



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Τεχνολογία Λογισμικού

8ο Εξάμηνο 2019-20

Ανάλυση απαιτήσεων

Δρ. Κώστας Σαΐδης (saiko@di.uoa.gr)

Ανάλυση απαιτήσεων

1. Οι απαιτήσεις λογισμικού και τα είδη τους
2. Από την ανάλυση απαιτήσεων στη σύνταξη των προδιαγραφών και ο ρόλος του επικεφαλής μηχανικού (EM)

1. Ανάλυση απαιτήσεων

Απαιτήσεις λογισμικού

Συνήθως είναι ανεξάρτητες από την τεχνολογία υλοποίησης.

- Απαίτηση: Τι θέλουμε να κάνει το λογισμικό
- Ικανοποίηση απαίτησης: Πώς το λογισμικό καλύπτει την απαίτηση (σχεδιαστική απόφαση και υλοποίηση).

Εκτός, βέβαια, αν η υποστήριξη κάποιας τεχνολογίας είναι ρητή απαίτηση

Είδη απαιτήσεων

- Λειτουργικές απαιτήσεις
- Μη λειτουργικές απαιτήσεις
- Απαιτήσεις συστήματος
- Αναδυόμενες απαιτήσεις

Λειτουργικές απαιτήσεις

- Οι απαιτήσεις της λειτουργίας του λογισμικού (συνήθως η νέα λειτουργικότητα που πρέπει να αναπτυχθεί στο πλαίσιο του έργου)
- Παραδείγματα: διαπίστευση και δικαιοδοσία χρηστών, επιχειρησιακή λογική, οι λειτουργίες που απαιτούνται από τις περιπτώσεις και τα σενάρια χρήσης, διαχειριστικές λειτουργίες

Διαπίστευση χρηστών (user authentication)

- Σημεία εισόδου στο σύστημα (entry-points) - μηχανισμοί διαπίστευσης χρηστών
- Επιβεβαίωση ότι ο χρήστης είναι:
 - "υπαρκτός" (π.χ. ο λογαριασμός του είναι εγγεγραμμένος στον "πίνακα" των χρηστών της ΒΔ)
 - "έγκυρος" (π.χ. γνωρίζει το προσωπικό του "μυστικό" που έχει συνδεθεί με το λογαριασμό του)
 - "ενεργός" (π.χ. δεν έχει απενεργοποιηθεί από το διαχειριστή)

Δικαιοδοσία χρηστών (user authorization)

- Διαχείριση δικαιωμάτων: ποιος χρήστης επιτρέπεται να εκτελέσει ποια ενέργεια.
- Στατική ή δυναμική ανάθεση/ανάκληση δικαιωμάτων;
- Ρόλοι
- Access Control Lists (ACLs)
- Guards

Επιχειρησιακή λογική (business logic)

- Οι επιχειρησιακές διαδικασίες (business processes)
- Οι επιχειρησιακοί κανόνες (business rules)
- Μερική ή πλήρης "ηλεκτρονικοποίηση" των λειτουργιών του οργανισμού

Πιο απλά

- Οι λειτουργίες που περιλαμβάνονται στα υποστηριζόμενα σενάρια και περιπτώσεις χρήσης (use cases) του λογισμικού
- Οι ενέργειες, λειτουργίες και διαδικασίες που εκτελούν οι χρήστες μέσω του συστήματος
- Τα user stories σε ένα Agile backlog

Διαχειριστικές λειτουργίες (administrative functions)

- Ποιες είναι οι διαχειριστικές λειτουργίες;
- Πώς εκτελούνται;
- Θα παραμετροποιούνται από το χρήστη και πόσο;

Λειτουργίες αναφορών (reporting requirements)

- Τι αναφορές απαιτούνται ;
- Πόσο παραμετροποιήσιμες πρέπει να είναι;
- Ποιος έχει δικαίωμα να τις προσπελάσει;

Μη λειτουργικές απαιτήσεις

- Απαιτήσεις για ποιοτικά χαρακτηριστικά του λογισμικού
 - Διαθεσιμότητα - Ανάνηψη από καταστροφές
 - Ασφάλεια - Ακεραιότητα
 - Ευελιξία - Επεκτασιμότητα
 - Απόδοση - Απροκρισιμότητα
 - Υποστήριξη διεθνών προτύπων
- Καθορίζουν σε μεγάλο βαθμό την επιτυχία του έργου

Απαιτήσεις συστήματος

- Απαιτήσεις σχετικές με το "σύστημα" ή το "περιβάλλον" στο οποίο εντάσσεται το υπό ανάπτυξη λογισμικό (π.χ. μπορεί να είναι συστατικό ενός μεγαλύτερου όλου)
- Απαιτήσεις υλικού, απαιτήσεις δικτύου, κ.ά

Αναδυόμενες απαιτήσεις

- Ποιοτικά χαρακτηριστικά που προκύπτουν από την ένταξη του λογισμικού στο "σύστημα" ή "περιβάλλον" και αναφέρονται στη λειτουργία του ως όλον.

Στην πράξη

- Στην πράξη, κάποιες απαιτήσεις μπορεί να βρίσκονται "κάπου ανάμεσα" στα παραπάνω είδη
- Αλλά αυτό ποικίλει ανάλογα με το υπό ανάπτυξη λογισμικό, το περιβάλλον λειτουργίας του και τους εμπλεκόμενους φορείς - χρήστες του
- Ας δούμε μερικά παραδείγματα

Τήρηση ημερολογίων (audit tracking)

- Ποιες ενέργειες καταγράφονται σε ημερολόγια;
- Σε ποιο βαθμό λεπτομέρειας;
- Για ποιο σκοπό και με ποια τελική χρήση;

Μαζική εισαγωγή / εξαγωγή δεδομένων (Data import / export)

- Απατείται η μαζική εισαγωγή / εξαγωγή δεδομένων;
- Θα παραμετροποιείται από το χρήστη και πόσο;
- Ποιοι μορφότυποι αρχείων / δεδομένων θα πρέπει να υποστηρίζονται;

Υποστήριξη διεθνούς προτύπου

- Ανάλογα με την περίπτωση, μπορεί να επηρεάζει:
 - Ένα συγκεκριμένο συστατικό
 - Ένα υποσύνολο των συστατικών
 - Το σχεδιασμό συνολικά

Διαλειτουργικότητα (interoperability)

- Το υπό ανάπτυξη σύστημα θα διαλειτουργεί με τρίτα συστήματα;
- Τι είδους δεδομένα θα ανταλλάσσονται;
- Με ποιους μηχανισμούς, μορφότυπους και πρωτόκολλα;

Νομικές ή κανονιστικές απαιτήσεις

- Το υπό ανάπτυξη σύστημα θα πρέπει να είναι σύννομο (προσωπικά δεδομένα, πνευματικά δικαιώματα, κτλ.)
- Το υπό ανάπτυξη σύστημα μπορεί να διέπεται από συγκεκριμένο νομικό ή κανονιστικό πλαίσιο (για παράδειγμα, πληροφοριακά συστήματα δημοσίου τομέα).

Ορισμός των απαιτήσεων

- Σαφήνεια (η απαίτηση είναι σαφώς διατυπωμένη)
- Συνέπεια (η μία απαίτηση δεν αντιβαίνει με την άλλη)
- Πληρότητα (η απαίτηση καλύπτει όλες τις περιπτώσεις)
- Επαληθευσιμότητα (η απαίτηση μπορεί να ελεγχθεί σε σχέση με την υλοποίηση)

Επίσης

Σε μεγάλα έργα

- Καταγωγή/γενεαλογία απαίτησης
 - Ποιος τη ζήτησε
 - Ποιον επιχειρησιακό στόχο εξυπηρετεί

Για παράδειγμα, θα πρέπει να γνωρίζει η ομάδα του έργου αν "το ζητάει ο διευθυντής"

Έγγραφο ανάλυσης απαιτήσεων

Περιγραφή σε φυσική γλώσσα:

- Σκοπός του συστήματος
- Κατηγορίες χρηστών (users & stakeholders)
- Εμβέλεια του συστήματος
- Περιορισμοί
- Παραδοχές
- Απαιτήσεις (λειτουργικές ή μη)

2. Από την ανάλυση απαιτήσεων στη σύνταξη των προδιαγραφών και ο ρόλος του επικεφαλής μηχανικού

Ζητούμενα από το κείμενο των τεχνικών προδιαγραφών

- Να προσδιορίσουν τον τρόπο με τον οποίο το λογισμικό θα ικανοποιήσει τις απαιτήσεις

Συγκεκριμένα

- Αρχιτεκτονικός σχεδιασμός και συστατικά του λογισμικού
- Σαφής περιγραφή της αλληλεπίδρασης του λογισμικού με το παραγωγικό του περιβάλλον (production environment)
- Λεπτομερής προδιαγραφή των λειτουργιών και των διαδικασιών που υποστηρίζονται από το λογισμικό
- Σύνδεση των προδιαγραφών με τις απαιτήσεις (πηγή/ προέλευση προδιαγραφής)
- Καθορισμός κριτηρίων αποδοχής / απόρριψης (τμημάτων ή όλου) του συστήματος

Έλεγχοι αποδοχής (acceptance tests)

- Έλεγχοι που καθορίζουν την αποδοχή ή την απόρριψη συστατικών ή λειτουργιών ή "οθονών" του λογισμικού
- Είναι καλή πρακτική να έχουν προβλεφθεί (μερικώς ή πλήρως) στις προδιαγραφές
- Ένας από τους κύριους στόχους ένταξης ελέγχων στον κύκλο του λογισμικού είναι η ελαχιστοποίηση των σφαλμάτων που θα προκύψουν κατά τους ελέγχους αποδοχής

Σχεδιαστικές αποφάσεις

- Για να μετατραπούν οι υψηλού επιπέδου απαιτήσεις του λογισμικού
- Σε χαμηλού επιπέδου τεχνικές προδιαγραφές αυτού
- Θα πρέπει να παρθούν συγκεκριμένες σχεδιαστικές αποφάσεις

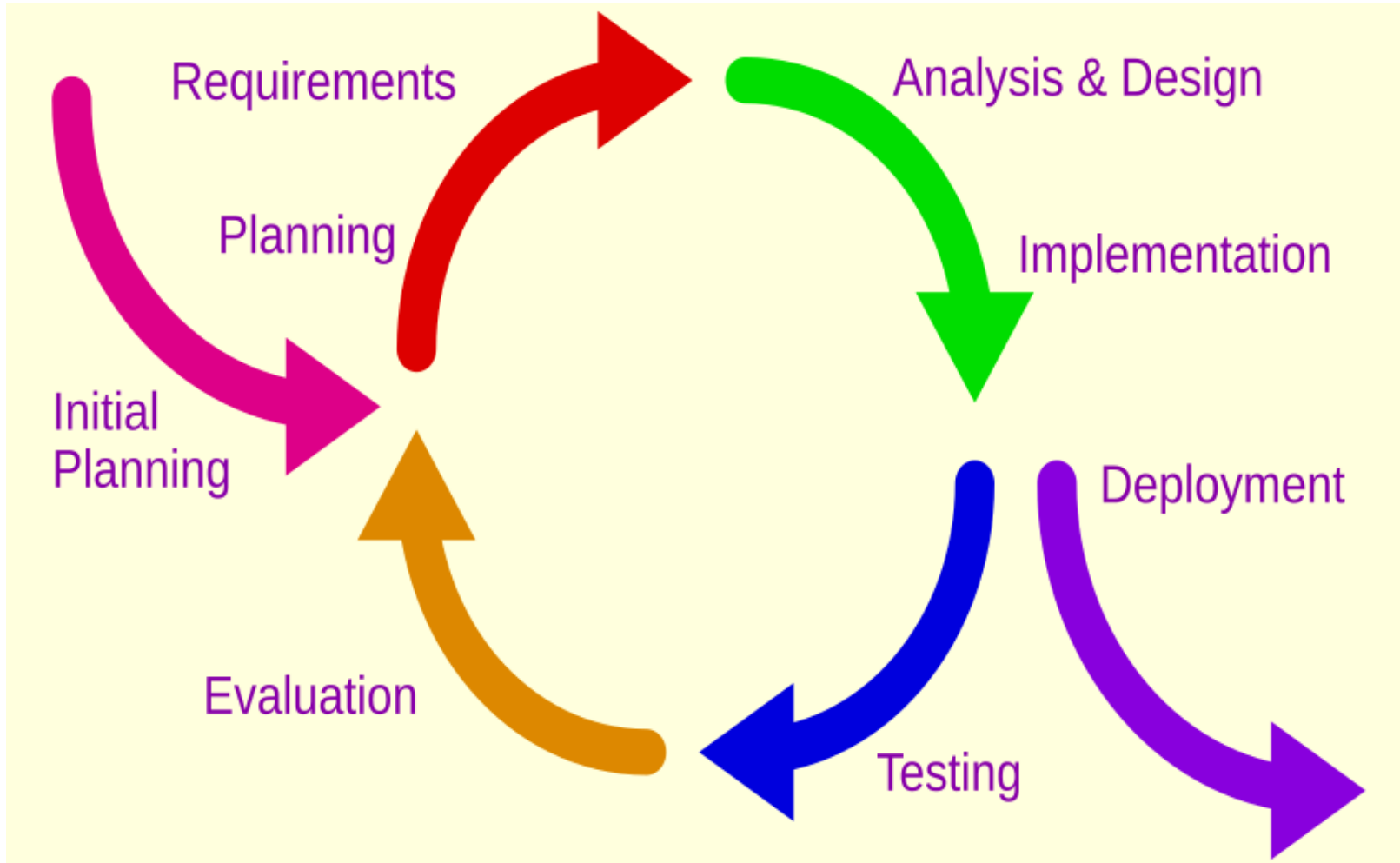
Δηλαδή

- Οι τεχνικές προδιαγραφές μοντελοποιούν τις απαιτήσεις εντός ενός οριοθετημένου σχεδιαστικού / αρχιτεκτονικού πλαισίου
- Δεν είναι πάντα εφικτό ο σχεδιασμός / αρχιτεκτονική να έχουν ολοκληρωθεί μέχρι τελευταίας λεπτομέρειας
- Από την άλλη δεν είναι αποδεκτό να προκύψει στην πορεία υλοποίησης μιας προδιαγραφής ένα ζήτημα που θα "ακυρώσει" το σχεδιασμό
- Το ότι θα τον αλλάξει είναι (σχεδόν) βέβαιο!

Σχεδιαστικές αποφάσεις

Οι αποφάσεις σχετικά με κάποια πτυχή του λογισμικού που είναι "ακριβό" ή "δύσκολο" να αλλάξουν άπαξ και παγιωθούν.

Θυμηθείτε



Κοινός τόπος

- Η ανάλυση των απαιτήσεων πρέπει να καταλήξει σε ένα κοινά αποδεκτό ζητούμενο
- Κάτι που μπορεί να είναι δύσκολο στην πράξη για μη τεχνικούς λόγους (επικοινωνίας, οργάνωσης, διαφορετικής κουλτούρας, παγίωσης μη βέλτιστων πρακτικών, κτλ.)

Θυμηθείτε από την εισαγωγή

Στα μεγάλα έργα λογισμικού εμπλέκονται πολλοί συμμετέχοντες:

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

Ο ρόλος του EM

Διαχείριση της πολυπλοκότητας

- 50% αρχιτεκτονική (ουσιώδης πολυπλοκότητα)
- 50% επικοινωνία (τεχνητή πολυπλοκότητα)
 - με τους stakeholders
 - με την επιχειρησιακή ομάδα έργου (Project Manager, Business Analyst, κ.ά)
 - με την ομάδα ανάπτυξης (engineers)

Κοινή γλώσσα

- Κάθε ένας από τους συμμετέχοντες έχει διαφορετικές εμπειρίες, βιώματα, αφετηρίες, στόχους και επιδιώξεις από την υλοποίηση του έργου
- Η ομάδα ανάπτυξης, η επιχειρησιακή ομάδα και οι stakeholders του έργου μάλλον "θα μιλάνε" σε διαφορετικές γλώσσες
- Ο EM θα πρέπει να κάνει το διερμηνέα και να "μιλάει" τη γλώσσα και των τριών

Ιεράρχηση απαιτήσεων

- Δεν είναι όλες οι απαιτήσεις εξίσου σημαντικές
- Μια μικρή -θεωρητικά- απαίτηση μπορεί να οδηγήσει το έργο εκτός δρόμου/στόχου
 - Η ομάδα ανάπτυξης ή η επιχειρησιακή ομάδα μπορεί να μη δώσουν προσοχή ("αφού το ζητάει ο πελάτης")
 - Ο stakeholder ενδέχεται να μην το καταλάβει
- Ο EM πρέπει να το διαγνώσει έγκαιρα και να προτείνει εναλλακτικές

Αντίστοιχα

- Μια "ακριβή" απαίτηση μπορεί να έχει πιο "φθηνή" εναλλακτική
- Μια "σύνθετη" απαίτηση μπορεί να έχει πιο "απλή" εναλλακτική
- κ.ο.κ

Προσοχή

- Ο ασφαλέστερος τρόπος να αποτύχει ένα έργο είναι να επιτραπεί στις "λάθος" απαιτήσεις να γίνουν προδιαγραφές!

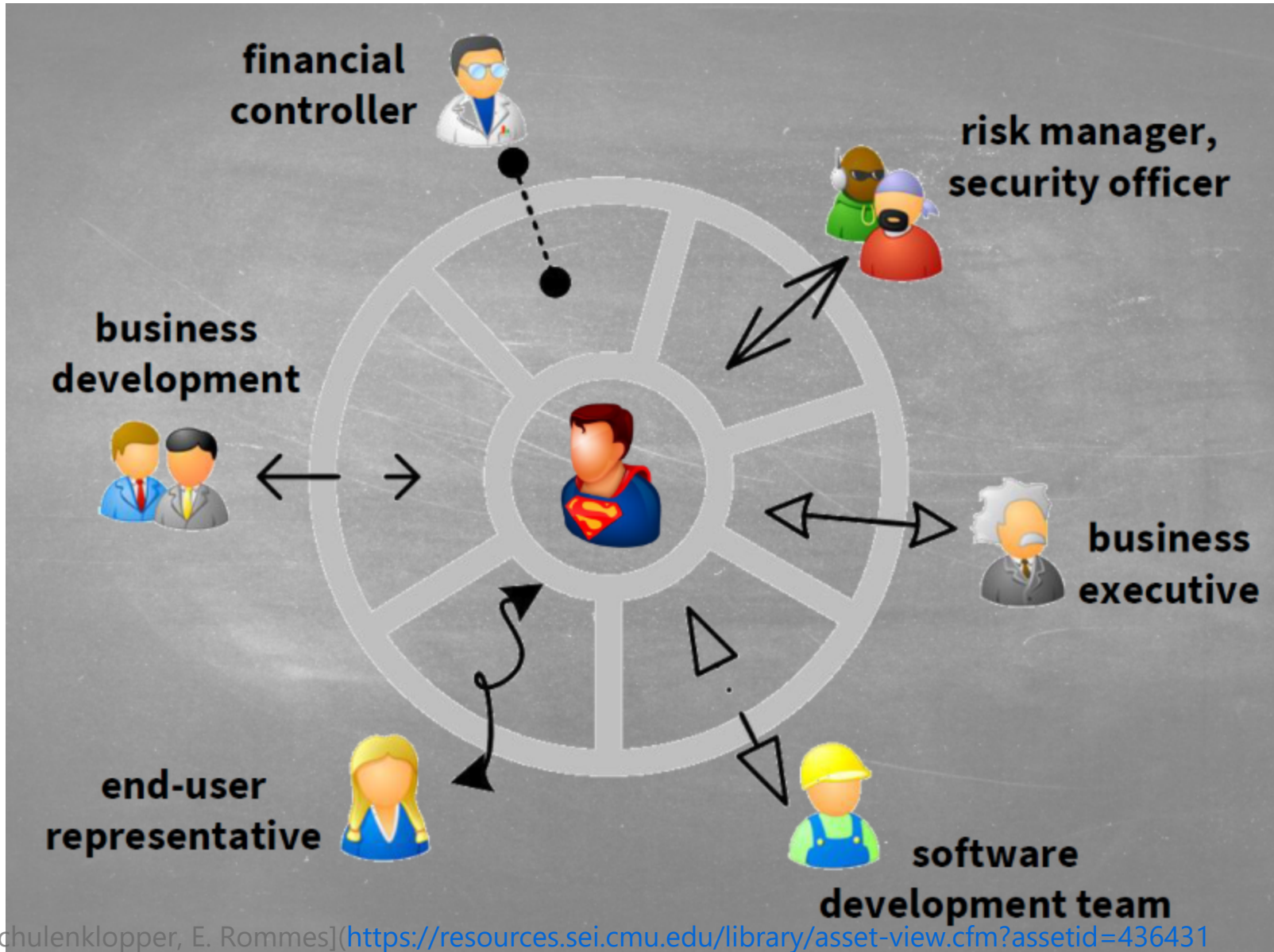
Συμπέρασμα

- Ο ΕΜ θα πρέπει να έχει απευθείας επαφή με τους stakeholders
 - Αναλύοντας και διαμορφώνοντας μαζί με την επιχειρησιακή ομάδα του έργου τις απαιτήσεις
- Ενώ παράλληλα σχεδιάζει το σύστημα
 - Για να είναι σε θέση να φιλτράρει τις "λάθος" απαιτήσεις
 - Και να μεταφράσει τις υπόλοιπες σε προδιαγραφές για την ομάδα ανάπτυξης

The belief that complex systems require armies of programmers and designers is wrong. A system that is not understood in its entirety, or at least to a significant degree of detail by a single individual, should probably not be built.

N. Wirth

Superman



Για να γίνεις Superman!

- Ενσυναίσθηση (empathy): Για τις ανάγκες και τους προβληματισμούς όλων των εμπλεκόμενων
- Τεχνική αρτιότητα: Για την επιλογή αρχιτεκτονικής / τεχνολογίας που να είναι συμβατή με τις ανάγκες του έργου και τις ικανότητες της ομάδας
- Επικοινωνιακή ικανότητα: Για τη συνεχή επικοινωνία με όλους τους εμπλεκόμενους για την εύρεση των βέλτιστων συμβιβασμών (trade-offs)
- Εμπειρία: Για να μπορούν να γίνουν καλά όλα τα παραπάνω

Ενδεικτικά 😊