



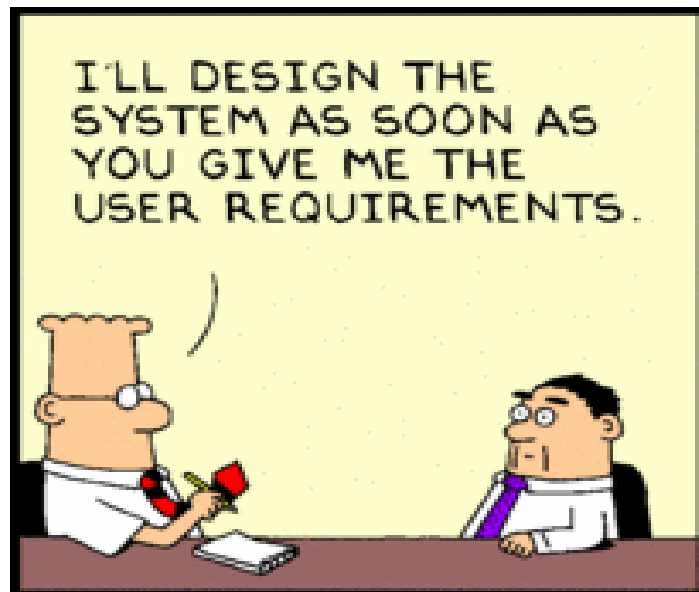
DIS501 Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων

**Lecture 4: Μηχανική Απαιτήσεων: Συλλογή Πληροφοριών,
Λειτουργικές και Μη Λειτουργικές Απαιτήσεις**

BY: ΔΡ ΑΥΓΟΥΣΤΑ ΚΥΡΙΑΚΙΔΟΥ ΖΑΧΑΡΟΥΔΙΟΥ

ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

1. Περιγραφή των κύριων δραστηριοτήτων της μηχανικής απαιτήσεων και των σχέσεων μεταξύ τους
2. Παρουσίαση μιας σειράς από τεχνικές εξαγωγής και ανάλυσης απαιτήσεων
3. Διάκριση μεταξύ λειτουργικών και μη λειτουργικών απαιτήσεων
4. Εισαγωγή στις ιστορίες χρηστών (user stories)

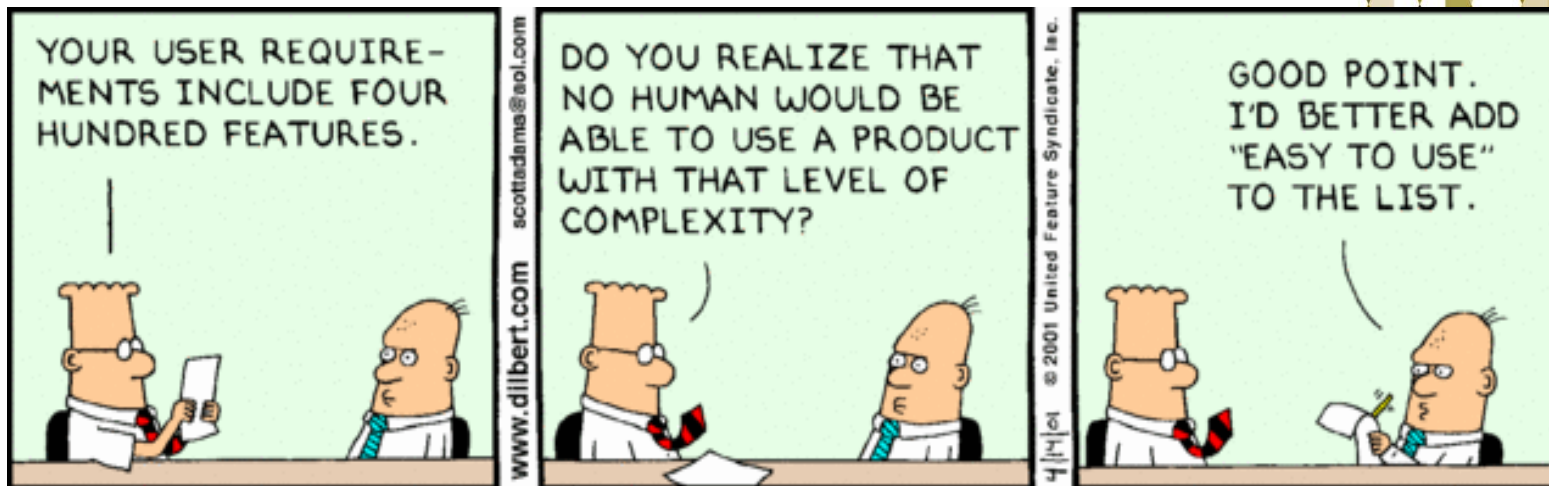


Στόχος της ανάπτυξης του συστήματος

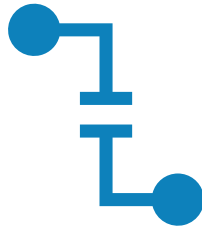
- Τι πρόκειται να κατασκευάσουμε (προϊόν) και πώς θα το κατασκευάσουμε (διαδικασία)

Και τα δύο ξεκινούν με την κατανόηση των απαιτήσεων

- Ποιο είναι το πρόβλημα που προσπαθούμε να λύσουμε, τι πρέπει να κάνει το προϊόν μας, τι λειτουργίες πρέπει να υποστηρίζει



ΤΙ ΕΊΝΑΙ Η ΑΠΑΙΤΗΣΗ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ



Ένα χαρακτηριστικό (feature) του νέου συστήματος που κάποιος επιθυμεί χρειάζεται.

Περιγραφή των λειτουργιών που θα κάνει το λογισμικό και των περιορισμών που το περιορίζουν.

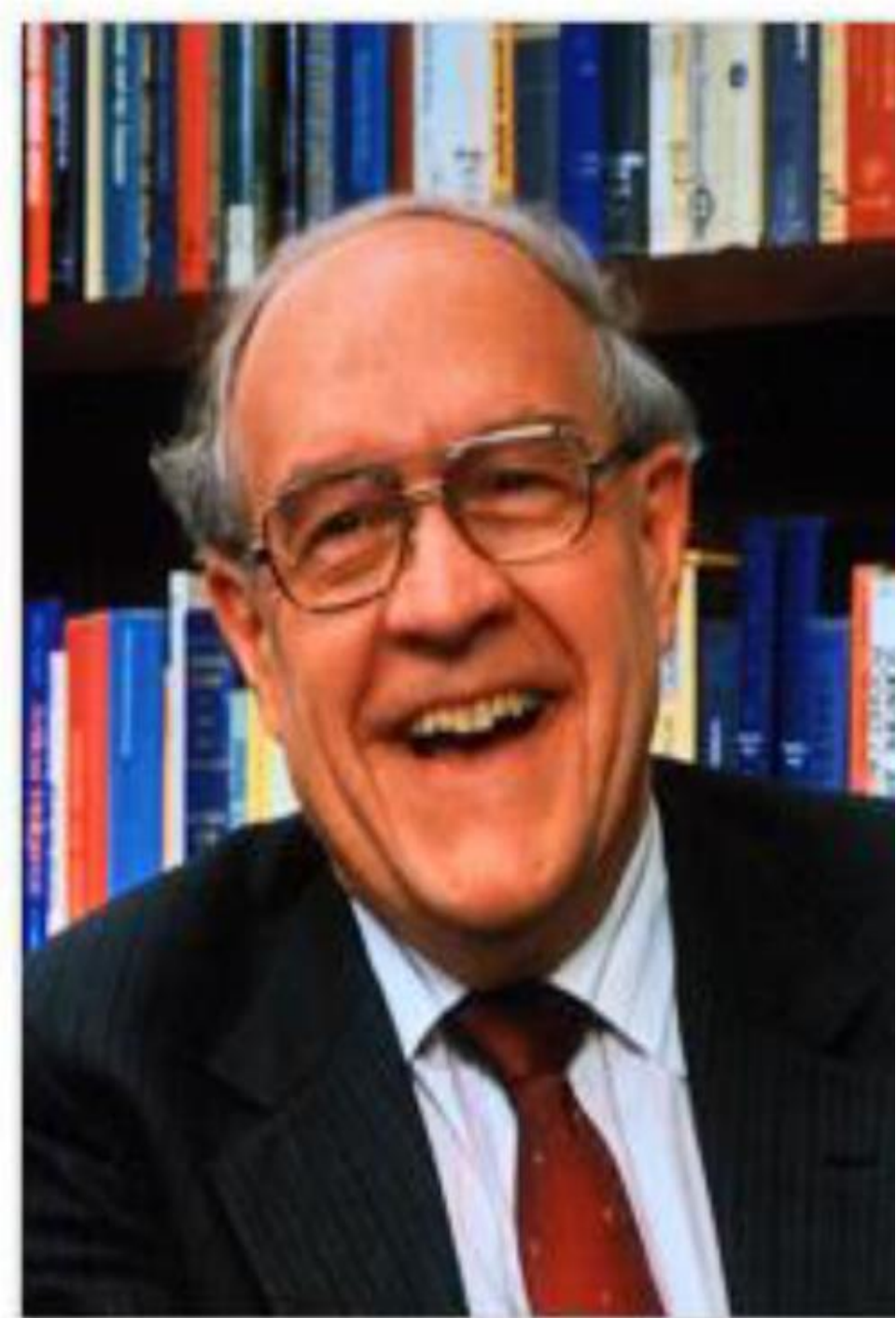
Μπορεί να είναι επίσης μια κριτική στις αδυναμίες ήδη υπάρχουσών συστημάτων



Τι θα κάνει το νέο σύστημα όσον αφορά:

- Δεδομένα που θα χρησιμοποιεί
- Διαδικασίες
- Συμπεριφορά

Πιστεύετε ότι ο καθορισμός
των απαιτήσεων για ένα
σύστημα είναι εύκολος;



"Το δυσκολότερο κομμάτι της δημιουργίας ενός συστήματος λογισμικού είναι να αποφασίσεις τι ακριβώς θα φτιάξεις. Κανένα άλλο μέρος δεν είναι τόσο δύσκολο όσο ο καθορισμός των λεπτομερών τεχνικών απαιτήσεων, συμπεριλαμβανομένων όλων των διεπαφών με τους ανθρώπους, τις μηχανές και τα άλλα συστήματα λογισμικού. Κανένα άλλο μέρος της εργασίας δεν μπορεί να προκαλέσει τόσο μεγάλο πρόβλημα στο σύστημα που θα δημιουργηθεί, αν πραγματοποιηθεί λανθασμένα. Κανένα άλλο μέρος δεν είναι πιο δύσκολο να διορθωθεί αργότερα".

Frederick Brooks 1987 (Turing Award 1999)

Η ΠΡΟΚΛΗΣΗ ΤΟΥ ΚΑΘΟΡΙΣΜΟΥ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ



Θα εκπλαγείτε από το πόσο δύσκολο είναι για τους περισσότερους ανθρώπους να εκφράσουν αυτό που θέλουν να αποκομίσουν από ένα νέο σύστημα ή μια νέα εφαρμογή στη δουλειά τους.



Οι αρχικές απαντήσεις είναι πολύ γενικές, όπως η ανάγκη να είναι σε θέση να ολοκληρώνουν μια εργασία ταχύτερα ή με πιο αποτελεσματικό τρόπο.



Η διοικητική ομάδα μπορεί να έχει γενικές απαιτήσεις, όπως να θέλει να βελτιώσει την παραγωγικότητα της εταιρείας ή να επωφεληθεί από μια νέα τεχνολογία.

ΕΜΠΟΔΙΑ ΕΠΙΚΟΙΝΩΝΙΑΣ - ΠΑΡΕΞΗΓΗΣΕΙΣ



From John Sowa, "Conceptual Structures: Information Processing in Mind and Machine."



Η ΠΡΟΚΛΗΣΗ ΤΟΥ ΚΑΘΟΡΙΣΜΟΥ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ

As an analyst, I need to know what do you want?



I want you to design the software for me.



But what do you want to do with the software?



I don't know until you tell me what the software can do.



Well, I can design the software to do anything!



Can you design the software to tell you my requirements?!



COMMUNICATION CHALLENGES VIDEO



ΠΡΟΚΛΗΣΕΙΣ ΣΤΗΝ ΔΙΑΔΙΚΑΣΙΑ ΜΗΧΑΝΙΚΗΣ ΑΠΑΙΤΗΣΕΩΝ (REQUIREMENTS ENGINEERING)

- Η μηχανική απαιτήσεων είναι μια δύσκολη δουλειά, ιδίως όταν το λογισμικό είναι μεγάλο και πολύπλοκο.
- Απαιτεί τρόπους εξεύρεσης και κατανόησης των απαιτήσεων του συστήματος από τις οπτικές γωνίες όλων των ενδιαφερομένων μερών
 - των χρηστών του συστήματος, της διοίκησης της εταιρείας, των πελατών, των προγραμματιστών του συστήματος κ.λπ.
- Υπάρχει εμπόδιο επικοινωνίας μεταξύ της ομάδας και του πελάτη και των χρηστών.
 - Η ομάδα δεν γνωρίζει το πεδίο εφαρμογής και τις ανάγκες του πελάτη και των χρηστών.
 - Ο πελάτης και οι χρήστες δεν γνωρίζουν τι μπορεί να κάνει το λογισμικό γι' αυτούς.
- Η σημασία και οι προκλήσεις της ανάδειξης απαιτήσεων συχνά υποτιμώνται.
- Οι **μη λειτουργικές απαιτήσεις** δεν προσδιορίζονται ή υποτιμώνται.
- Οι απαιτήσεις αλλάζουν καθ' όλη τη διάρκεια του κύκλου ζωής του λογισμικού.

REQUIREMENTS ENGINEERING - ΔΙΑΔΙΚΑΣΙΑ



Inception – Σύλληψη: Καθιέρωση μιας βασικής κατανόησης του προβλήματος, των ανθρώπων που επιθυμούν μια λύση και της φύσης της λύσης που είναι επιθυμητή, σημαντικό να καθιερωθεί αποτελεσματική επικοινωνία με τον πελάτη και τον μηχανικό λογισμικού.



Elicitation - Συλλογή: Επίσης, ονομάζεται καταγραφή απαιτήσεων, συλλογή απαιτήσεων και ανακάλυψη απαιτήσεων. Αναζήτηση/συλλογή απαιτήσεων και επιχειρησιακών αναγκών από όλα τα ενδιαφερόμενα μέρη.

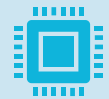


Elaboration- Επεξεργασία: Η ανάλυση των απαιτήσεων. Επικεντρώνεται στην ανάπτυξη ενός εκλεπτυσμένου μοντέλου απαιτήσεων που προσδιορίζει πτυχές της λειτουργίας και της συμπεριφοράς του λογισμικού. Λειτουργικές και μη λειτουργικές απαιτήσεις

REQUIREMENTS ENGINEERING - ΔΙΑΔΙΚΑΣΙΑ



Negotiation – Διαπραγμάτευση: συμφωνία σχετικά με το πεδίο εφαρμογής ενός παραδοτέου συστήματος (scope) που είναι ρεαλιστικό για τους developers και τους πελάτες.



Specification – Προδιαγραφές: Μοντελοποίηση λειτουργικών απαιτήσεων. Μπορεί να περιλαμβάνει οποιοδήποτε ή όλα τα ακόλουθα: γραπτά έγγραφα, γραφικά μοντέλα, σενάρια χρήσης και prototype του συστήματος.



Validation – Επικύρωση: Τα παραδοτέα που παράγονται κατά τη διάρκεια της μηχανικής απαιτήσεων αξιολογούνται ως προς την ποιότητα και τη συνέπεια.



Requirements management - Διαχείριση απαιτήσεων: σύνολο δραστηριοτήτων που βοηθούν την ομάδα έργου να εντοπίζει, να ελέγχει και να παρακολουθεί τις απαιτήσεις και τις αλλαγές στις απαιτήσεις κατά την πρόοδο του έργου.



REQUIREMENTS INCEPTION AND ELICITATION - ΣΥΛΛΗΨΗ ΚΑΙ ΣΥΛΛΟΓΗ ΣΤΟΧΩΙ

- **Τι θα κάνει το σύστημα; Ποιες λειτουργίες/χαρακτηριστικά θα υποστηρίξει το σύστημα**
- **Ποιος χρειάζεται/θα χρησιμοποιήσει το σύστημα;**
 - Προσδιορίστε τα ενδιαφερόμενα μέρη, τις ανησυχίες τους, τις ανάγκες τους, τους πολιτιστικούς παράγοντες που θα διαμορφώσουν τον τρόπο παρουσίασης/επικοινωνίας του συστήματος.
 - Stakeholder analysis
- **Τι θα πρέπει να κάνει το σύστημα για να καλύψει αυτές τις ανάγκες;**
 - Εξετάστε πώς θα γνωρίζουμε ότι κατασκευάζουμε το σωστό σύστημα (acceptance testing criteria) κατά τη φάση της συλλογής απαιτήσεων.
- **Οι τεχνολόγοι λογισμικού πρέπει να συνεργάζονται με τους πελάτες (end-users, managers, etc.) για να διαπιστώσουν:**
 - Τον τομέα της εφαρμογής/συστήματος
 - Τις λειτουργίες που πρέπει να παρέχει
 - Τους περιορισμούς



REQUIREMENTS ELICITATION TECHNIQUES

- **Interviewing** Συνεντεύξεις με χρήστες, διαχειριστές και άλλους ενδιαφερόμενους
- **Brainstorming in groups** - δημιουργικό, συμμετοχικό
- **Observing people at work** - Παρατήρηση ανθρώπων στην εργασία. Παθητική ή ενεργητική παρατήρηση.
- **Questionnaires and Surveys**
- **Rich Pictures**
- **Experimenting with a prototype** – δημιουργία ενός πρωτοτύπου συστήματος για ανατροφοδότηση από χρήστες
- **Ανάλυση του άμεσου περιβάλλοντος των χρηστών** - των τμημάτων ενός οργανισμού που το νέο σύστημα προορίζεται να υποστηρίξει
- **Εξέταση υφιστάμενων παρόμοιων συστημάτων**
- **Ερευνητική τεκμηρίωση, εκθέσεις και άρθρα περιοδικών που συζητούν για παρόμοια συστήματα και μελέτες περιπτώσεων.**

REQUIREMENTS ELICITATION TECHNIQUES

Χρήστες που είναι έμπειροι, ικανοί και πρόθυμοι να εκφράσουν με σαφήνεια τις απόψεις τους

και

Το νέο σύστημα θα βοηθήσει στην εκτέλεση μιας σταθερής διαδικασίας

τότε

➤ **Χρησιμοποιήστε τεχνικές όπως συνεντεύξεις, brainstorming κ.λπ. και εξετάζοντας τις διαδικασίες χειρισμού των υφιστάμενων συστημάτων.**

Οι χρήστες που είναι άπειροι, φοβισμένοι ή δεν μπορούν να εκφράσουν με σαφήνεια τις απόψεις τους

και

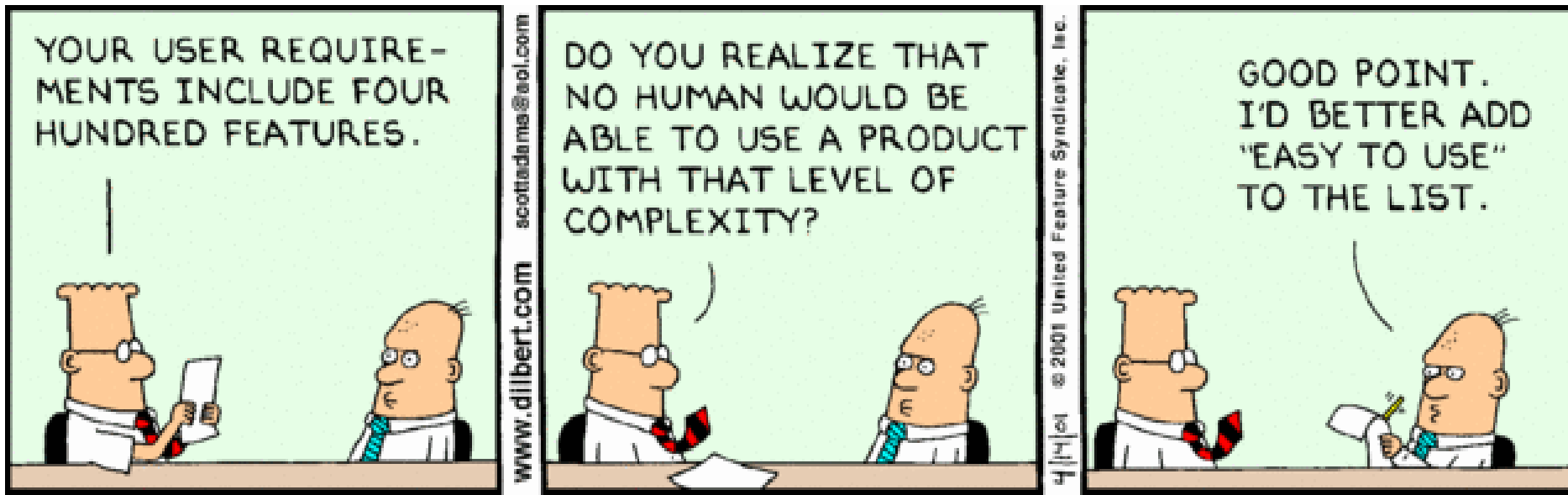
Το σύστημα αποτελεί μέρος μιας σκόπιμης προσπάθειας για την εισαγωγή καινοτομίας ή αλλαγής

τότε

➤ **Χρησιμοποιήστε τεχνικές όπως η δημιουργία πρωτοτύπων και η ανάλυση του άμεσου περιβάλλοντος των χρηστών**

REQUIREMENTS ANALYSIS - ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΔΙΑΠΡΑΓΜΑΤΕΥΣΗ





REQUIREMENTS ANALYSIS - ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

Ανάλυση απαιτήσεων: Ο προσδιορισμός του κατά πόσον οι απαιτήσεις που έχουν αναδειχθεί μέσω διαβουλεύσεων με τα ενδιαφερόμενα μέρη είναι σαφείς, πλήρεις, συνεπείς και ξεκάθαρες και η επίλυση τυχόν εμφανών συγκρούσεων.

Άσκηση: Εξετάστε την ακόλουθη απαίτηση

“A website to change student’s personal details online. It should be quick and easy to use with a pleasant user interface.”

- Είναι σαφής, πλήρης και ξεκάθαρη;
- Είναι επαληθεύσιμη (verifiable), θα μπορούσατε να σχεδιάσετε και να κατασκευάσετε το σύστημα και να ελέγξετε ότι το σύστημα πληροί αυτή την προδιαγραφή;
- Χρειάζεται να κάνετε περισσότερη συλλογή απαιτήσεων;



REQUIREMENTS ANALYSIS

CLASSIFYING REQUIREMENTS

- Η εξεύρεση απαιτήσεων είναι μόνο η αρχή.
- Μόλις εντοπιστούν οι απαιτήσεις, πρέπει να **αναλυθούν**, να **ταξινομηθούν** και να **ιεραρχηθούν**.
- **Ποιες απαιτήσεις καταγράφουμε;**
 - **Functional - Λειτουργικές**
 - Απαραίτητα χαρακτηριστικά (features) και δυνατότητες
 - **What** does the system need to do - **Τι** πρέπει να κάνει το σύστημα
 - **Τι λειτουργίες πρέπει να υποστηρίζει**
 - **Non-functional – Μη λειτουργικές**
 - Περιορισμοί στον τρόπο λειτουργίας του συστήματος
 - e.g. performance, security, usability, speed etc.
 - **How** does the system need to do it - **Πώς** πρέπει να το κάνει το σύστημα



FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

ΛΕΙΤΟΥΡΓΙΚΕΣ ΚΑΙ ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

FUNCTIONAL REQUIREMENTS - ΛΕΙΤΟΥΡΓΙΚΕΣ

Τι πρέπει να κάνει το σύστημα. Ποια είναι τα απαραίτητα χαρακτηριστικά και οι δυνατότητες που υποστηρίζει το σύστημα;

- Οι ενδιαφερόμενοι/πελάτες/χρήστες του συστήματος συχνά εκφράζουν τις λειτουργικές απαιτήσεις χρησιμοποιώντας φυσική γλώσσα
- Στη συνέχεια, είναι δουλειά του αναλυτή συστήματος να καταγράψει αυτές τις απαιτήσεις με σαφή και ξεκάθαρο τρόπο

Example: Health care system

- Ο χρήστης πρέπει να μπορεί να αναζητήσει τους καταλόγους ραντεβού για όλες τις κλινικές.
- Το σύστημα παράγει κάθε μέρα, για κάθε κλινική, κατάλογο των ασθενών που αναμένεται να παραστούν σε ραντεβού εκείνη την ημέρα.
- Οι εργαζόμενοι πρέπει να μπορούν να κλείνουν ραντεβού για ένα πελάτη μέσω του συστήματος

NON-FUNCTIONAL REQUIREMENTS – ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ



Περιορισμοί (constraints), στόχοι ή μηχανισμοί ελέγχου: ιδιότητες του ίδιου του προϊόντος λογισμικού



Performance and speed requirements

e.g. 20,000 συναλλαγές την μέρα



Security and access requirements

e.g μόνο registered χρήστες να μπορούν να χρησιμοποιήσουν συγκεκριμένες λειτουργίες, να ανταποκρίνεται σε legal απαιτήσεις



Technical constraints (including interoperability, scalability and platform constraints)

e.g. Να μπορεί να τρέξει σε υπάρχον hardware, η να μπορεί να χρησιμοποιήσει ήδη υπάρχοντα δίκτυα



Project constraints

e.g. Να είναι τελειωμένο σε 6 μήνες



Organisational constraints (including cost and time to market)

e.g. Το training να ολοκληρωθεί σε 2 εβδομάδες



Usability and reliability issues

e.g. Να είναι εύκολο στη χρήση και να μπορεί να χρησιμοποιηθεί από άτομα που είναι colour blind

NON-FUNCTIONAL REQUIREMENTS - ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ

Must be objective and testable (use measurable metrics) - Πρέπει να είναι αντικειμενικές και ελέγξιμες (χρήση μετρήσιμων μετρήσεων).

Property	Metric
Performance/ Speed	1. Processed transactions per second 2. Response time to user input 3. Screen Refresh time
Reliability	1. Rate of occurrence of failure 2. Mean time to failure
Availability	Probability of failure on demand
Size	Mbytes Number of ROM chips
Usability	1. Time taken to learn 80% of the facilities 2. Number of errors made by users in given time period
Robustness	Time to restart after system failure Percentage of events causing failure Probability of data corruption on failure
Portability	Number of target systems

FUNCTIONAL VS NON-FUNCTIONAL EXAMPLE

“Το σύστημα διασφαλίζει ότι τα δεδομένα προστατεύονται από μη εξουσιοδοτημένη πρόσβαση.”

Συμβατικά, αυτή θα θεωρούνταν μη λειτουργική απαίτηση, επειδή δεν προσδιορίζει συγκεκριμένη λειτουργικότητα του συστήματος που πρέπει να παρέχεται.

Ωστόσο, θα μπορούσε να είχε καθοριστεί με ελαφρώς περισσότερες λεπτομέρειες ως εξής:

“Το σύστημα πρέπει να περιλαμβάνει διαδικασία εξουσιοδότησης του χρήστη, όπου οι χρήστες πρέπει να ταυτοποιούνται χρησιμοποιώντας όνομα σύνδεσης και κωδικό πρόσβασης. Μόνο οι χρήστες που έχουν εξουσιοδοτηθεί με αυτόν τον τρόπο μπορούν να έχουν πρόσβαση στα δεδομένα του συστήματος.”

Με αυτή τη μορφή, η απαίτηση μοιάζει μάλλον περισσότερο με λειτουργική απαίτηση, καθώς προσδιορίζει μια λειτουργία (σύνδεση χρήστη - login) που πρέπει να ενσωματωθεί στο σύστημα.



BREAKOUT SESSION

IN GROUPS

Προσδιορίστε τουλάχιστον 3-4 λειτουργικές και μη λειτουργικές απαιτήσεις για:

1. A car rental system - Σύστημα ενοικίασης αυτοκινήτων
2. A study abroad system - Ένα σύστημα για σπουδές στο εξωτερικό
3. An ATM System - Ένα σύστημα ATM

Συμβουλή:

- Σκεφτείτε πρώτα ποιοι είναι οι χρήστες και ποιες είναι οι ανάγκες τους. Ποιες λειτουργίες/χαρακτηριστικά πρέπει να υποστηρίζει το σύστημα γι' αυτούς;
- Μη λειτουργικές απαιτήσεις - σκεφτείτε τις απαιτήσεις απόδοσης, τις απαιτήσεις ασφάλειας και άλλους περιορισμούς

PRIORITISATION OF THE REQUIREMENTS – ΙΕΡΑΡΧΗΣΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ

- **Why prioritise – Γιατί να ιεραρχήσουμε τις απαιτήσεις?**

- Δεν υπάρχει αρκετός χρόνος για να υλοποιήσετε τα πάντα
- Δεν υπάρχουν αρκετοί πόροι για να τα κάνετε όλα
 - Έλλειψη χρημάτων ή έλλειψη ανθρώπων (ή και τα δύο)

Το MoSCoW σημαίνει ότι τα σημαντικά πράγματα γίνονται πρώτα

Το MOSCOW είναι μια τεχνική που βοηθά στην κατανόηση και διαχείριση των προτεραιοτήτων

- Οι προτεραιότητες του MoSCoW καθορίζουν τη σειρά υλοποίησης

MOSCOW PRIORITISATION

Must have for requirements that are fundamental to the system - (θεμελιώδεις για το σύστημα)

- **M**inimum **U**sable **S**ubse**T** (Guaranteed)] Ελάχιστο χρησιμοποιήσιμο υποσύνολο (εγγυημένο)]
- Το λογισμικό δεν είναι νόμιμο, δεν είναι ασφαλές και δεν είναι βιώσιμο χωρίς αυτές
- Ρωτήστε "τι θα συμβεί αν αυτή η απαίτηση δεν ικανοποιηθεί;". Εάν η απάντηση είναι "ακυρώστε το έργο - δεν έχει νόημα η υλοποίηση μιας λύσης που δεν ικανοποιεί αυτή την απαίτηση", τότε πρόκειται για μια απαίτηση που πρέπει να έχετε.

Should have for important requirements that would probably be classed as mandatory in a less time-constrained environment

- Σημαντικές αλλά όχι ζωτικής σημασίας
- Μπορεί να είναι επώδυνο να παραλειφθούν, αλλά η λύση εξακολουθεί να είναι βιώσιμη
- Εάν υπάρχει κάποιος τρόπος να παρακαμφθεί, ακόμη και αν πρόκειται για χειροκίνητη και επίπονη λύση, τότε πρόκειται για ένα Should Have ή Could have.

Could have for requirements that can more easily be left out of the increment under development – μπορούν να παραληφθούν από αυτό το sprint

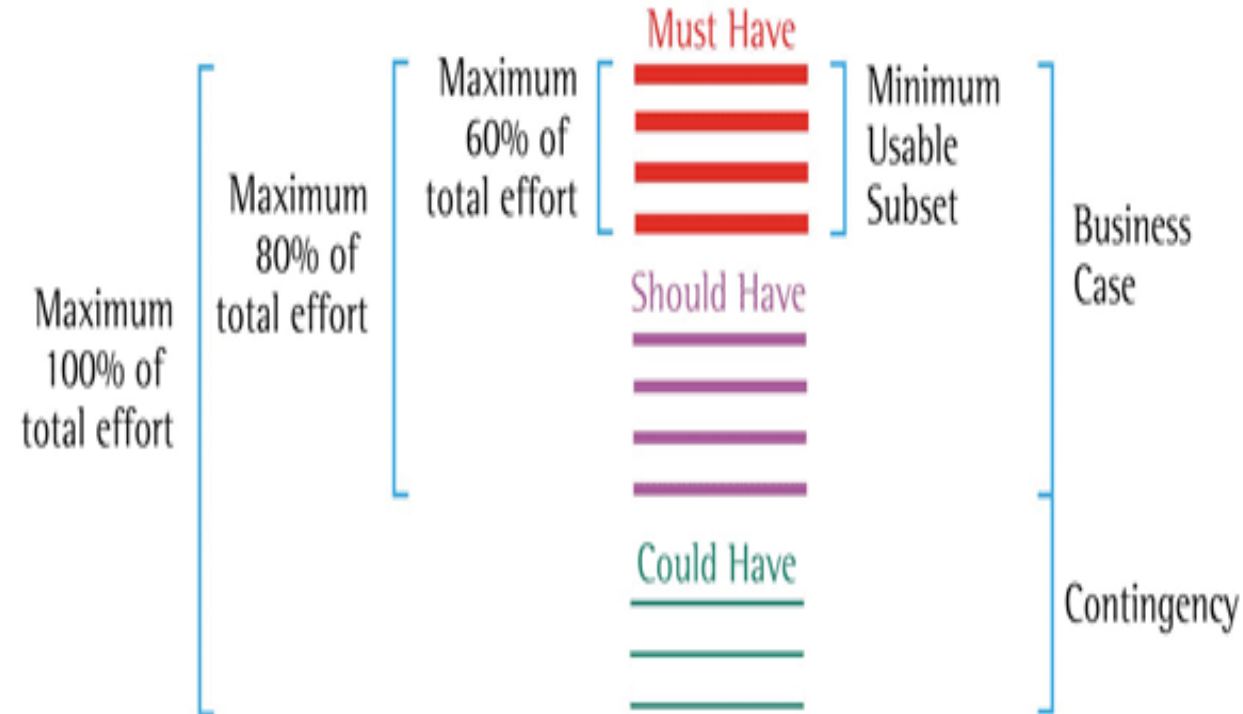
Επιθυμητές αλλά λιγότερο σημαντικές

Μικρότερος αντίκτυπος αν παραλειφθεί (σε σύγκριση με το Should)

Won't have this time round

Απαιτήσεις που η ομάδα έργου έχει συμφωνήσει ότι δεν θα παραδοθούν αυτή την στιγμή. Μπορούν να περιμένουν αργότερα

MOSCOW RULES AND 80:20 RULE



- Τα Musts και Shoulds συχνά προσφέρουν το 80% του συνολικού επιχειρηματικού οφέλους
- Το υπόλοιπο 20% είναι για απρόβλεπτες ανάγκες - αν δεν μπορούν να υλοποιηθούν, τουλάχιστον δεν είναι Must Haves or Should haves.

PRIORITISATION OF THE REQUIREMENTS

Σε ένα παραδοσιακό έργο, όλες οι απαιτήσεις αντιμετωπίζονται ως "Must Haves", καθώς τίθεται εξ αρχής η προσδοκία ότι όλα θα παραδοθούν και ότι τυπικά ο χρόνος (η ημερομηνία λήξης) θα γλιστρήσει αν παρουσιαστούν προβλήματα.

Τα έργα που χρησιμοποιούν πιο σύγχρονες προσεγγίσεις, π.χ. ευέλικτες μέθοδοι, έχουν μια πολύ διαφορετική προσέγγιση: Καθορίζουν χρόνο, κόστος και ποιότητα και διαπραγματεύονται τα χαρακτηριστικά που πρέπει να παραδοθούν. Ως εκ τούτου, προσδιορίζονται οι λειτουργικότητες που πρέπει να παραδοθούν, διασφαλίζοντας ότι αυτές θα παραδοθούν εντός του συμφωνημένου κόστους και χρονοδιαγράμματος.



BREAKOUT SESSION

GROUP IN CLASS ACTIVITY

Η EventsBright είναι μια αδελφή εταιρεία της MusicFun που πωλεί εισιτήρια για θέατρα, συναυλίες και άλλες εκδηλώσεις. Η εταιρεία ζήτησε από την ομάδα σας να αναπτύξει ένα σύστημα ηλεκτρονικής πώλησης εισιτηρίων και σας έχει προμηθεύσει έναν αρχικό κατάλογο απαιτήσεων.

1. Προσδιορίστε ποιες από τις απαιτήσεις που περιλαμβάνονται στον παρακάτω κατάλογο είναι Λειτουργικές και ποιες μη Λειτουργικές και εξηγήστε τους λόγους.
2. Ιεραρχήστε τις λειτουργικές απαιτήσεις χρησιμοποιώντας το MOSCOW

GROUP IN CLASS ACTIVITY

Initial List of Requirements

1. Users must be able to browse for events by name or date or venue.
1. Searching for events by keywords should also be possible.
1. User will select ticket price from those offered and also specify the number of tickets they require.
1. The system should display availability of the tickets within five seconds.
1. Users must be able to choose their seats and purchase tickets 24/7.
1. Only registered customers can buy tickets.
1. Daily reports of ticket sales must be generated. These must be retained for two years.
1. All data is to be backed up nightly.
1. It should not be possible to double-book tickets (i.e. buy a ticket that has already been sold).
1. The system should show ticket sales over a week.
1. The pages of the online system should load in less than 2 seconds for 99% of customers, considering their location and bandwidth.

AGILE METHODS AND REQUIREMENTS – ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΙ ΚΑΙ ΑΠΑΙΤΗΣΕΙΣ



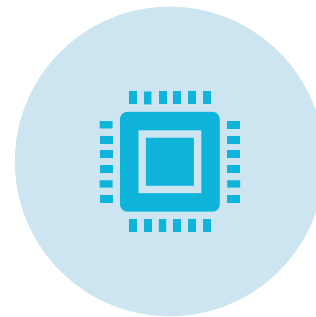
Πολλές ευέλικτες μέθοδοι υποστηρίζουν ότι η παραγωγή λεπτομερών απαιτήσεων του συστήματος είναι χάσιμο χρόνου, καθώς οι απαιτήσεις αλλάζουν τόσο γρήγορα.



Ως εκ τούτου, το έγγραφο απαιτήσεων είναι πάντα ξεπερασμένο (out of date)



Οι ευέλικτες μέθοδοι χρησιμοποιούν συνήθως αυξητική μηχανική απαιτήσεων **(incremental requirements engineering)** και εκφράζουν τις απαιτήσεις ως "ιστορίες χρήστη".



Αυτό είναι πρακτικό για επιχειρησιακά συστήματα, αλλά προβληματικό για συστήματα που απαιτούν ανάλυση πριν από την παράδοση (π.χ. κρίσιμα συστήματα) ή συστήματα που αναπτύσσονται από διάφορες ομάδες.

AGILE USER STORIES – ΙΣΤΟΡΙΕΣ ΧΡΗΣΤΩΝ

USER STORIES

- Οι ιστορίες χρήστη είναι το εργαλείο που χρησιμοποιείται για την περιγραφή των απαιτήσεων που καταχωρούνται στο Product Backlog
 - Features, enhancements/bug fixes, non-functional requirements
 - Είναι απαιτήσεις που εκφράζονται από την οπτική γωνία ενός στόχου του τελικού χρήστη.
- Γρήγορος και απλός τρόπος περιγραφής του τρόπου με τον οποίο ένας χρήστης θα χρησιμοποιήσει το λογισμικό.

As a <type of user>, I want <some goal> so that <some reason>

Ως <τύπος χρήστη>, θέλω <κάποιο στόχο> έτσι ώστε <κάποιος λόγος>

- Μεταξύ μίας και τεσσάρων προτάσεων
- Περιγράψτε το ΠΟΙΟΣ, ΤΙ, ΓΙΑΤΙ
 - Περιγράψτε ΚΡΙΤΗΡΙΑ ΑΠΟΔΟΧΗΣ ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ - **HIGH LEVEL ACCEPTANCE CRITERIA**
- Should fit on the front of a 3x5 index card (**Called Story cards**)

USER STORY TEMPLATE

Who (user role)

- Is this a customer, employee, admin, etc.?

What (goal)

- What functionality must be achieved/developed?

Why (reason)

- Why does user want to accomplish this goal?

☐	Story ID:	Story Title:
User Story:		Importance:
As a: <role>		
I want: <some goal>		
So that: <some reason>		Estimate:
Acceptance Criteria		Type:
And I know I am done when:		☐ Search
		☐ Workflow
		☐ Manage Data
		☐ Payment
		☐ Report/ View

Οι Ιστορίες Χρήστη συχνά γράφονται σε κάρτες ή σημειώσεις που κολλάνε, αποθηκεύονται σε μια κουτί, και τοποθετούνται σε τοίχους ή τραπέζια για να διευκολυνθεί ο σχεδιασμός και η συζήτηση. Είναι σημαντικό τα user stories να είναι σύντομα και συνοπτικά - ο στόχος είναι να έχουμε μια συζήτηση για τις απαιτήσεις αντί να θεωρούμε τα user stories ως ένα συμβόλαιο.

USER STORIES EXAMPLE

ID	As a...	I want to be able to...	So that...
1	Administrator	see a list of all members and visitors	I can monitor site visits
2	Administrator	add new categories	I can allow members to create engaging content
3	Administrator	add new security groups	security levels are appropriate
4	Administrator	add new keywords	content is easy to group and search for
5	Administrator	delete comments	offensive content is removed
6	Administrator	block entries	competitors and offenders cannot submit content
7	Administrator	change site branding	the site is future-proofed in case brand changes
8	Member	change my password	I can keep secure
9	Member	update my contact details	I can be contacted by Administrators
10	Member	update my email preferences	I'm not bombarded with junk email
11	Member	share content to social networks	I can promote what I find interesting
12	Visitor	create an account	I can benefit from member discounts
13	Visitor	login	I can post new entries
14	Visitor	add comments	I can have a say
15	Visitor	suggest improvements	I can contribute to the site usability
16	Visitor	contact the Administrators	I can directly submit a query
17	Visitor	follow a member's updates	I'm informed of updates from members I find interesting
18	Visitor	view a member's profile	I can know more about a member
19	Administrator	generate incoming traffic report	I can understand where traffic is coming from



BREAKOUT SESSION

IN GROUPS - CASE STUDY

Ένα πρακτορείο αναπτύσσει μια ταξιδιωτική πλατφόρμα. Ο χρήστης της ταξιδιωτικής πλατφόρμας θα πρέπει να μπορεί να κάνει κρατήσεις πτήσεων, δωματίων ξενοδοχείων, αλλά και ενοικιαζόμενων αυτοκινήτων στην ίδια πλατφόρμα.

Σε ομάδες Προσδιορίστε 3 ιστορίες χρηστών για το ακόλουθο παράδειγμα.

USER STORIES ACCEPTANCE CRITERIA



Είναι σαν ένα σενάριο για το πώς ένας χρήστης θα χρησιμοποιούσε ένα συγκεκριμένο user story του συστήματος.

Υπάρχουν πεδία εισόδου όπου οι χρήστες πρέπει να προσθέσουν δεδομένα?
κουμπιά που πρέπει να καταχωρηθούν?
μηνύματα σφάλματος κλπ.



Αναφέρονται επίσης ως "Συνθήκες Ικανοποίησης" για την Ιστορία Χρήστη, γραμμένες στο πίσω μέρος της κάρτας. Είναι σαν κουτάκια που πρέπει να γίνουν ticked για να μπορεί να εγκριθεί το user story από τον Product Owner



Υψηλού επιπέδου λίστα ελέγχου για το τι χρειάζεται κάθε ιστορία χρήστη να επιτύχει (όταν είναι πλήρως υλοποιημένη και δοκιμασμένη).

Some acceptance criteria describe:

Την αναμενόμενη λειτουργικότητα. Περιορισμούς που πρέπει να υπάρχουν.

Πώς το feature πρέπει να συνδέεται με άλλα δεδομένα και χαρακτηριστικά. Ένα αναμενόμενο επίπεδο απόδοσης.

USER STORIES ACCEPTANCE CRITERIA EXAMPLE

As a user, I can
cancel a reservation.

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

- ☐ Verify that a premium member can cancel the same day without a fee.
- ☐ Verify that a non-premium member is charged 10% for a same-day cancellation.
- ☐ Verify that an email confirmation is sent.
- ☐ Verify that the hotel is notified of any cancellation.

Concisely written criteria help development teams avoid ambiguity about a client's demands and prevent miscommunication.

USER STORIES ACCEPTANCE CRITERIA EXAMPLE

User story: As a new user, I would like to login in order to be able to use the system.

Some acceptance criteria for login user story for example could be:

- *On the sign in page there are 2 input fields for username and password.*
- *If fields are completed with correct authentication credentials and the sign in button is clicked then the system signs users in.*
- *If credentials are wrong then...*
- *Successfully logged in users should be navigated to home page with a welcome message*
- *Welcome message should display in the upper right corner of the home page*
- *Home page should load within 1 seconds*

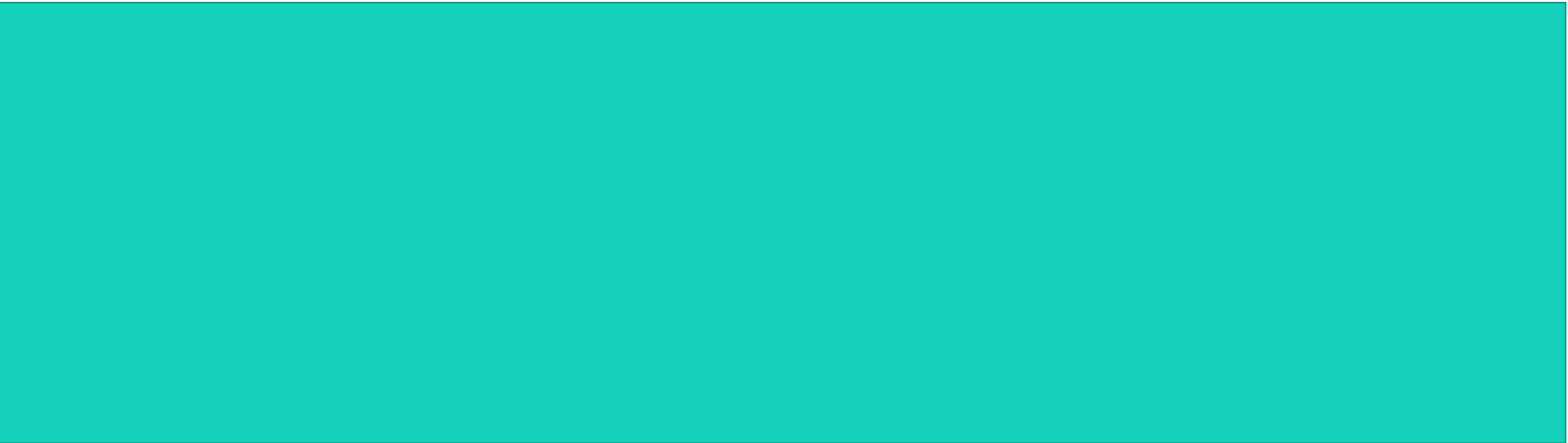


BREAKOUT SESSION

IN GROUPS

Ένα πρακτορείο αναπτύσσει μια ταξιδιωτική πλατφόρμα. Ο χρήστης της ταξιδιωτικής πλατφόρμας θα πρέπει να μπορεί να κάνει κρατήσεις πτήσεων, δωματίων ξενοδοχείων, αλλά και ενοικιαζόμενων αυτοκινήτων στην ίδια πλατφόρμα.

Σε ομάδες δημιουργήστε τις συνθήκες ικανοποίησης για 2 από τα user stories που αναφέρατε πιο πάνω



NON FUNCTIONAL REQUIREMENTS AS USER STORIES

Non Functional Requirements

- Δεν αφορούν συγκεκριμένη λειτουργικότητα/χαρακτηριστικό
- Αφορούν ένα χαρακτηριστικό ή μια ιδιότητα του συστήματος ή τους περιορισμούς που θέτουμε στο σύστημα.
- π.χ reliability, availability, portability, scalability, usability, maintainability, security, performance, robustness etc.

As a customer, I want to be able to run your product on all versions of Windows from Windows 95 on.

As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain.

As a user, I want the site to be available 99.999 percent of the time I try to access it, so that I don't get frustrated and find another site to use.

NON FUNCTIONAL REQUIREMENTS AS USER STORIES EXAMPLE

Performance Constraint

As Mary, I want to receive a response to any input in less than one second.

Acceptance Criteria

- Measure the time between clicking a button or hitting a key to the response being displayed.
- 50 users active on the website; no other booking is taking place.

THE SIGNIFICANCE OF “DEFINITION OF DONE”

Οι "Συνθήκες Ικανοποίησης", ή τα acceptance test criteria για ένα user story είναι πολύ σημαντικά για την ομάδα ανάπτυξης.

- Παρέχει στην ομάδα ανάπτυξης έναν συγκεκριμένο **Definition of done**

Όλοι πρέπει να κατανοούν τι σημαίνει “DONE”

- Το **Definition of done** είναι μια συμφωνία μεταξύ του Product owner και του development team το οποίο περιγράφει τι ακριβώς σημαίνει για αυτούς όταν λέμε ότι ένα user story είναι ολοκληρωμένο, *its is done*.

Για να χαρακτηριστεί η ιστορία χρήστη ως "Done", πρέπει να έχει γίνει όλη η εργασία που απαιτείται για την κατανόησή της, την κατασκευή της, τη δοκιμή και την εφαρμογή της.

- **The user story is “Done” όταν ο χρήστης**
 - μπορεί να χρησιμοποιήσει το λογισμικό
 - εκτελέσει κάθε μία από τις συνθήκες που ορίζονται acceptance test criteria με τον ίδιο ακριβώς τρόπο που θα το έκανε στο τέλος του Sprint και κατά την διάρκεια του Sprint Review.

WELL CONSTRUCTED USER STORIES SHOULD BE...

Independent

Οι ιστορίες θα πρέπει να είναι όσο το δυνατόν πιο ανεξάρτητες από άλλες ιστορίες, ώστε να μπορούν να υλοποιούνται ανεξάρτητα. Αν οι ιστορίες εξαρτώνται στενά μεταξύ τους, εξετάστε το ενδεχόμενο να τις συνδυάσετε σε μία ενιαία ιστορία χρήστη.

Negotiable

Οι ιστορίες δεν είναι συμβόλαια. Αποτελούν "σημεία αναφοράς" για λειτουργίες που η ομάδα θα συζητήσει και θα αποσαφηνίσει κοντά στον χρόνο ανάπτυξης.

Valuable

Οι ιστορίες πρέπει να αντιπροσωπεύουν λειτουργίες που παρέχουν σαφή επιχειρηματική αξία στον χρήστη/ιδιοκτήτη της λύσης και να είναι γραμμένες σε κατάλληλη γλώσσα. Πρέπει να είναι λειτουργίες (features) του συστήματος και όχι απλές εργασίες.

Estimable

Οι ιστορίες πρέπει να είναι αρκετά σαφείς ώστε να εκτιμηθούν (πόση προσπάθεια χρειάζεται για την υλοποίηση τους), χωρίς να είναι υπερβολικά λεπτομερείς.

Small

Οι ιστορίες πρέπει να είναι αρκετά μικρές ώστε να μπορούν να εκτιμηθούν. Οι μεγαλύτερες ιστορίες ("Epic") θα πρέπει να χωρίζονται σε μικρότερες ιστορίες χρήστη καθώς το έργο προχωρά. Μετά τον διαχωρισμό, οι ιστορίες πρέπει να εξακολουθούν να ακολουθούν τα κριτήρια INVEST

Testable

Οι ιστορίες πρέπει να είναι διατυπωμένες με σαφήνεια και ακρίβεια, ώστε να μπορούν να ελεγχθούν μέσω δοκιμών.

ΠΩΣ ΤΑ ΠΗΓΑΜΕ ΑΥΤΗ ΤΗΝ ΕΒΔΟΜΑΔΑ? ΤΙ ΜΑΘΑΜΕ?



Reply at:

PollEv.com/avgoustakyri977

- | | |
|---------------------------------------|--|
| 1 Go to PollEv.com | 1 Text AVGOUSTAKYRI977 to 22333 |
| 2 Enter AVGOUSTAKYRI977 | 2 Text in your message |
| 3 Respond to activity | |

WRAP UP



Examined the requirements phase of the life-cycle.



Moscow prioritisation



Requirements capturing: What tools and for which situations.



Agile and User stories, user story template



Requirements analysis: functional and non-functional.



Acceptance test criteria and Definition of Done

READINGS/REFERENCES

1. Kendall, K. E., και Kendall, J. E. Ανάλυση & Σχεδίαση Συστημάτων. 10η έκδοση, Εκδόσεις Μ. Γκιούρδας, Κεφάλαιο 4 και 5, 2023.
2. Βεσκούκης 2015 – Κεφάλαιο 4
3. Pressman, R.S. (2020). Software Engineering: A practitioner's approach, 9th Edition, Mc-Graw Hill – Chapter 7
4. Kung, D.C (2024) Software Engineering, 2nd Edition, McGraw Hill – Chapter 4