize them.



- Δείτε το http://www.sparxsystems.com.au/downloads/whitepapers/Requirements_Management_in_Enterprise_Architect.pdf

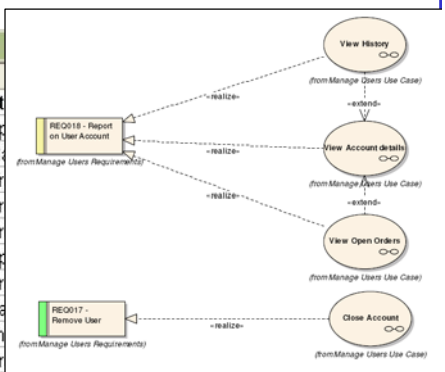
Requirements Management in EA (Enterprise Architect)



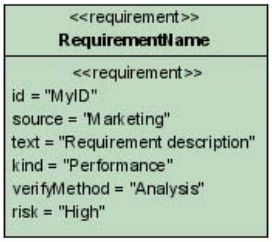


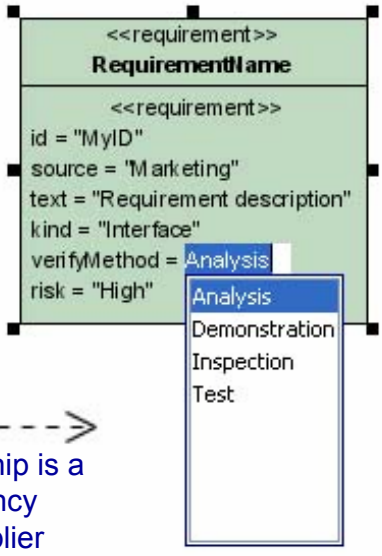
req.xls

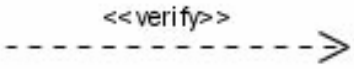
	A	B	C	D	E	
1	**Name**	**Type**	**Notes**	**Phase**	**Version**	**St**
2	FR27 - Retrieval of histo	Requirement	The system must be able to	1	1	Ap
3	FR28 - Transmission of	Requirement		1	1	Ma
4	FR29 - The staff must n	Requirement	The system must use standard u	1	1	Pr
5	FR30 - The interface mu	Requirement	Critical to the success of the app	1	1	Pr
6	FR44 - The interface sh	Requirement	An important aspect of the propo	1	1	Pr
7	FR5 - 2000 hours mean	Requirement	The Mean time between failure (	1	1	Ap
8	FR6 - Must be recoverat	Requirement	In the event of software or hardwa	1	1	Pr
9	FR7 - 99.999% accurac	Requirement	The system accuracy defines	1	1	Va
10	FR8 - 99.999% precisio	Requirement	The precision	1	1	Im
11	FR17 - All messages m	Requirement		1	1	Pr
12	FR34 - Physical storage	Requirement		1	1	Proposed Medium Medium



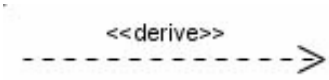
Visual Paradigm



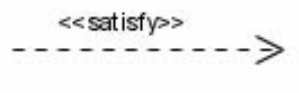




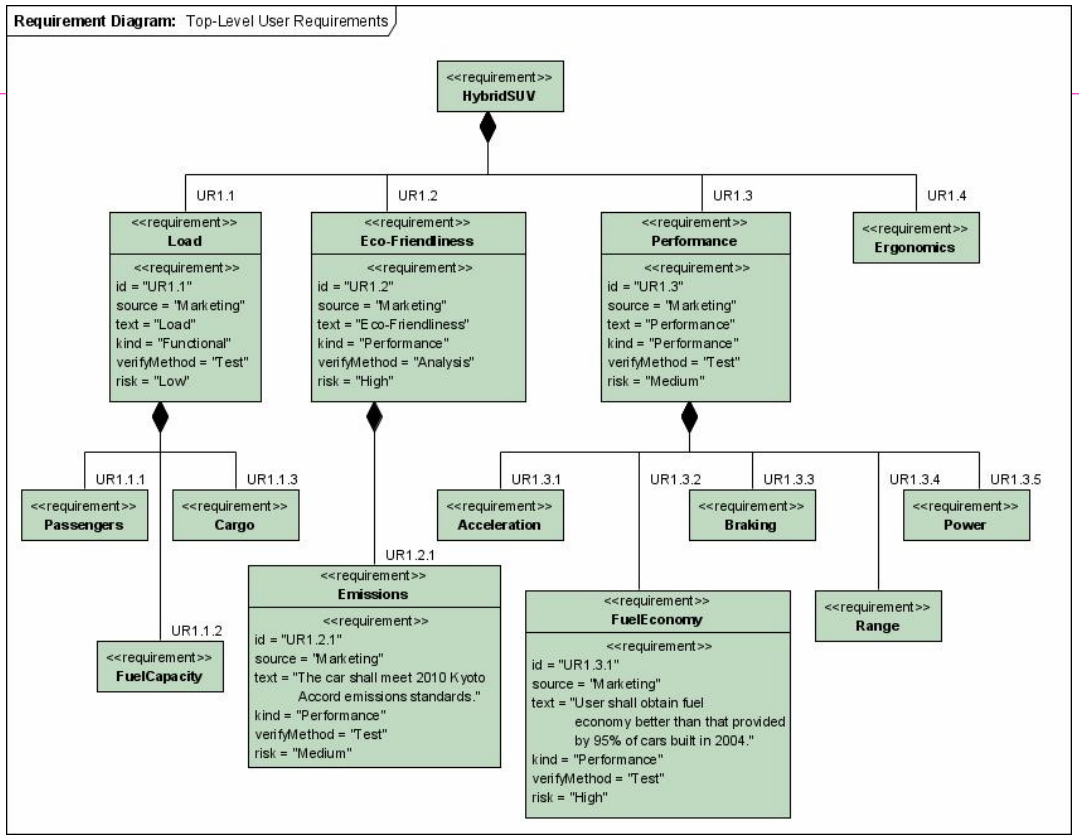
•A Verify relationship is a trace dependency between a supplier requirement and a client test case that determines whether a system fulfills the requirement.

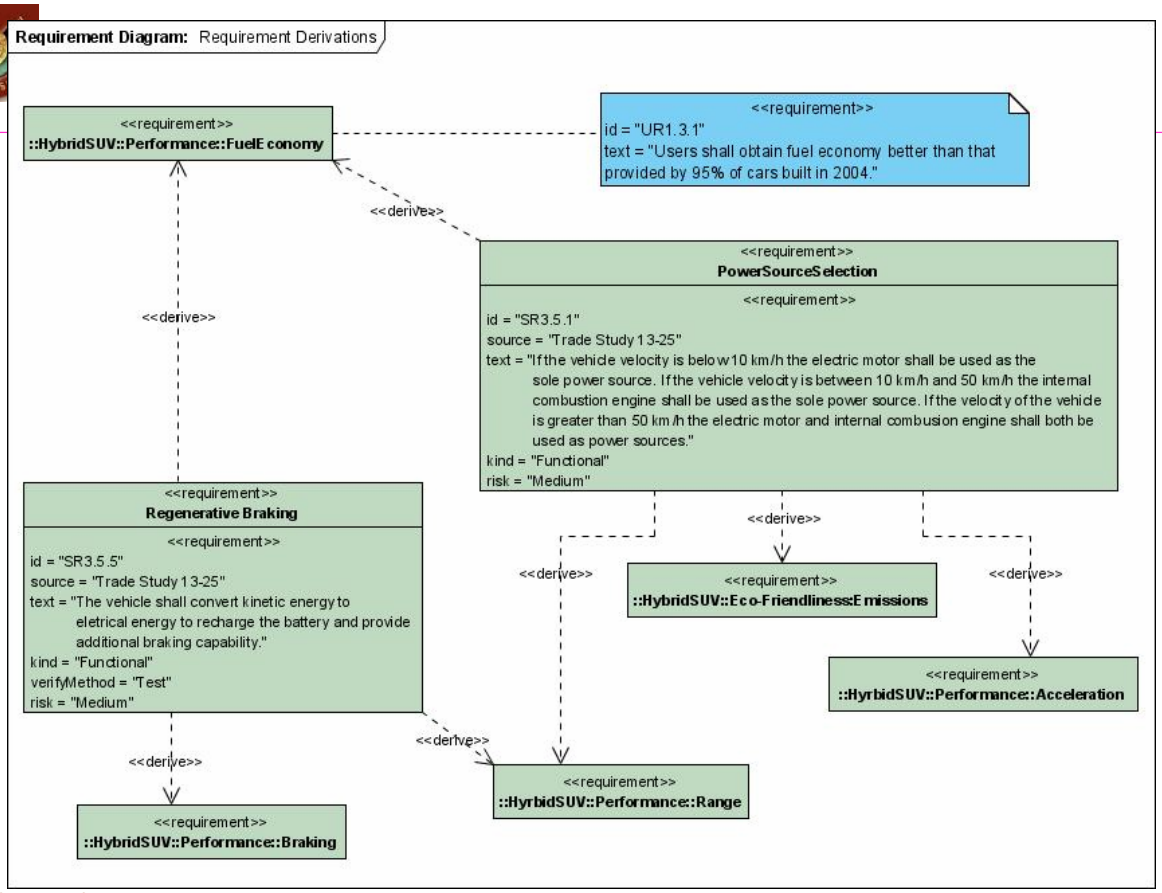


A **Derive** relationship is a trace dependency between a derived requirement and a source requirement, where the derived requirement is generated or inferred from the source requirement.

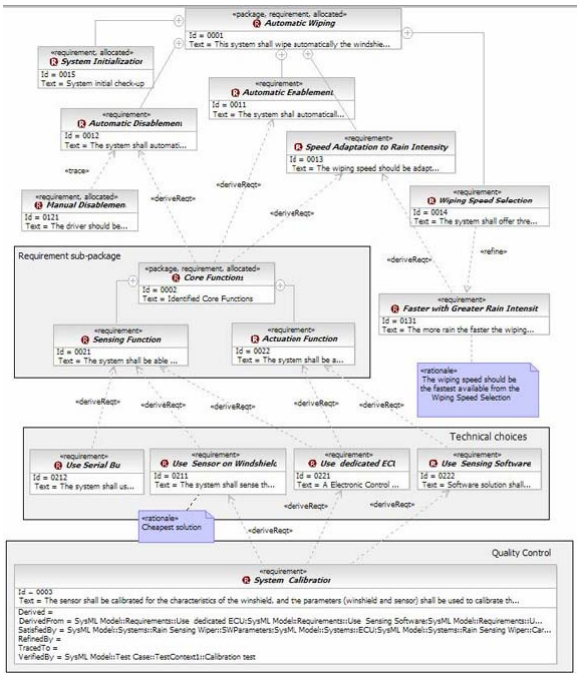


A **Satisfy** relationship is dependency between a supplier requirement and a client model element that fulfills the requirement.


Requirement Diagram: Top-Level User Requirements
<<requirement>>
HybridSUV
UR1.1
UR1.2
UR1.3
UR1.4
<<requirement>>
Load
<<requirement>>
id = "UR1.1"
source = "Marketing"
text = "Load"
kind = "Functional"
verifyMethod = "Test"
risk = "Low"
<<requirement>>
Eco-Friendliness
<<requirement>>
id = "UR1.2"
source = "Marketing"
text = "Eco-Friendliness"
kind = "Performance"
verifyMethod = "Analysis"
risk = "High"
<<requirement>>
Performance
<<requirement>>
id = "UR1.3"
source = "Marketing"
text = "Performance"
kind = "Performance"
verifyMethod = "Test"
risk = "Medium"
<<requirement>>
Ergonomics
UR1.1.1
UR1.1.3
<<requirement>>
Passengers
<<requirement>>
Cargo
UR1.2.1
UR1.3.1
UR1.3.2
UR1.3.3
UR1.3.4
UR1.3.5
<<requirement>>
Acceleration
<<requirement>>
Braking
<<requirement>>
Power
<<requirement>>
Emissions
<<requirement>>
id = "UR1.2.1"
source = "Marketing"
text = "The car shall meet 2010 Kyoto Accord emissions standards."
kind = "Performance"
verifyMethod = "Test"
risk = "Medium"
<<requirement>>
FuelEconomy
<<requirement>>
id = "UR1.3.1"
source = "Marketing"
text = "User shall obtain fuel economy better than that provided by 95% of cars built in 2004."
kind = "Performance"
verifyMethod = "Test"
risk = "High"
<<requirement>>
Range
UR1.1.2
<<requirement>>
FuelCapacity


Requirement Diagram: Requirement Derivations
<<requirement>>
::HybridSUV::Performance::FuelEconomy
<<requirement>>
id = "UR1.3.1"
text = "Users shall obtain fuel economy better than that provided by 95% of cars built in 2004."
<<derive>>
<<requirement>>
PowerSourceSelection
<<requirement>>
id = "SR3.5.1"
source = "Trade Study 13-25"
text = "If the vehicle velocity is below 10 km/h the electric motor shall be used as the sole power source. If the vehicle velocity is between 10 km/h and 50 km/h the internal combustion engine shall be used as the sole power source. If the velocity of the vehicle is greater than 50 km/h the electric motor and internal combusion engine shall both be used as power sources."
kind = "Functional"
risk = "Medium"
<<derive>>
<<requirement>>
Regenerative Braking
<<requirement>>
id = "SR3.5.5"
source = "Trade Study 13-25"
text = "The vehicle shall convert kinetic energy to eletrical energy to recharge the battery and provide additional braking capability."
kind = "Functional"
verifyMethod = "Test"
risk = "Medium"
<<derive>>
<<requirement>>
::HybridSUV::Eco-Friendliness::Emissions
<<derive>>
<<requirement>>
::HybridSUV::Performance::Acceleration
<<derive>>
<<derive>>
<<requirement>>
::HybridSUV::Performance::Range
<<derive>>
<<requirement>>
::HybridSUV::Performance::Braking

Requirements Diagram in Systems Modeling Language (IBM)



Πηγές

- **Systems Analysis and Design with UML Version 2.0** (2nd edition) by A. Dennis, B. Haley Wixom, D. Tegarden, Wiley, 2005. CHAPTER 5
- **Requirements Analysis and System Design** (2nd edition) by Leszek A. Maciaszek, Addison Wesley, 2005, Chapter 2
- Δείτε το http://www.sparxsystems.com.au/downloads/whitepapers/Requirements_Management_in_Enterprise_Architect.pdf
  - Θα το χρειαστείτε στην εργασία σας
- Δείτε το Requirements Tutorial with Visual Paradigm
  - http://www.visual-paradigm.com/product/vpuml/demos/requirements/requirement.jsp