

**Τίτλος έργου:** Διασυνδεδεμένες Έξυπνες Πόλεις για την Ελλάδα 2.0

**Κωδικός:** TAEDR-0536642

**MIS (ΟΠΣ):** 5149305

**Παραδοτέο:** Π1.1

**Τίτλος:** Απαιτήσεις πλατφόρμας και Μεθοδολογία ανάπτυξης

## Συμμετέχοντες

Φορέας
Ίδρυμα Τεχνολογίας και Έρευνας (ΙΤΕ)
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών (ΕΚΠΑ)
Εθνικό Μετσόβιο Πολυτεχνείο (ΕΜΠ)
Πανεπιστήμιο Θεσσαλίας (ΠΘ)
Πανεπιστήμιο Δυτικής Μακεδονίας (ΠΔΜ)
Πανεπιστήμιο Πατρών (ΠΠ)

## Περιεχόμενα

Περιεχόμενα.....	3
Λίστα Εικόνων.....	4
Λίστα Πινάκων .....	5
1. Εισαγωγή .....	6
1.1 Επισκόπηση του έργου «Διασυνδεδεμένες Έξυπνες Πόλεις για την Ελλάδα 2.0» .....	6
1.2 Σκοπός και πεδίο εφαρμογής του παραδοτέου .....	6
1.3 Δομή του παραδοτέου .....	7
2. Μεθοδολογία Προσδιορισμού των Απαιτήσεων.....	8
2.1 Λειτουργικές και Μη-Λειτουργικές Απαιτήσεις .....	8
2.2 Μοντέλο Διαδικασίας Ανάλυσης Απαιτήσεων .....	10
2.3 Η μέθοδος ιεράρχησης MoSCoW .....	11
2.4 Ορολογία.....	14
3. Λειτουργικές Απαιτήσεις Πλατφόρμας .....	16
4. Ευέλικτη Μεθοδολογία Ανάπτυξης .....	35
4.1 Εισαγωγή .....	35
4.2 Σύνοψη Ευέλικτων (Agile) μεθοδολογιών και διαδικασιών .....	36
4.3 Διαδικασία Ευέλικτης Ανάπτυξης (Agile Development Process) .....	38
4.3.1 Οφέλη της Μεθοδολογίας Ευέλικτης Ανάπτυξης.....	41
4.3.2 Ευελιξία (Flexibility).....	42
4.3.3 Ταχύτητα στην αγορά (Speed to Market) .....	42
4.3.4 Συνεργασία (Collaboration) .....	43
4.3.5 Ποιότητα (Quality).....	43
4.4 Διαδικασία Ευέλικτης Ανάπτυξης Γραφικών Διεπαφών Χρήστη (Agile UX).....	44
4.4.1 Εισαγωγή.....	44
4.4.2 Βασικές Αρχές του Agile .....	44
4.4.3 Διαδικασία Agile UX .....	44
4.4.4 Σύνοψη .....	45
4.5 Εφαρμογή των Μεθοδολογιών Ευέλικτης Ανάπτυξης στο πλαίσιο του έργου .....	46
Σύνοψη .....	50
Αναφορές.....	51

## Λίστα Εικόνων

Εικόνα 1: Αναμενόμενοι αναγνώστες των διαφορετικών τύπων απαιτήσεων .....	9
Εικόνα 2: Μοντέλο ανάλυσης απαιτήσεων.....	11
Εικόνα 3: Προτεραιότητα απαιτήσεων .....	16
Εικόνα 4: Απαιτήσεις ανά κατηγορία .....	18
Εικόνα 5: Μέθοδος Scrum.....	37
Εικόνα 6:Τυπικός πίνακας Kanban.....	38
Εικόνα 7: Κύκλος ζωής Ευέλικτης Ανάπτυξης.....	39

## Λίστα Πινάκων

Πίνακας 1: Κατηγορίες Μη-Λειτουργικών Απαιτήσεων.....	9
Πίνακας 2: Πεδία υποδείγματος συλλογής απαιτήσεων.....	14
Πίνακας 3: Απαιτήσεις ανά τύπο.....	17
Πίνακας 4: Βασικές αρχές της μεθοδολογίας Ευέλικτης Ανάπτυξης.....	36
Πίνακας 5: Τα Πλεονεκτήματα της Διαδικασίας Agile UX.....	46

## 1. Εισαγωγή

### 1.1 Επισκόπηση του έργου «Διασυνδεδεμένες Έξυπνες Πόλεις για την Ελλάδα 2.0»

Ο βασικός στόχος του έργου είναι η ανάπτυξη μίας καθολικής πλατφόρμας για τη διαχείριση Έξυπνων Πόλεων και η δημιουργία ενός κεντρικού σημείου αναφοράς οικοσυστημάτων του Διαδικτύου των Πραγμάτων (ΔτΠ).

Συγκεκριμένα, από τεχνικής σκοπιάς έχει προχωρήσει η ανάπτυξη μιας πλατφόρμας διαλειτουργικότητας, υλοποιημένη και κατανομημένη σε πολλαπλά επίπεδα, η οποία εν τέλει θα συγκεντρώσει τεχνολογίες και υπηρεσίες που προέρχονται από το σύνολο των υπαρχόντων κάθετων υλοποιήσεων που βρίσκονται ήδη εν λειτουργία στις μεγαλύτερες πόλεις της Ελλάδας που είχαν ήδη εκδηλώσει το ενδιαφέρον είτε κατά την περίοδο συγγραφής της πρότασης (Τρίκαλα, Πάτρα, Ηράκλειο, Θεσσαλονίκη), ή το έκαναν κατά τη διάρκεια εκτέλεσης του έργου (Αθήνα, Λάρισα, Κοζάνη).

Η πλατφόρμα, μέσω του πλήθους των παρεχόμενων ενοποιημένων υπηρεσιών που θα επιτρέπουν την απευθείας άντληση δεδομένων ετερογενών συστημάτων, αλλά και την οπτικοποίηση του συνόλου της πληροφορίας, θα καθιστά πλέον εφικτή την ανάπτυξη νέων καινοτόμων υπηρεσιών και προϊόντων προστιθέμενης αξίας στο πλαίσιο του προγράμματος Ελλάδα 2.0 και συγκεκριμένα στον τομέα των Έξυπνων Πόλεων.

Με αυτόν τον τρόπο θα τοποθετηθεί ένας ακρογωνιαίος λίθος για την ουσιαστική δικτύωση και ταχύτερη αξιοποίηση των υφιστάμενων υποδομών, για τη δημιουργία σύνθετων προϊόντων και υπηρεσιών με στόχους:

- τη μεταφορά και αξιοποίηση υφιστάμενης τεχνογνωσίας
- τη δημιουργία καινοτόμων εφαρμογών και υπηρεσιών.
- την ενδυνάμωση της επιχειρηματικότητας και ανταγωνιστικότητας.
- την προώθηση της συμμετοχικότητας στη λειτουργία της τοπικής αυτοδιοίκησης
- την προώθηση της σύμπραξης ακαδημαϊκών και ερευνητικών φορέων με τη βιομηχανία.

### 1.2 Σκοπός και πεδίο εφαρμογής του παραδοτέου

Το παρόν παραδοτέο καταγράφει και αναλύει τις απαιτήσεις λειτουργίας και διαλειτουργικότητας που πρέπει να πληροί η υπό ανάπτυξη μετα-πλατφόρμα, προκειμένου να επιτευχθούν οι στόχοι του έργου. Αναμένεται να αποτελέσει ένα πολύτιμο εγχειρίδιο για την ελληνική επιχειρηματική κοινότητα, προσφέροντας τεκμηριωμένη καθοδήγηση στη λήψη αποφάσεων για την ανάπτυξη νέων προϊόντων και υπηρεσιών. Συγκεκριμένα, μπορεί να συμβάλει στη δημιουργία καινοτόμων προϊόντων που δεν

υφίστανται σήμερα στην ελληνική αγορά, καθώς και στην υλοποίηση προηγμένων υπηρεσιών του ΔτΠ, διασφαλίζοντας μέγιστη διαλειτουργικότητα από τον σχεδιασμό τους (by design).

Επιπλέον, το παραδοτέο τεκμηριώνει τη μεθοδολογία ανάπτυξης του ΔτΠ, η οποία έχει επιλεγεί με βάση τις βέλτιστες πρακτικές που ακολουθούν τα μεγαλύτερα ερευνητικά και ακαδημαϊκά ιδρύματα της χώρας. Η προσέγγιση αυτή διασφαλίζει την υιοθέτηση σύγχρονων τεχνολογικών και επιστημονικών προσεγγίσεων, συμβάλλοντας στην αποτελεσματική και βιώσιμη ανάπτυξη της μεταπλατφόρμας.

### 1.3 Δομή του παραδοτέου

Το παραδοτέο ξεκινά με μια εισαγωγή που περιγράφει τον σκοπό, το πεδίο εφαρμογής και τη δομή του. Στη συνέχεια, παρουσιάζεται η μεθοδολογία προσδιορισμού των απαιτήσεων, η οποία περιλαμβάνει λειτουργικές και μη λειτουργικές απαιτήσεις, το μοντέλο ανάλυσης απαιτήσεων, τη μέθοδο ιεράρχησης MoSCoW και σχετική ορολογία. Έπειτα, αναλύονται οι λειτουργικές απαιτήσεις της πλατφόρμας, ακολουθούμενες από την παρουσίαση μιας ευέλικτης μεθοδολογίας ανάπτυξης. Σε αυτήν την ενότητα περιγράφονται οι βασικές αρχές και τα οφέλη των Agile διαδικασιών, όπως η ευελιξία, η ταχύτητα στην αγορά, η συνεργασία και η ποιότητα, καθώς και η εφαρμογή της Agile προσέγγισης στον σχεδιασμό γραφικών διεπαφών χρήστη (Agile UX). Το έγγραφο ολοκληρώνεται με τη σύνοψη των ευρημάτων και τις αναφορές. Το παραδοτέο αποτελείται από τις παρακάτω ενότητες:

## 2. Μεθοδολογία Προσδιορισμού των Απαιτήσεων

Η ανάλυση απαιτήσεων αποτελεί μία ιδιαίτερα σημαντική διαδικασία στη μηχανική λογισμικού, αφού καθορίζει τις προϋποθέσεις που πρέπει να πληρούνται ώστε να θεωρηθεί επιτυχής η ανάπτυξη ενός συστήματος. Στο συγκεκριμένο έργο υιοθετούμε την ευέλικτη μεθοδολογία ανάπτυξης λογισμικού (Agile Software development [1]), το οποίο (σε επίπεδο συλλογής και καθορισμού απαιτήσεων) σημαίνει ότι η λίστα των απαιτήσεων μπορεί να αναπροσαρμοστεί σύμφωνα με τις ανάγκες, κατά τη διάρκεια του αρχιτεκτονικού σχεδιασμού του συστήματος.

### 2.1 Λειτουργικές και Μη-Λειτουργικές Απαιτήσεις

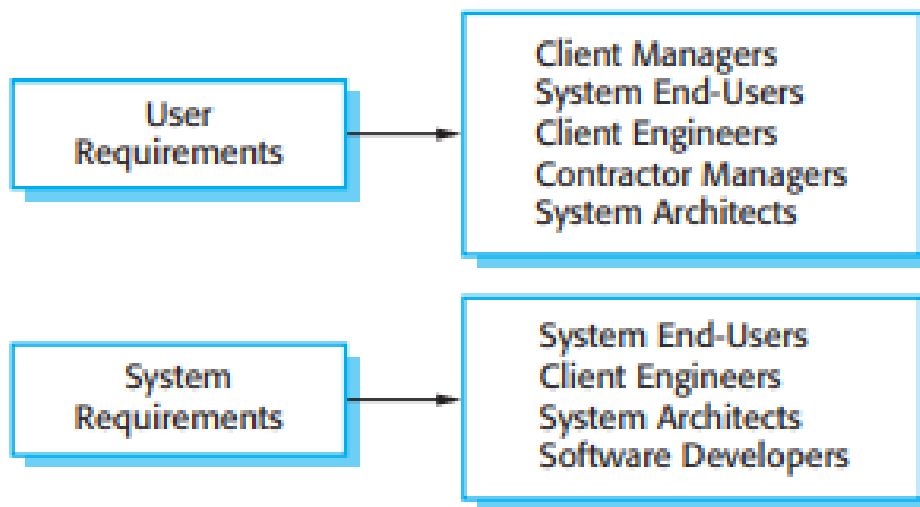
Οι παρακάτω ορισμοί υιοθετούνται:

**Λειτουργικές Απαιτήσεις (ΛΑ):** “Πρόκειται για δηλώσεις (statements) των υπηρεσιών που πρέπει να παρέχει το σύστημα, του τρόπου με τον οποίο αυτό πρέπει να αντιδρά σε συγκεκριμένες εισόδους ή να συμπεριφέρεται σε συγκεκριμένες καταστάσεις. Σε ορισμένες περιπτώσεις, οι λειτουργικές απαιτήσεις μπορεί επίσης να δηλώνουν ρητά τι δεν πρέπει να κάνει το σύστημα” [5].

**Μη-Λειτουργικές Απαιτήσεις (ΜΛΑ):** "Πρόκειται για περιορισμούς στις υπηρεσίες ή τις λειτουργίες που προσφέρει το σύστημα. Περιλαμβάνουν χρονικούς περιορισμούς, περιορισμούς στη διαδικασία ανάπτυξης, περιορισμούς που επιβάλλονται από πρότυπα (standards), και συχνά αφορούν το σύστημα ως σύνολο και όχι μεμονωμένα χαρακτηριστικά ή υπηρεσίες του, π.χ.: απόδοση, ασφάλεια ή διαθεσιμότητα" [5].

#### Διαφοροποίηση μεταξύ των διαφορετικών τύπων απαιτήσεων:

Διάκριση μεταξύ διαφορετικών τύπων απαιτήσεων: "Στην πραγματικότητα, η διάκριση μεταξύ διαφορετικών τύπων απαιτήσεων δεν είναι τόσο ξεκάθαρη όσο υποδηλώνουν αυτοί οι απλοί ορισμοί. Μια απαίτηση χρήστη που αφορά την ασφάλεια, όπως για παράδειγμα μια δήλωση που περιορίζει την πρόσβαση σε εξουσιοδοτημένους χρήστες, μπορεί να φαίνεται ότι είναι μια μη-λειτουργική απαίτηση. Ωστόσο, όταν εμπλουτιστεί με περισσότερες λεπτομέρειες, η απαίτηση αυτή μπορεί να δημιουργήσει άλλες απαιτήσεις που είναι λειτουργικές, όπως η ανάγκη να ενσωματώσης στο σύστημα μηχανισμών ελέγχου ταυτότητας του χρήστη. Αυτό δείχνει ότι οι απαιτήσεις δεν είναι ανεξάρτητες και ότι μια απαίτηση συχνά δημιουργεί ή περιορίζει άλλες απαιτήσεις. Συνεπώς, οι απαιτήσεις του συστήματος δεν προσδιορίζουν μόνο τις υπηρεσίες ή τα χαρακτηριστικά του συστήματος που απαιτούνται-προσδιορίζουν επίσης την απαραίτητη λειτουργικότητα για να διασφαλιστεί ότι αυτές οι υπηρεσίες/χαρακτηριστικά παρέχονται σωστά." [5].



Εικόνα 1: Αναμενόμενοι αναγνώστες των διαφορετικών τύπων απαιτήσεων

Οι ανάγκες και οι απαιτήσεις των χρηστών προσθέτουν αξία σε ένα προϊόν, μια υπηρεσία ή ένα σύστημα. Το γεγονός ότι λαμβάνονται υπόψη στις φάσεις του σχεδιασμού και της υλοποίησης αποσκοπεί στο να αντικατοπτρίζει το τελικό αποτέλεσμα τις προτιμήσεις, τους στόχους, τους περιορισμούς του χρήστη ή να παρέχει λύση σε συγκεκριμένο πρόβλημα, αυξάνοντας την τελική αποδοχή του από τους χρήστες. Καθώς οι ανάγκες και οι απαιτήσεις των χρηστών αποτελούν τη βάση για το σχεδιασμό των λύσεων του έργου που ανταποκρίνονται στις προσδοκίες τους και παρέχουν το επιθυμητό αποτέλεσμα του συστήματος, η κατανόηση των διαφορετικών τύπων απαιτήσεων των χρηστών και η ομαδοποίησή τους σε διαφορετικές κατηγορίες που έχουν κοινά χαρακτηριστικά, είναι απαραίτητη για την καλύτερη καταγραφή και αντιμετώπιση των συγκεκριμένων αναγκών.

Οι μη-λειτουργικές απαιτήσεις αφορούν ποιοτικές ιδιότητες που δεν αναφέρονται σε ένα συγκεκριμένο χαρακτηριστικό του συστήματος, αλλά σε χαρακτηριστικά της συνολικής λειτουργίας του (Πίνακας 1). Για τη διευκόλυνση της ανάδειξής τους και τη διατήρηση της συνέπειας, έχουν προσδιοριστεί συγκεκριμένες κατηγορίες αυτών.

Πίνακας 1: Κατηγορίες Μη-Λειτουργικών Απαιτήσεων

Κατηγορία ΜΛΑ	Περιγραφή
<b>Συμβατότητα</b>	Ιδιότητες που περιγράφουν το βαθμό στον οποίο το λογισμικό μπορεί να ανταλλάσσει πληροφορίες με άλλα συστήματα. Περιλαμβάνει τις έννοιες της <b>συνύπαρξης (co-existence)</b> και της <b>διαλειτουργικότητας (interoperability)</b> .

<b>Δυνατότητα Συντήρησης</b>	Ιδιότητες που περιγράφουν την ικανότητα του λογισμικού να αλλάζει, να βελτιώνεται και να υποστηρίζεται με την πάροδο του χρόνου με αρκετή ευκολία. Περιλαμβάνει έννοιες όπως: <b>αρθρωτός σχεδιασμός, δυνατότητα επαναχρησιμοποίησης, δυνατότητας τροποποίησης</b> , κ.λπ.
<b>Απόδοση</b>	Ιδιότητες που περιγράφουν τη σχέση μεταξύ του επιπέδου απόδοσης του λογισμικού και του αριθμού των πόρων που χρησιμοποιούνται, υπό καθορισμένες συνθήκες. Περιλαμβάνει τις έννοιες της <b>χωρητικότητας (capacity), της χρήσης των πόρων και της επεκτασιμότητας</b> .
<b>Φορητότητα</b>	Χαρακτηριστικά που περιγράφουν την ικανότητα του λογισμικού να μεταφέρεται από ένα περιβάλλον σε ένα άλλο. Περιλαμβάνει τις έννοιες της <b>προσαρμοστικότητας, της δυνατότητας αντικατάστασης</b> , κ.λπ.
<b>Αξιοπιστία</b>	Χαρακτηριστικά που περιγράφουν την ικανότητα του λογισμικού να διατηρεί το επίπεδο απόδοσής του υπό καθορισμένες συνθήκες για μια καθορισμένη χρονική περίοδο. Περιλαμβάνει τις έννοιες της <b>διαθεσιμότητας, της δυνατότητας ανάκτησης, της ανοχής σε σφάλματα</b> κ.λπ.
<b>Ασφάλεια</b>	Ιδιότητες που περιγράφουν την προστασία του λογισμικού από επιθέσεις. Περιλαμβάνει τις έννοιες της <b>εμπιστευτικότητας, της ακεραιότητας, της αυθεντικότητας, της λογοδοσίας</b> , κ.λπ.
<b>Ευχρηστία</b>	Ιδιότητες που περιγράφουν την προσπάθεια που απαιτείται από τους χρήστες για τη χρήση του λογισμικού. Περιλαμβάνει έννοιες όπως: <b>προσβασιμότητα, της προστασίας από τα λάθη των χρηστών, UX, αισθητική του UI</b> κ.λπ.
<b>Περιορισμοί υλικού και λογισμικού Constraints</b>	Περιορισμοί υλικού και λογισμικού λόγω ενσωμάτωσης υφιστάμενων στοιχείων (assets).

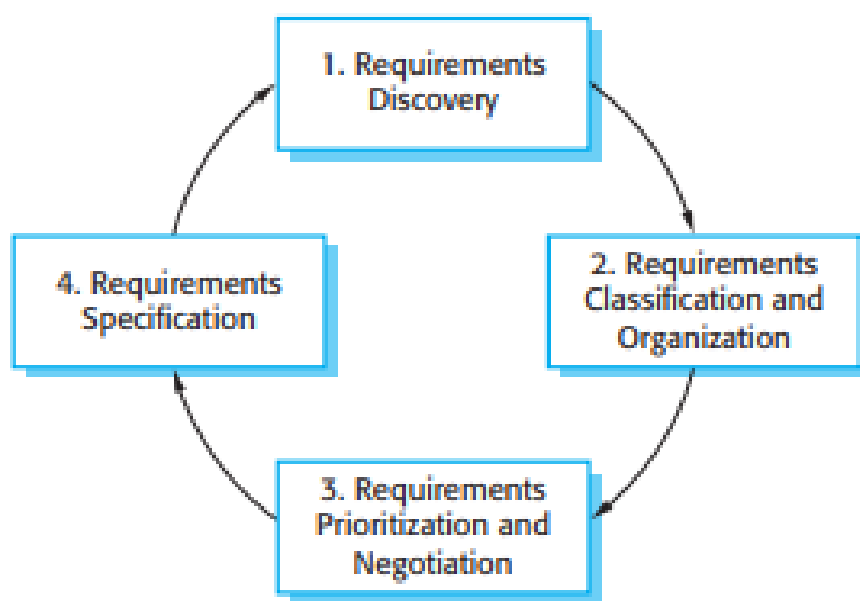
## 2.2 Μοντέλο Διαδικασίας Ανάλυσης Απαιτήσεων

Ένα μοντέλο της διαδικασίας ανάλυσης απαιτήσεων παρουσιάζεται στην Εικόνα 2. Οι δραστηριότητες της διαδικασίας είναι οι εξής:

1. **«Ανακάλυψη» απαιτήσεων:** Πρόκειται για τη διαδικασία αλληλεπίδρασης με τους «μετόχους» (stakeholders) του συστήματος για την “ανακάλυψη” των απαιτήσεών τους. Κατά τη διάρκεια αυτής της δραστηριότητας ανακαλύπτονται επίσης οι απαιτήσεις του πεδίου εφαρμογής από τους «μετόχους» καθώς και οι απαιτήσεις τεκμηρίωσης.
2. **Ταξινόμηση και οργάνωση των απαιτήσεων:** Η δραστηριότητα αυτή λαμβάνει την αδόμητη συλλογή απαιτήσεων, ομαδοποιεί τις σχετικές απαιτήσεις και τις οργανώνει σε συνεκτικές ομάδες. Ο συνηθέστερος τρόπος ομαδοποίησης των απαιτήσεων είναι η χρήση ενός μοντέλου της αρχιτεκτονικής του συστήματος για τον προσδιορισμό των υποσυστημάτων και τη

συσχέτιση των απαιτήσεων με κάθε υποσύστημα. Στην πράξη, ο σχεδιασμός απαιτήσεων και ο αρχιτεκτονικός σχεδιασμός δεν μπορούν να είναι εντελώς ξεχωριστές δραστηριότητες

3. **Ιεράρχηση απαιτήσεων και επίλυση συγκρούσεων:** Αναπόφευκτα, όταν εμπλέκονται πολλά ενδιαφερόμενα μέρη, οι απαιτήσεις θα συγκρούονται. Αυτή η δραστηριότητα αφορά την ιεράρχηση των απαιτήσεων και την εύρεση και επίλυση των συγκρούσεων απαιτήσεων μέσω διαπραγμάτευσης. Συνήθως, τα ενδιαφερόμενα μέρη πρέπει να συναντηθούν για να επιλύσουν τις διαφορές και να συμφωνήσουν σε συμβιβαστικές απαιτήσεις.
4. **Προδιαγραφές απαιτήσεων:** Οι απαιτήσεις τεκμηριώνονται και εισάγονται στον επόμενο γύρο της σπείρας. Μπορούν να παραχθούν επίσημα ή ανεπίσημα έγγραφα απαιτήσεων



Εικόνα 2: Μοντέλο ανάλυσης απαιτήσεων

### 2.3 Η μέθοδος ιεράρχησης MoSCoW

Η μέθοδος ιεράρχησης MoSCoW χρησιμοποιείται στη διοίκηση, την επιχειρηματική ανάλυση, τη διαχείριση έργων και την ανάπτυξη λογισμικού για την επίτευξη κοινής αντίληψης με τα ενδιαφερόμενα μέρη/συνεργάτες σχετικά με τη σημασία που αποδίδουν στην παράδοση κάθε απαίτησης. Η μέθοδος αναπτύχθηκε για πρώτη φορά από τον Dai Clegg το 1994 [4, 5] και χρησιμοποιείται ευρέως από το 2002. Το ακρωνύμιο "MoSCoW" αντιπροσωπεύει τις ακόλουθες τέσσερις κατηγορίες απαιτήσεων:

- **Must have:** Η απαίτηση πρέπει να ικανοποιηθεί απαραίτητως, ώστε η ανάπτυξη του συστήματος να θεωρείται επιτυχής.

- **Should have:** Η απαίτηση είναι υψηλής προτεραιότητας, αλλά θα μπορούσε να ικανοποιηθεί με άλλο τρόπο, αν είναι αυστηρά απαραίτητο.
- **Could have:** Η απαίτηση είναι επιθυμητή, αλλά όχι απαραίτητη και θα καλυφθεί αν υπάρχουν επαρκείς πόροι και χρόνος.
- **Will Not (Won't) have:** Η απαίτηση δεν θα ικανοποιηθεί στην παρούσα έκδοση του συστήματος. Πιθανώς να ικανοποιηθεί σε μελλοντική έκδοση.

Η μέθοδος MoSCoW είναι μια απλή αλλά αποτελεσματική τεχνική για την ιεράρχηση των απαιτήσεων, επειδή αναγκάζει τους ενδιαφερόμενους να σκεφτούν προσεκτικά τη σημασία κάθε απαίτησης και να κάνουν συμβιβασμούς. Είναι επίσης μια ευέλικτη τεχνική που μπορεί να χρησιμοποιηθεί σε διάφορα πλαίσια, από μικρά έργα έως μεγάλες πρωτοβουλίες σε επίπεδο επιχείρησης.

Προσφέρει σημαντικά οφέλη στο έργο. Πρώτα απ' όλα, πρόκειται για μια ευέλικτη τεχνική, οπότε η ιεράρχηση μπορεί να προσαρμοστεί ανάλογα με τις ανάγκες. Αυτό μπορεί να είναι απαραίτητο λόγω αλλαγών στο επιχειρηματικό περιβάλλον, στις ανάγκες των πελατών ή σε άλλους παράγοντες. Επιπλέον, βοηθά τα ενδιαφερόμενα μέρη/συνεργάτες να επικοινωνούν και να συνεργάζονται αποτελεσματικότερα, παρέχοντας ένα κοινό πλαίσιο για τη συζήτηση και την ιεράρχηση των απαιτήσεων. Ως εκ τούτου, βοηθά τις ομάδες να είναι πιο αποτελεσματικές και παραγωγικές, εστιάζοντας στις πιο σημαντικές απαιτήσεις και αποφεύγοντας τις περιττές επανεργασίες. Μια άλλη σημαντική πτυχή είναι ότι η μέθοδος MoSCoW συμβάλλει στη βελτίωση της ικανοποίησης των πελατών, διασφαλίζοντας ότι ικανοποιούνται οι προαναφερθείσες πιο σημαντικές απαιτήσεις..

Για να επιτευχθεί μια επιτυχής εφαρμογή της μεθόδου MoSCoW, θα πρέπει να ακολουθηθούν προσεκτικά τα ακόλουθα βήματα: Α) Πρώτον, πρέπει να προσδιορίσει κανείς όλες τις απαιτήσεις για το εκάστοτε υποσύστημα ή τις αρμοδιότητές του στο πλαίσιο του έργου. Β) Δεύτερον, πρέπει να κατηγοριοποιήσει κάθε απαίτηση ως "Πρέπει", "Πρέπει", "Θα μπορούσε", "Δεν θα" έχει, αν και οι τελευταίες δεν θα αποτελούν μέρος της τρέχουσας έκδοσης του παρόντος παραδοτέου. Γ) Τρίτον, πρέπει να ιεραρχήσει κανείς τις απαιτήσεις που πρέπει να έχει και στη συνέχεια να εκτιμήσει την προσπάθεια που απαιτείται για την υλοποίηση κάθε απαίτησης. Τέλος, δ) ο υπεύθυνος για τη συγγραφή κάθε ομάδας απαιτήσεων θα πρέπει να κάνει συμβιβασμούς και να προσαρμόσει την ιεράρχηση ανάλογα με τις ανάγκες.

Συνοψίζοντας, η μέθοδος MoSCoW είναι μια απλή αλλά ισχυρή τεχνική για την ιεράρχηση των απαιτήσεων. Προσφέρει μια σειρά από οφέλη, όπως η βελτίωση της επικοινωνίας και της συνεργασίας, η μείωση της διασποράς του πεδίου εφαρμογής, η αύξηση της αποδοτικότητας και της παραγωγικότητας και η βελτίωση της ικανοποίησης του πελάτη. Είναι εύκολη στη χρήση και μπορεί να προσαρμοστεί σε διάφορα πλαίσια. Ωστόσο, θα πρέπει να έχει κανείς κατά νου ότι είναι σημαντικό

να λάβει τη σύμφωνη γνώμη όλων των ενδιαφερομένων μερών/εταίρων και να είναι ευέλικτος κατά τη χρήση της μεθόδου.

Σε αυτή τη βάση, από την αρχή του έργου ζητήθηκε από όλους τους εταίρους να παρέχουν απαιτήσεις συστήματος, βασισμένες σε εσωτερική τεχνική ανάλυση, εμπειρία από προηγούμενα σχετικά έργα, προσδοκίες τελικών χρηστών κ.τ.λ. Για τις ανάγκες της συλλογής απαιτήσεων δημιουργήθηκε ένα τυποποιημένο υπόδειγμα με τη μορφή Google Sheet, το οποίο και συμπλήρωσε ο κάθε εταίρος ανεξάρτητα. Η συμπλήρωση από κάθε εταίρο έγινε ανεξάρτητα, με σκοπό να περιοριστεί η μεροληψία απάντησης (response bias) που θα μπορούσε να προκύψει σε περίπτωση συγκέντρωσης των απαιτήσεων σε ένα κοινό διαμοιραζόμενο έγγραφο.

Κατ' αρχάς, συγκεντρώθηκε ένα αρκετά ευρύ σύνολο από απαιτήσεις, το οποίο ανανεωνόταν καθώς προχωρούσε η σχεδίαση της αρχιτεκτονικής. Στη συνέχεια, ακολούθησε μία διαδικασία μετά-επεξεργασίας (καθαρισμού και ομογενοποίησης) των αρχικών απαιτήσεων, η οποία και οδήγησε στο τελικό σύνολο απαιτήσεων που παρουσιάζεται στο παρόν παραδοτέο.

## 2.4 Ορολογία

Στον παρακάτω πίνακα παραθέτουμε τα πεδία του υποδείγματος που χρησιμοποιήθηκε για τη συλλογή των απαιτήσεων από τους εταίρους.

Πίνακας 2: Πεδία υποδείγματος συλλογής απαιτήσεων

Πεδίο	Περιγραφή
Τύπος	<p>Ο τύπος καθορίζει αν η απαίτηση είναι <b>λειτουργική</b> (functional) ή <b>μη-λειτουργική</b> (non-functional) [2], [3].</p> <ul style="list-style-type: none"> <li>• Λειτουργικές είναι οι απαιτήσεις που περιγράφουν λειτουργίες που οφείλουν να επιτελούν στοιχεία του συστήματος και είναι προσανατολισμένες στη δράση (action-oriented).</li> </ul> <p>Μη λειτουργικές είναι οι απαιτήσεις που αφορούν σε χαρακτηριστικά ή ιδιότητες που πρέπει/δεν πρέπει να διαθέτουν τα στοιχεία του συστήματος, πλην της παρατηρούμενης συμπεριφοράς τους.</p>
Πρωτεύουσα κατηγορία	<p>Η πρωτεύουσα κατηγορία στην οποία υπάγεται η απαίτηση.</p> <p>Ενδεικτικές κατηγορίες απαιτήσεων είναι οι παρακάτω:</p> <ul style="list-style-type: none"> <li>• <i>Διαλειτουργικότητα (Interoperability)</i>: η ικανότητα δύο ή περισσότερων συστημάτων να ανταλλάσσουν πληροφορία με συγκροτημένο τρόπο και να χρησιμοποιούν την πληροφορία που ανταλλάχθηκε. Μπορεί να διακριθεί σε επιπλέον επίπεδα, π.χ. τεχνική, συντακτική, σημασιολογική κ.τ.λ.</li> <li>• <i>Λειτουργικότητα (Functionality)</i>: η ικανότητα ενός συστήματος να παρέχει λειτουργίες που ικανοποιούν τις ανάγκες των χρηστών.</li> <li>• <i>Συμβατότητα (Compatibility)</i>: η ικανότητα ενός συστήματος να μπορεί να λειτουργήσει αποτελεσματικά σε συνεργασία με άλλα στοιχεία στο περιβάλλον του.</li> <li>• <i>Συντηρησιμότητα (Maintainability)</i>: η ευκολία με την οποία ένα σύστημα μπορεί να μεταβληθεί/αναβαθμιστεί για να διορθώσει σφάλματα, να βελτιώσει την απόδοσή του, να επανέλθει σε μία προγενέστερη κατάστασή του ή να προσαρμοστεί σε μεταβολή του περιβάλλοντος.</li> <li>• <i>Αξιοπιστία (Reliability)</i>: η ικανότητα ενός συστήματος να λειτουργεί χωρίς την εμφάνιση σφαλμάτων για δεδομένη χρονική περίοδο.</li> </ul>

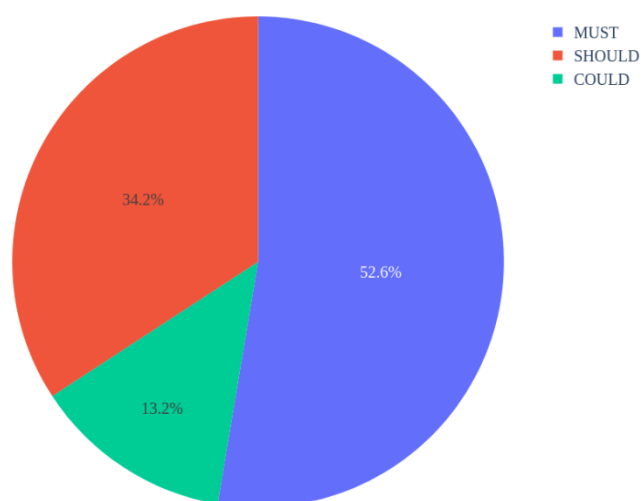
	<ul style="list-style-type: none"> <li>• <i>Αποδοτικότητα (Performance Efficiency)</i>: η ικανότητα ενός συστήματος να επιτελεί με αποδοτικό τρόπο (με περιορισμένη χρήση πόρων) τις λειτουργίες για τις οποίες έχει σχεδιαστεί.</li> <li>• <i>Φορητότητα (Portability)</i>: η ικανότητα ενός συστήματος να μεταφέρεται από ένα περιβάλλον σε άλλο (μεταφορά και προσαρμογή).</li> <li>• <i>Κλιμακωσιμότητα (Scalability)</i>: η ικανότητα ενός συστήματος να διαχειρίζεται όλο και μεγαλύτερο φόρτο εργασίας, προσθέτοντάς κατάλληλους πόρους.</li> <li>• <i>Ασφάλεια (Security)</i>: η ικανότητα ενός συστήματος να προστατεύει τα δεδομένα και την πληροφορία που διαθέτει, έτσι ώστε να επιτρέπεται η ανάκτηση και η τροποποίησή τους από κατάλληλα εξουσιοδοτημένα άτομα. Ιδιότητες της ασφάλειας είναι η αυθεντικοποίηση, ακεραιότητα, εμπιστευτικότητα, λογοδοσία, εξουσιοδότηση κ.ά.</li> <li>• <i>Ιδιωτικότητα (Privacy)</i>: η ικανότητα ενός συστήματος να ελαχιστοποιεί τα δεδομένα που αποκαλύπτει σε τρίτα μέρη αλλά και να επεξεργάζεται δεδομένα με συγκεκριμένο σκοπό, ύστερα από συναίνεση μέσω πολιτικών και ελέγχου.</li> <li>• <i>Χρηστικότητα</i>: η ευκολία με την οποία ένα χρήστης μπορεί να χρησιμοποιήσει το σύστημα, συμπεριλαμβανομένης της προσπάθειας που απαιτείται για την εκμάθηση, το χειρισμό και την ερμηνεία της εξόδου του συστήματος.</li> </ul> <p><i>Προσβασιμότητα (Accessibility)</i>: Η ικανότητα ενός συστήματος να είναι προσβάσιμο από όλους τους χρήστες για τους οποίους προορίζεται.</p>
<b>Δευτερεύουσα κατηγορία</b>	<p>Η δεύτερη κατηγορία στην οποία μπορεί να ανήκει μία απαίτηση. Ισχύουν οι ίδιες ενδεικτικές κατηγορίες που αναφέρθηκαν στην πρωτεύουσα κατηγορία.</p>
<b>Προτεραιότητα</b>	<p>Η προτεραιότητα ικανοποίησης της απαίτησης:</p> <ul style="list-style-type: none"> <li>• <i>MUST</i></li> <li>• <i>SHOULD</i></li> <li>• <i>COULD</i></li> <li>• <i>WON'T</i></li> </ul>
<b>Περιγραφή</b>	<p>Περιλαμβάνει μία σύντομη και περιεκτική περιγραφή της απαίτησης.</p>

Σκεπτικό	Περιγράφεται η ανάγκη που καλύπτει η συγκεκριμένη απαίτηση και το σκεπτικό επιλογής της.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Περιγράφεται η μεθοδολογία ή το κριτήριο ελέγχου της απαίτησης, ώστε να διασφαλιστεί η ικανοποίησή της στο υλοποιημένο σύστημα.

### 3. Λειτουργικές Απαιτήσεις Πλατφόρμας

Η παρούσα ενότητα περιλαμβάνει μία υψηλού επιπέδου ανάλυση και παρουσίαση των τελικών απαιτήσεων συστήματος, έτσι ώστε να είναι εφικτή μία συνολική επισκόπηση των αναμενόμενων αποτελεσμάτων του έργου.

Στην Εικόνα 3 απεικονίζεται η κατανομή της προτεραιότητας των τελικών απαιτήσεων. Σύμφωνα με το διάγραμμα, η ικανοποίηση των περισσότερων απαιτήσεων (52.8%) κρίνεται απολύτως αναγκαία (MUST), το 34.2% αυτών είναι υψηλής προτεραιότητας (SHOULD) και το 13.2% αυτών είναι επιθυμητό να ικανοποιηθεί. Δεν υπάρχουν απαιτήσεις οι οποίες δε θα ικανοποιηθούν σε αυτή την έκδοση του συστήματος (WON'T).



Εικόνα 3: Προτεραιότητα απαιτήσεων

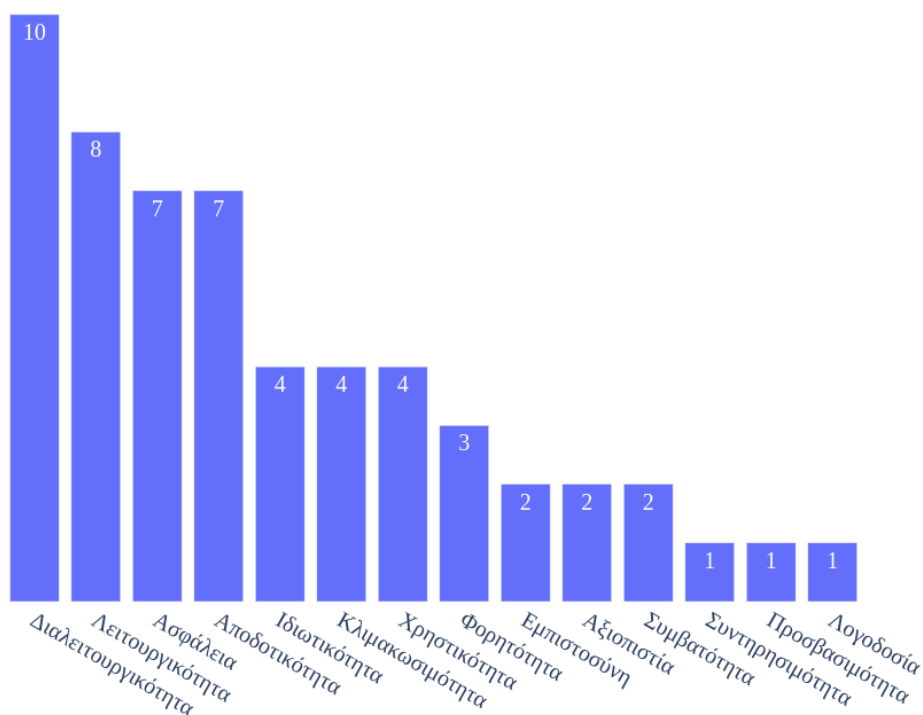
Στον Πίνακα 2 φαίνεται το πλήθος των απαιτήσεων που ανήκουν σε κάθε τύπο. Υπενθυμίζεται ότι, κατά τη συνήθη πρακτική, ορίζουμε δύο τύπους απαιτήσεων: (α) λειτουργικές, οι οποίες αφορούν στις λειτουργίες που το σύστημα οφείλει να προσφέρει στους χρήστες, και (β) μη-λειτουργικές, οι οποίες περιγράφουν ποιοτικά χαρακτηριστικά και περιορισμούς στους οποίους πρέπει να υπόκειται το σύστημα. Φαίνεται ότι υπάρχει μια ισορροπία μεταξύ των λειτουργικών και των μη-λειτουργικών απαιτήσεων που έχουν οριστεί.

Πίνακας 3: Απαιτήσεις ανά τύπο

Τύπος απαίτησης	Πλήθος
Λειτουργική	18
Μη-λειτουργική	20

Στην Εικόνα 4 απεικονίζεται το ραβδόγραμμα συχνοτήτων των κατηγοριών στις οποίες ανήκουν οι τελικές απαιτήσεις συστήματος. Επισημαίνεται ότι το διάγραμμα απεικονίζει συγκεντρωτικά τις συχνότητες για πρωτεύουσες και δευτερεύουσες κατηγορίες. Οι επικρατούσες κατηγορίες είναι η διαλειτουργικότητα, η λειτουργικότητα, η ασφάλεια και η αποδοτικότητα.

Η μετα-πλατφόρμα που θα αναπτυχθεί στα πλαίσια του έργου «Διασυνδεδεμένες Πόλεις για την Ελλάδα 2.0» σκοπεύει να αποτελέσει μία λύση διαλειτουργικότητας για την ενοποίηση υπάρχουσών κάθετων υλοποιήσεων Έξυπνων Πόλεων, τόσο ορίζοντας ένα κοινό εννοιολογικό μοντέλο περιγραφής δεδομένων και υπηρεσιών, όσο και αναπτύσσοντας μια κοινή υπολογιστική διεπαφή για την επίτευξη της διασύνδεσης εξωτερικών υπηρεσιών κατανάλωσης των δεδομένων. Επίσης, η λύση που πρόκειται να σχεδιαστεί και υλοποιηθεί, έχει ως στόχο να προσφέρει στους τελικούς ωφελούμενους ένα ασφαλές και αξιόπιστο περιβάλλον για το διαμοιρασμό και την ανταλλαγή των ομογενοποιημένων δεδομένων έξυπνων πόλεων, προσφέροντας μηχανισμούς ελέγχου πρόσβασης και χρήσης αλλά και ασφαλούς διακίνησης και αποθήκευσης αυτών. Τέλος, η μετα-πλατφόρμα σκοπεύει να υποστηρίξει την ανάπτυξη εφαρμογών μεγάλων σε όγκο δεδομένων από πολλαπλές πηγές, π.χ. εφαρμογές ανάλυσης δεδομένων με τεχνικές μηχανικής μάθησης. Συνεπώς, κρίνεται απαραίτητο η τεχνική λύση που θα υλοποιηθεί να επιτελεί με αποδοτικό τρόπο τις σχετικές λειτουργίες. Τα παραπάνω δικαιολογούν τις πιο συχνά εμφανιζόμενες κατηγορίες απαιτήσεων, όπως παρατηρούνται στο διάγραμμα.



Εικόνα 4: Απαιτήσεις ανά κατηγορία

Συγκεντρωτικά οι τελικές απαιτήσεις συστήματος του έργου παρουσιάζονται παρακάτω.

ID	ΑΠ. 1
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Ασφάλεια
Δευτερεύουσα κατηγορία	Εμπιστοσύνη
Προτεραιότητα	MUST
Περιγραφή	Υλοποίηση λύσης Διαχείρισης Ταυτότητας και Πρόσβασης, ευθυγραμμισμένη με τις ανάγκες της πλατφόρμας.
Σκεπτικό	Η πλατφόρμα πρέπει να ενσωματώνει ένα ολοκληρωμένο σύστημα διαχείρισης ταυτοτήτων και ελέγχου πρόσβασης στους πόρους και τις υπηρεσίες της, τόσο από την πλευρά των παρόχων πλατφορμών έξυπνων πόλεων, όσο και από την πλευρά των ωφελουμένων. Ορισμός διακριτών ρόλων, με διαφορετικά επίπεδα πρόσβασης στους πόρους της πλατφόρμας.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Επιτυχής αυθεντικοποίηση χρηστών με διαφορετικούς ρόλους και πρόσβαση μόνο στους εξουσιοδοτημένους πόρους και υπηρεσίες, βάσει ρόλου.

ID	ΑΠ. 2
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Ασφάλεια
Δευτερεύουσα κατηγορία	
Προτεραιότητα	SHOULD
Περιγραφή	Ορισμός τουλάχιστον τριών διακριτών ρόλων χρήστη (super admin, admin για τους φορείς και user για τους απλούς χρήστες).
Σκεπτικό	Η διαφοροποίηση των ρόλων χρηστών προσφέρει στο σύστημα καλύτερη διαχείριση, επιτρέποντας/περιορίζοντας τον εκάστοτε ρόλο το περιεχόμενο που του αναλογεί.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Κατάλληλος διαχωρισμός αρμοδιοτήτων και ρόλων.

ID	ΑΠ. 3
Τύπος	Μη λειτουργική
Πρωτεύουσα κατηγορία	Ασφάλεια
Δευτερεύουσα κατηγορία	Ιδιωτικότητα
Προτεραιότητα	MUST
Περιγραφή	Σύστημα εγγραφής και διαγραφής χρηστών, σύμφωνα με το ΓΚΠΔ.
Σκεπτικό	Η πλατφόρμα πρέπει να επιτρέπει την εγγραφή και διαγραφή χρηστών, διασφαλίζοντας τη συμμόρφωση με το ΓΚΠΔ.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Επιτυχής δημιουργία και διαγραφή χρήστη σύμφωνα με τις απαιτήσεις του ΓΚΠΔ.

ID	ΑΠ. 4
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Ιδιωτικότητα
Δευτερεύουσα κατηγορία	
Προτεραιότητα	MUST
Περιγραφή	Το σύστημα πρέπει να παρέχει απομονωμένη πρόσβαση στους πόρους και τις υπηρεσίες της, για πολλαπλούς οργανισμούς/χρήστες (πολύ-ενοικίαση).

<b>Σκεπτικό</b>	Η διατήρηση επαρκούς απομόνωσης μεταξύ των πόρων (δεδομένων, υπηρεσιών) των διαφορετικών tenants κρίνεται απαραίτητη για τη διασφάλιση της προστασίας ευαίσθητων δεδομένων των χρηστών της πλατφόρμας.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής πρόσβαση αυθεντικοποιημένων χρηστών διαφορετικών ρόλων και διαφορετικών οργανισμών μόνο στους εξουσιοδοτημένους πόρους του συγκεκριμένου οργανισμού.

ID	ΑΠ. 5
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Ασφάλεια
<b>Δευτερεύουσα κατηγορία</b>	Λογοδοσία
<b>Προτεραιότητα</b>	COULD
<b>Περιγραφή</b>	Το σύστημα θα μπορούσε να παρέχει δυνατότητες καταγραφής ιστορικού επερωτήσεων ανάκτησης από ταυτοποιημένους χρήστες.
<b>Σκεπτικό</b>	Οι περιπτώσεις διαρροής δεδομένων ή κακόβουλης χρήσης/πρόσβασης στην πληροφορία αποτελούν σημαντικό κίνδυνο. Σε περιπτώσεις που τα δεδομένα είναι απαραίτητα για τη διευρέυση ατυχημάτων ή καταστροφών πρέπει να διασφαλίζεται η ακεραιότητα στις διαδικασίες πρόσβασης.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	ΠΑΝ. ΠΑΤΡΩΝ

ID	ΑΠ. 6
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Ασφάλεια
<b>Δευτερεύουσα κατηγορία</b>	Ιδιωτικότητα
<b>Προτεραιότητα</b>	SHOULD
<b>Περιγραφή</b>	Το σύστημα πρέπει να παρέχει ασφαλή αποθήκευση δεδομένων με κατάλληλες τεχνικές.
<b>Σκεπτικό</b>	Η ασφάλεια των δεδομένων των χρηστών αποτελεί θέμα μείζονος προτεραιότητας και σημασίας για τη διασφάλιση της ιδιωτικότητας.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής χρήση σύγχρονων τεχνικών κρυπτογράφησης και κατακερματισμού για την προστασία δεδομένων at-rest.

ID	ΑΠ. 7
Τύπος	Μη λειτουργική
Πρωτεύουσα κατηγορία	Ασφάλεια
Δευτερεύουσα κατηγορία	Ιδιωτικότητα
Προτεραιότητα	MUST
Περιγραφή	Το σύστημα πρέπει να εξασφαλίζει τον ασφαλή διαμοιρασμό των δεδομένων (μεταφορά και ανάκτηση).
Σκεπτικό	Η ασφαλής μεταφορά των δεδομένων είναι πρωταρχικής σημασίας για το σύστημα ώστε να εξασφαλιστεί η εμπιστευτικότητα και η ακεραιότητα των δεδομένων που διαμοιράζονται.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Επιτυχής χρήση σύγχρονων τυποποιημένων πρωτοκόλλων ασφάλειας στο επίπεδο μεταφοράς, π.χ. Transport Layer Security (TLS), κατά την ανταλλαγή των δεδομένων.

ID	ΑΠ. 8
Τύπος	Μη λειτουργική
Πρωτεύουσα κατηγορία	Εμπιστοσύνη
Δευτερεύουσα κατηγορία	Ασφάλεια
Προτεραιότητα	SHOULD
Περιγραφή	Η πλατφόρμα είναι χρήσιμο να παρέχει μηχανισμό σύναψης συμφωνίας/συμβολαίου μεταξύ παρόχου και καταναλωτή δεδομένων.
Σκεπτικό	Η ύπαρξη συμβολαίου επιτρέπει την εφαρμογή κανόνων πρόσβασης και χρήσης των δεδομένων, διασφαλίζοντας την κυριαρχία του παραγωγού στα δεδομένα του (data sovereignty).
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Επιτυχής σύναψη συμβολαίων με συγκεκριμένους κανόνες πρόσβασης και χρήσης των παρεχόμενων δεδομένων, χρησιμοποιώντας πρωτόκολλο διαπραγμάτευσης συμβολαίου (π.χ. IDS Contract Negotiation Protocol)

ID	ΑΠ. 9
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Διαλειτουργικότητα
Δευτερεύουσα κατηγορία	

<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Ορισμός οντολογίας που συγκεντρώνει τις έννοιες συστημάτων ΕΠ, χρησιμοποιώντας σημασιολογική μοντελοποίηση των δεδομένων.
<b>Σκεπτικό</b>	Απαραίτητο για την ενοποίηση του πλαισίου δεδομένων που προέρχονται από διαφορετικά συστήματα ΕΠ και διαμοιράζονται διαμέσου της μετα-πλατφόρμας ΔτΠ.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Ικανότητα περιγραφής των εννοιών και των σχέσεων μεταξύ δεδομένων που ακολουθούν διαφορετικά data format/μοντέλα δεδομένων και προέρχονται από ετερογενείς πηγές.

<b>ID</b>	<b>ΑΠ. 10</b>
<b>Τύπος</b>	Λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Διαλειτουργικότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Ομογενοποίηση δεδομένων που προέρχονται από διαφορετικές πηγές, έτσι ώστε να ακολουθούν το κοινό μοντέλο δεδομένων της πλατφόρμας.
<b>Σκεπτικό</b>	Απαραίτητο για την επίτευξη σημασιολογικής διαλειτουργικότητας στο επίπεδο των δεδομένων.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής συλλογή/ανταλλαγή δεδομένων από διαφορετικά κάθετα συστήματα ΕΠ, σύμφωνα με το μοντέλο δεδομένων της πλατφόρμας.

<b>ID</b>	<b>ΑΠ. 11</b>
<b>Τύπος</b>	Λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Διαλειτουργικότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Διεπαφές της πλατφόρμας με συστήματα που παρέχουν ετερογενή δεδομένα μέσω των πιο δημοφιλών πρωτοκόλλων επικοινωνίας.
<b>Σκεπτικό</b>	Απαραίτητο για την εύκολη ενσωμάτωση πηγών δεδομένων από διάφορους παρόχους κάθετων υλοποιήσεων ΕΠ.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής συλλογή και ανάκτηση δεδομένων από διαφορετικές πηγές μέσω διαφορετικών πρωτοκόλλων επικοινωνίας (π.χ. HTTP, MQTT κ.ά.)

ID	ΑΠ. 12
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Διαλειτουργικότητα
Δευτερεύουσα κατηγορία	Συμβατότητα
Προτεραιότητα	MUST
Περιγραφή	Το σύστημα πρέπει να παρέχει σαφώς ορισμένες και τυποποιημένες διεπαφές για την ανάκτηση των δεδομένων από τους ωφελούμενους (καταναλωτές δεδομένων).
Σκεπτικό	Η χρήση τυποποιημένων και σαφώς ορισμένων διεπαφών είναι απαραίτητη για την επίτευξη διαλειτουργικότητας μεταξύ ετερογενών συστημάτων (π.χ. παρόχων πλατφορμών ΕΠ, παρόχων εφαρμογών/υπηρεσιών τρίτων), στο επίπεδο διεπαφών δεδομένων.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Επιτυχής ανταλλαγή δεδομένων μέσω τυποποιημένου REST API δεδομένων, π.χ. NGSI-LD.

ID	ΑΠ. 13
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Διαλειτουργικότητα
Δευτερεύουσα κατηγορία	Χρηστικότητα
Προτεραιότητα	COULD
Περιγραφή	Εδραίωση βασικών προτύπων κάθετων υλοποιήσεων με στόχο την ταχύτερη ενσωμάτωσή τους.
Σκεπτικό	Με την ύπαρξη βασικών προτύπων, η ενσωμάτωση των κάθετων υλοποιήσεων γίνεται ανώδυνα και προάγει την ομοιογένεια του συστήματος. Δηλαδή αν πολλές κάθετες εφαρμογές απαιτούν την ίδια οπτικοποίηση και έχουν τον ίδιο τύπο εισόδου/εξόδου, τότε αυτό μπορεί να αναπτυχθεί σαν ένα template.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Η χρηστικότητα και οι δυνατότητες διαλειτουργικότητας των προτύπων αυτών.

ID	ΑΠ. 14
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Διαλειτουργικότητα

<b>Δευτερεύουσα κατηγορία</b>	Συμβατότητα
<b>Προτεραιότητα</b>	SHOULD
<b>Περιγραφή</b>	Κάθε κάθετη υλοποίηση ΕΠ που λειτουργεί ως πάροχος δεδομένων είναι απαραίτητο να διαθέτει τυποποιημένη τεκμηρίωση της προγραμματιστικής διεπαφής που παρέχει για την ανάκτηση των δεδομένων (π.χ. περιγραφή OpenAPI/Swagger)
<b>Σκεπτικό</b>	Η ύπαρξη τυποποιημένης τεκμηρίωσης της προγραμματιστικής διεπαφής μιας κάθετης υλοποίησης διευκολύνει την υλοποίηση κατάλληλου προσαρμογέα για την ανάκτηση των δεδομένων και την προσαρμογή τους στο μοντέλο δεδομένων της μετα-πλατφόρμας.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Έλεγχος ύπαρξης τυποποιημένης τεκμηρίωσης.

<b>ID</b>	<b>ΑΠ. 15</b>
<b>Τύπος</b>	Λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Διαλειτουργικότητα
<b>Δευτερεύουσα κατηγορία</b>	Φορητότητα
<b>Προτεραιότητα</b>	COULD
<b>Περιγραφή</b>	Η πλατφόρμα θα μπορούσε να επιτρέπει το διαμοιρασμό υπηρεσιών μεταξύ των κάθετων υλοποιήσεων ΕΠ.
<b>Σκεπτικό</b>	Ο διαμοιρασμός υπηρεσιών επιτρέπει την επαναχρησιμοποίηση υπαρχουσών ανεπτυγμένων και δοκιμασμένων λειτουργικοτήτων από άλλες κάθετες υλοποιήσεις. Αυτό θα ωφελήσει με άμεση προστιθέμενη αξία επί των δεδομένων τις κάθετες υλοποιήσεις που θα υιοθετήσουν τις υπηρεσίες αυτές.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Υλοποίηση App Store με υπηρεσίες που προσφέρονται από τις κάθετες υλοποιήσεις ΕΠ.

<b>ID</b>	<b>ΑΠ. 16</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Αξιοπιστία
<b>Δευτερεύουσα κατηγορία</b>	

<b>Προτεραιότητα</b>	SHOULD
<b>Περιγραφή</b>	Το σύστημα θα πρέπει να διασφαλίζει την ανθεκτικότητα των δεδομένων σε περιπτώσεις αστοχίας υλικού ή διακοπών δικτύου.
<b>Σκεπτικό</b>	Στο σύστημα θα πρέπει να εφαρμόζονται στρατηγικές αναπαραγωγής και δημιουργίας αντιγράφων ασφαλείας, για την ελαχιστοποίηση της πιθανότητας απώλειας δεδομένων.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	ΠΑΝ ΠΑΤΡΩΝ

<b>ID</b>	<b>ΑΠ. 17</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Αξιοπιστία
<b>Δευτερεύουσα κατηγορία</b>	Αποδοτικότητα
<b>Προτεραιότητα</b>	SHOULD
<b>Περιγραφή</b>	Η πλατφόρμα θα πρέπει να ενσωματώνει μηχανισμούς εποπτείας των υποσυστημάτων του μέσω κατάλληλων μετρικών και analytics (π.χ. χρησιμοποίηση CPU, RAM, δικτυακών πόρων κ.ά.).
<b>Σκεπτικό</b>	Η επιτήρηση της κατάστασης των υποσυστημάτων επιτρέπει τη βελτιστοποίηση της απόδοσης λειτουργίας της πλατφόρμας, την ανίχνευση και διόρθωση σφάλματων αλλά και το σχεδιασμό της κλιμακωσιμότητας.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	ΠΑΝ ΠΑΤΡΩΝ

<b>ID</b>	<b>ΑΠ. 18</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Κλιμακωσιμότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Δυνατότητα προσαρμογής των πόρων (υπολογιστικών, αποθηκευτικών, δικτυακών) που γίνονται διαθέσιμοι στην πλατφόρμα προκειμένου η εκτέλεσή της να είναι πάντοτε ομαλή.
<b>Σκεπτικό</b>	Απαραίτητη σε περίπτωση που η πλατφόρμα αποκτήσει περισσότερους χρήστες ή/και υπηρεσίες και συνεπώς οι απαιτήσεις σε υπολογιστικούς πόρους ξεπεράσουν τους αρχικά διαθέσιμους.

<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής οριζόντια κλιμάκωση στοιχείων της πλατφόρμας στις εξής περιπτώσεις: (α) αύξησης του όγκου των ανακτώμενων/ επεξεργαζόμενων δεδομένων από τις κάθετες υλοποιήσεις, (β) αύξησης του πλήθους των χρηστών που αιτούνται χρήσης των παρεχόμενων υπηρεσιών.
--	--

<b>ID</b>	<b>ΑΠ. 19</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Κλιμακωσιμότητα
<b>Δευτερεύουσα κατηγορία</b>	Αποδοτικότητα
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Το σύστημα θα πρέπει να μπορεί να μεταφέρει μεγάλου όγκου δεδομένα με αποδοτικό τρόπο.
<b>Σκεπτικό</b>	Τα δεδομένα που ανταλλάσσονται εντός της πλατφόρμας μπορεί να αυξηθούν σημαντικά (ιδίως στην περίπτωση πραγματικού ή σχεδόν πραγματικού χρόνου δεδομένων), με αποτέλεσμα να επιβαρύνουν συγκεκριμένα στοιχεία του συστήματος.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Δυνατότητα διαμοιρασμού μεγάλου όγκου ετερογενών δεδομένων από διάφορες πηγές, για διαφορετικούς τύπους εφαρμογών.

<b>ID</b>	<b>ΑΠ. 20</b>
<b>Τύπος</b>	Λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Κλιμακωσιμότητα
<b>Δευτερεύουσα κατηγορία</b>	Αποδοτικότητα
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Η αρχιτεκτονική της πλατφόρμας θα πρέπει να υιοθετήσει μία προσέγγιση ομοσπονδίας δεδομένων (data federation) και όχι ένα κεντρικοποιημένο μοντέλο δεδομένων.
<b>Σκεπτικό</b>	Τα δεδομένα των παρόχων (κάθετων υλοποιήσεων ΕΠ) δε θα αποθηκεύονται σε κεντρικό αποθετήριο, αλλά θα διατηρούνται με αποκεντρωμένο τρόπο σε κάθε πάροχο, έτσι ώστε να βελτιώνεται η αποδοτικότητα και κλιμακωσιμότητα του συστήματος.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής διατήρηση και διαμοιρασμός δεδομένων που είναι αποθηκευμένα με αποκεντρωμένο τρόπο (πλην πιθανώς αυτών που προσωρινά

	συγκεράζονται ή υπόκεινται σε επεξεργασία από τις υπηρεσίες της πλατφόρμας).
--	--

ID	ΑΠ. 21
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Λειτουργικότητα
Δευτερεύουσα κατηγορία	
Προτεραιότητα	MUST
Περιγραφή	Η πλατφόρμα θα προσφέρει προηγμένες δυνατότητες πολυεπίπεδης και δυναμικής οπτικοποίησης ετερογενών δεδομένων.
Σκεπτικό	Η οπτικοποίηση ετερογενών δεδομένων που συλλέγονται από τις κάθετες υλοποιήσεις ΕΠ (παρόχους δεδομένων) επιτρέπει την ανάλυση, την εξαγωγή συμπερασμάτων και την κατανόηση της πληροφορίας.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Υλοποίηση κατάλληλων web dashboards για τη δυναμική οπτικοποίηση των δεδομένων που συλλέγονται από τους παρόχους δεδομένων.

ID	ΑΠ. 22
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Λειτουργικότητα
Δευτερεύουσα κατηγορία	Διαλειτουργικότητα
Προτεραιότητα	MUST
Περιγραφή	Η πλατφόρμα πρέπει να παρέχει κατάλληλους καταλόγους των διαθέσιμων δεδομένων.
Σκεπτικό	Η ύπαρξη καταλόγων δεδομένων είναι απαραίτητη για την εύκολη καταχώρηση των διαθέσιμων δεδομένων και των μεθόδων ανάκτησής τους από τους παρόχους, και την αποδοτική αναζήτηση αυτών από τους καταναλωτές.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Επιτυχής αναζήτηση πηγών δεδομένων σε κατάλογο δεδομένων με χρήση κατάλληλου REST API.

ID	ΑΠ. 23
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Λειτουργικότητα

<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Καθορισμός κριτηρίων αναζήτησης μέσα στις κάθετες υλοποιήσεις ΕΠ.
<b>Σκεπτικό</b>	Καθίσταται εφικτή η ανάκτηση φιλτραρισμένων δεδομένων (π.χ. χρονικό διάστημα, τύπος μετρήσεων, τύπος επεξεργασίας, κ.τ.λ.).
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής εφαρμογή φίλτρων για τον προσδιορισμό των δεδομένων οπτικοποίησης.

<b>ID</b>	<b>ΑΠ. 24</b>
<b>Τύπος</b>	Λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Λειτουργικότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Απεικόνιση χαρακτηριστικών κάθετων υλοποιήσεων ΕΠ με γενικές πληροφορίες (π.χ. πλήθος και τύποι αισθητήρων).
<b>Σκεπτικό</b>	Ο κάθε χρήστης θα μπορεί να ενημερωθεί για τις πληροφορίες των κάθετων υλοποιήσεων ΕΠ.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής και έγκυρη πληροφόρηση χρηστών.

<b>ID</b>	<b>ΑΠ. 25</b>
<b>Τύπος</b>	Λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Λειτουργικότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Η πλατφόρμα θα επιτρέπει τη λήψη συνόλου ιστορικών δεδομένων από τις κάθετες υλοποιήσεις ΕΠ (παρόχους δεδομένων).
<b>Σκεπτικό</b>	Η λήψη συνόλου ιστορικών δεδομένων επιτρέπει την οπτικοποίηση και ανάλυση τους με τεχνικές επεξεργασίας παρτίδων (batch processing).
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής λήψη συνόλου δεδομένων που αντιστοιχεί σε συγκεκριμένο χρονικό παραθύρο από τις κάθετες υλοποιήσεις ΕΠ, μέσω τυποποιημένων διεπαφών.

ID	ΑΠ. 26
Τύπος	Μη λειτουργική
Πρωτεύουσα κατηγορία	Λειτουργικότητα
Δευτερεύουσα κατηγορία	Αποδοτικότητα
Προτεραιότητα	MUST
Περιγραφή	Η πλατφόρμα θα πρέπει να επιτρέπει την ανταλλαγή δεδομένων "σωστού χρόνου" (right time) με αποδοτικό τρόπο.
Σκεπτικό	Απαραίτητο για την διασφάλιση της άμεσης εξαγωγής συμπερασμάτων και ειδοποιήσεων προς τους χρήστες της πλατφόρμας.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Υλοποίηση μηχανισμού pub/sub για την ανταλλαγή δεδομένων "σωστού χρόνου".

ID	ΑΠ. 27
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Λειτουργικότητα
Δευτερεύουσα κατηγορία	Διαλειτουργικότητα
Προτεραιότητα	SHOULD
Περιγραφή	Δυνατότητα υποβολής αιτήματος από αρμόδιο Φορέα για ενσωμάτωση κάθετης εφαρμογής στην κεντρική πλατφόρμα.
Σκεπτικό	Λειτουργία της πλατφόρμας, ώστε να είναι εφικτή η ενσωμάτωση νέων κάθετων εφαρμογών. Εδώ θα πρέπει να ορίζεται και ο τύπος της εφαρμογής, το συμβόλαιο κλήσης του API, η επιθυμητή οπτικοποίηση του αποτελέσματος κ.τ.λ.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Ικανότητα επιτυχούς υποβολής αιτήματος στην κεντρική διαχείριση της πλατφόρμας.

ID	ΑΠ. 28
Τύπος	Λειτουργική
Πρωτεύουσα κατηγορία	Λειτουργικότητα
Δευτερεύουσα κατηγορία	Διαλειτουργικότητα
Προτεραιότητα	SHOULD
Περιγραφή	Αποδοχή/Απόρριψη αιτήματος ενσωμάτωσης από την κεντρική διαχείριση της πλατφόρμας.

<b>Σκεπτικό</b>	Λειτουργία της πλατφόρμας, ώστε να είναι εφικτή η αποδοχή ή απόρριψη αιτημάτων ενσωμάτωσης. Καθορισμός και άλλων τυχόν απαραίτητων στοιχείων για την ενσωμάτωση μίας κάθετης εφαρμογής.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Ικανότητα επιτυχούς αποδοχής/απόρριψης και σχετική ενημέρωση του συστήματος και του Φορέα. Εμφάνιση της κάθετης εφαρμογής στην κεντρική πλατφόρμα.

<b>ID</b>	<b>ΑΠ. 29</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Χρηστικότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Κατάλληλη τεκμηρίωση του τρόπου εγκατάστασης και των λειτουργικοτήτων της πλατφόρμας (π.χ. οδηγοί χρήστη, τεκμηρίωση προγραμματιστικών διεπαφών, βίντεο εκπαίδευσης κ.ά.)
<b>Σκεπτικό</b>	Απαραίτητο για τη διευκόλυνση υιοθέτησης και χρήσης της πλατφόρμας.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Έλεγχος ύπαρξης και πρόσβασης από το χρήστη (διαχειριστή κάθετου συστήματος ΕΠ, τελικού ωφελούμενου) λεπτομερούς online τεκμηρίωσης.

<b>ID</b>	<b>ΑΠ. 30</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Χρηστικότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	SHOULD
<b>Περιγραφή</b>	Responsive design για βέλτιστη εμφάνιση σε όλες τις συσκευές.
<b>Σκεπτικό</b>	Βέλτιστη εμπειρία χρήστη από συσκευές διαφορετικού τύπου.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Επιτυχής εμπειρία χρήσης της πλατφόρμας από διαφορετικού τύπου συσκευές.

<b>ID</b>	<b>ΑΠ. 31</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Χρηστικότητα
<b>Δευτερεύουσα κατηγορία</b>	

<b>Προτεραιότητα</b>	<b>SHOULD</b>
<b>Περιγραφή</b>	Σχεδιασμός μινιμαλιστικών γραφικών διεπαφών χρήστη που χαρακτηρίζονται από απλότητα και αμεσότητα, καθιστώντας εύκολη τη χρήση της εφαρμογής και την υποστήριξη των λειτουργιών που προσφέρει χωρίς να απαιτείται ιδιαίτερη καθοδήγηση ή εκπαίδευση του χρήστη.
<b>Σκεπτικό</b>	Το περιβάλλον αλληλεπίδρασης οφείλει να είναι φιλικό προς μη εξειδικευμένους χρήστες, ώστε να αυξάνεται η χρηστικότητα.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Ευχάριστη εμπειρία χρήσης της πλατφόρμας από μη εξειδικευμένους χρήστες.

<b>ID</b>	<b>ΑΠ. 32</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Φορητότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	<b>SHOULD</b>
<b>Περιγραφή</b>	Η υλοποίηση της πλατφόρμας θα επιτρέπει την εύκολη εγκατάστασή της σε διαφορετικού τύπου υποδομές (π.χ. υπολογιστικό νέφος, on-premises κ.τ.λ.).
<b>Σκεπτικό</b>	Αναγκαία σε περίπτωση που η πλατφόρμα (ή μελλοντικές εκδόσεις αυτής) αποφασιστεί να φιλοξενηθεί σε άλλη υποδομή από αυτή που θα οριστεί αρχικά.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Υλοποίηση που θα επιτρέπει την εύκολη και επιτυχή εγκατάσταση νέων στιγμιότυπων της πλατφόρμας σε διαφορετικές υποδομές.

<b>ID</b>	<b>ΑΠ. 33</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Φορητότητα
<b>Δευτερεύουσα κατηγορία</b>	Κλιμακωσιμότητα
<b>Προτεραιότητα</b>	<b>COULD</b>
<b>Περιγραφή</b>	Τα στοιχεία της πλατφόρμας μπορούν να είναι εικονοποιημένα για λόγους φορητότητας.
<b>Σκεπτικό</b>	Διευκόλυνση της φορητότητας, αυτοματοποίησης και κλιμακωσιμότητας του συστήματος.

Περίπτωση ελέγχου / Κριτήριο αποδοχής	Ενσωμάτωση της χρήσης Docker containers κατά την ανάπτυξη και εγκατάσταση της πλατφόρμας.
---------------------------------------	---

<b>ID</b>	<b>ΑΠ. 34</b>
Τύπος	Μη λειτουργική
Πρωτεύουσα κατηγορία	Συντηρησιμότητα
Δευτερεύουσα κατηγορία	
Προτεραιότητα	SHOULD
Περιγραφή	Δυνατότητα εύκολης επέκτασης ή προσαρμογής των λειτουργιών και υποσυστημάτων που απαρτίζουν την πλατφόρμα.
Σκεπτικό	Απαραίτητη σε περίπτωση που χρειαστεί να γίνει οποιαδήποτε αλλαγή στον αρχικό σχεδιασμό της πλατφόρμας, ειδικά αν αυτή πραγματοποιηθεί από προγραμματιστή που δεν ανήκει στην αρχική ομάδα έργου.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Λεπτομερής περιγραφή της αρχιτεκτονικής και της λειτουργίας της πλατφόρμας, αρθρωτή δομή, χρήση κοινών τεχνολογιών και γλωσσών προγραμματισμού, εφαρμογή καλών πρακτικών προγραμματισμού.

<b>ID</b>	<b>ΑΠ. 35</b>
Τύπος	Μη λειτουργική
Πρωτεύουσα κατηγορία	Αποδοτικότητα
Δευτερεύουσα κατηγορία	
Προτεραιότητα	MUST
Περιγραφή	99% διαθεσιμότητα του συστήματος για την ανταλλαγή και εποπτεία των δεδομένων.
Σκεπτικό	Απαραίτητο για την διασφάλιση της απρόσκοπτης λειτουργίας του συστήματος.
Περίπτωση ελέγχου / Κριτήριο αποδοχής	Μέγιστο ποσοστό μη διαθεσιμότητας του συστήματος 1%.

<b>ID</b>	<b>ΑΠ. 36</b>
Τύπος	Μη λειτουργική
Πρωτεύουσα κατηγορία	Αποδοτικότητα

<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	MUST
<b>Περιγραφή</b>	Το σύστημα πρέπει να είναι αποδοτικό όσον αφορά την ταχύτητα υπολογισμών πάνω στα δεδομένα και το χρόνο απόκρισης.
<b>Σκεπτικό</b>	Το σύστημα θα πρέπει να μπορεί να αποκρίνεται χωρίς σημαντικές καθυστερήσεις στα ερωτήματα των χρηστών. Τεχνικές μετασχηματισμού, κατάτμησης ή/και συμπίεσης δεδομένων ή η αξιοποίηση edge computing δυνατοτήτων μπορούν να συμβάλουν στην βελτιστοποίηση της απόδοσης του συστήματος.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Χρονοκαθυστερήση μικρότερη των 10 λεπτών κατά την ανάκτηση δεδομένων.

<b>ID</b>	<b>ΑΠ. 37</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Αποδοτικότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	COULD
<b>Περιγραφή</b>	Υποστήριξη υπηρεσιών με διαφορετικές απαιτήσεις αποκρισιμότητας (delay critical, delay tolerant)
<b>Σκεπτικό</b>	Χρήσιμο για την υποστήριξη εφαρμογών για την συλλογή, διαχείριση, και επεξεργασία δεδομένων και την εξαγωγή συμπερασμάτων.
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	Υποστήριξη συστημάτων αίσθησης-απόφασης-ενέργειας (Sense-Decide-Act) για ετερογενή περιβάλλοντα, λαμβάνοντας υπόψη όλο το εύρος της δικτυακής υποδομής (υπολογιστικά νέφη, άκρα δικτύων πρόσβασης, συσκευές ΔτΠ - cloud edge, IoT continuum) και όλα τα είδη συνδεσιμότητας (κινητά και σταθερά δίκτυα ΔτΠ)

<b>ID</b>	<b>ΑΠ. 38</b>
<b>Τύπος</b>	Μη λειτουργική
<b>Πρωτεύουσα κατηγορία</b>	Προσβασιμότητα
<b>Δευτερεύουσα κατηγορία</b>	
<b>Προτεραιότητα</b>	SHOULD

<b>Περιγραφή</b>	Να αναπτυχθούν διεπαφές σύμφωνα με τα πρότυπα προσβασιμότητας για ΑμΕΑ (π.χ. Web Content Accessibility Guidelines – WCAG")
<b>Σκεπτικό</b>	ΠΑΝ ΠΑΤΡΩΝ
<b>Περίπτωση ελέγχου / Κριτήριο αποδοχής</b>	ΠΑΝ ΠΑΤΡΩΝ

## 4. Ευέλικτη Μεθοδολογία Ανάπτυξης

### 4.1 Εισαγωγή

Οι παραδοσιακές, καθοδηγούμενες από σχέδια προσεγγίσεις ανάπτυξης λογισμικού και συστημάτων, συχνά οδηγούν σε μακροχρόνιους κύκλους ανάπτυξης, με άκαμπτες διαδικασίες και προϊόντα που δεν ανταποκρίνονται πλήρως στις ανάγκες των πελατών. Ως απάντηση σε αυτούς τους περιορισμούς προέκυψε η Ευέλικτη Μέθοδος Ανάπτυξης Λογισμικού (Agile Software Development) είναι ένα σύνολο μεθοδολογιών και πρακτικών που αποσκοπούν στην παροχή λογισμικού υψηλής ποιότητας με ευέλικτο και προσαρμοστικό τρόπο.

Το Agile Manifesto, που συντάχθηκε από μια ομάδα προγραμματιστών λογισμικού το 2001, αποτελεί το θεμέλιο των ευέλικτων μεθοδολογιών και δίνει έμφαση στις παρακάτω τέσσερις βασικές αξίες:

- **Άτομα και αλληλεπιδράσεις έναντι διαδικασιών και εργαλείων (Individuals and Interactions over Processes and Tools):** Η Ευέλικτη Μέθοδος δίνει προτεραιότητα στην ανθρώπινη συνεργασία και επικοινωνία έναντι της στήριξης αποκλειστικά σε άκαμπτες διαδικασίες ή εργαλεία.
- **Λειτουργικό λογισμικό έναντι της ολοκληρωμένης τεκμηρίωσης (Working Software over Comprehensive Documentation):** Αν και η τεκμηρίωση είναι σημαντική, η Ευέλικτη Μέθοδος προωθεί την παράδοση λειτουργικού λογισμικού που μπορεί να δοκιμαστεί και να επικυρωθεί από τους χρήστες.
- **Συνεργασία με τον πελάτη έναντι διαπραγμάτευσης συμβάσεων (Customer Collaboration over Contract Negotiation):** Η Ευέλικτη Μέθοδος ενθαρρύνει την ενεργό συμμετοχή και την ανατροφοδότηση των πελατών καθ' όλη τη διάρκεια της διαδικασίας ανάπτυξης, διασφαλίζοντας ότι το τελικό προϊόν ευθυγραμμίζεται με τις προσδοκίες και ανάγκες τους.
- **Ανταπόκριση στην αλλαγή έναντι της τήρησης ενός σχεδίου (Responding to Change over Following a Plan):** Η Ευέλικτη Μέθοδος αναγνωρίζει το αναπόφευκτο της αλλαγής και δίνει έμφαση στην ικανότητα προσαρμογής και γρήγορης ανταπόκρισης στις εξελισσόμενες απαιτήσεις και προτεραιότητες.

Το Agile manifesto εξειδικεύεται περαιτέρω προτείνοντας δώδεκα αρχές που καθοδηγούν τις ομάδες ανάπτυξης. Οι αρχές αυτές δίνουν έμφαση στην ικανοποίηση των πελατών, την επαναληπτική ανάπτυξη, τη συνεργασία και τη συνεχή βελτίωση (Πίνακας 4).

Πίνακας 4: Βασικές αρχές της μεθοδολογίας Ευέλικτης Ανάπτυξης

Αρχή	Περιγραφή
1	Πρώτη μας προτεραιότητα είναι η ικανοποίηση του πελάτη/στόχου μέσω της έγκαιρης και συνεχούς παράδοσης χρήσιμου λογισμικού.
2	Οι αλλαγές στις απαιτήσεις είναι ευπρόσδεκτες, ακόμα και σε προχωρημένα στάδια της ανάπτυξης. Οι ευέλικτες διαδικασίες δαμάζουν τις αλλαγές με στόχο την ενίσχυση του ανταγωνιστικού πλεονεκτήματος του πελάτη/στόχου.
3	Παραδίδουμε συχνά λογισμικό που λειτουργεί, σε διαστήματα μερικών εβδομάδων ή μηνών, με προτίμηση στη συντομότερη χρονική κλίμακα.
4	Οι προγραμματιστές και οι ειδικοί της αγοράς πρέπει να συνεργάζονται καθημερινά καθ' όλη τη διάρκεια του έργου.
5	Θεμελιώνουμε τα έργα γύρω από άτομα με πάθος και ενδιαφέρον. Διαμορφώνουμε το κατάλληλο περιβάλλον, τους παρέχουμε την αναγκαία υποστήριξη, και εμπιστευόμαστε την ικανότητά τους να φέρουν σε πέρας την αποστολή τους.
6	Η πιο αποδοτική και αποτελεσματική μέθοδος για τη μετάδοση πληροφορίας προς και εντός της ομάδας ανάπτυξης λογισμικού είναι η συνομιλία πρόσωπο με πρόσωπο.
7	Το λογισμικό που λειτουργεί είναι το κύριο μέτρο προόδου.
8	Οι ευέλικτες διαδικασίες προάγουν την αειφόρο ανάπτυξη. Οι χορηγοί, η ομάδα ανάπτυξης λογισμικού και οι χρήστες θα πρέπει να είναι σε θέση να διατηρούν ένα σταθερό ρυθμό επ' αόριστον.
9	Η διαρκής έμφαση στην τεχνική αρτιότητα και στην εύρυθμη σχεδίαση ενισχύουν την ευελιξία.
10	Η απλότητα -- η τέχνη της μεγιστοποίησης του όγκου της δουλειάς που δεν χρειάζεται να γίνει -- είναι ουσιώδης.
11	Οι καλύτερες αρχιτεκτονικές, απαιτήσεις και σχέδια προκύπτουν από ομάδες που οργανώνονται μόνες τους.
12	Σε τακτά χρονικά διαστήματα, η ομάδα συλλογίζεται για το πώς θα γίνει πιο αποτελεσματική, ρυθμίζοντας και προσαρμόζοντας τη συμπεριφορά της αναλόγως.

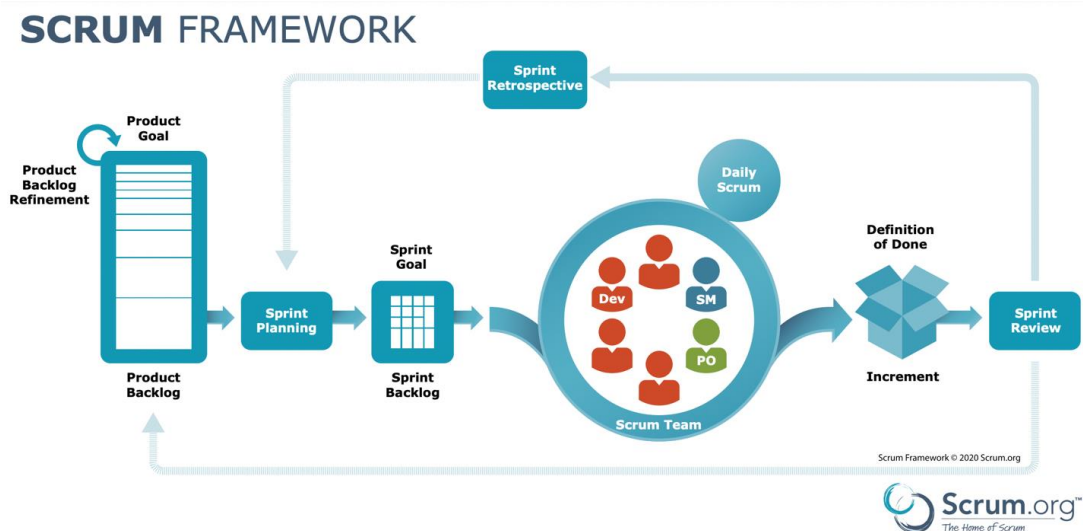
## 4.2 Σύνοψη Ευέλικτων (Agile) μεθοδολογιών και διαδικασιών

Ενώ η Ευέλικτη Ανάπτυξη είναι μια νοοτροπία και ένα σύνολο αρχών, διάφορες μεθοδολογίες θέτουν σε λειτουργία αυτές τις αρχές σε συγκεκριμένα πλαίσια για την ανάπτυξη λογισμικού. Ορισμένες από τις πιο ευρέως χρησιμοποιούμενες ευέλικτες μεθοδολογίες είναι οι εξής:

- **Scrum:** Το Scrum έχει δημιουργηθεί για τη διαχείριση έργων ανάπτυξης λογισμικού και οι διάφορες ομάδες, για παράδειγμα, των developers, εργάζονται σε σύντομους, επαναληπτικούς

κύκλους, τα επονομαζόμενα sprints με διάρκεια περίπου από δύο έως τέσσερις εβδομάδες. Κατά τη διάρκεια αυτού του διαστήματος οι ομάδες αυτές σχεδιάζουν, κατασκευάζουν, δοκιμάζουν και τελικά παραδίδουν το ολοκληρωμένο έργο. Χρησιμοποιεί ρόλους (Scrum Master, Product Owner, ομάδα ανάπτυξης), artifacts (Product Backlog, Sprint Backlog, Increment) και γεγονότα (Sprint Planning, Daily Standup, Sprint Review, Sprint Retrospective) για να δομήσει τη διαδικασία ανάπτυξης.

- **Kanban:** Το Kanban είναι ένα οπτικό ευέλικτο πλαίσιο που επικεντρώνεται στην οπτικοποίηση της εργασίας, στον περιορισμό της εργασίας σε εξέλιξη (Work In Progress - WIP) και στη μεγιστοποίηση της ροής. Χρησιμοποιεί έναν πίνακα Kanban με στήλες που αντιπροσωπεύουν διαφορετικά στάδια εργασίας και κάρτες που αντιπροσωπεύουν εργασίες ή ιστορίες χρήστη. Το Kanban δίνει έμφαση στη συνεχή παράδοση και τη βελτιστοποίηση της ροής εργασίας.



Εικόνα 5: Μέθοδος Scrum

Η Εικόνα 5 παρέχει μια οπτική αναπαράσταση της μεθόδου Scrum, απεικονίζοντας τους ρόλους, τα αντικείμενα και τα γεγονότα που εμπλέκονται στη διαδικασία. Επισημαίνει τον επαναληπτικό χαρακτήρα του Scrum, με συχνές/τακτικές ευκαιρίες για επιθεώρηση και προσαρμογή.



Εικόνα 6:Τυπικός πίνακας Kanban

Η οπτικοποίηση της εργασίας είναι κεντρικό στοιχείο του Kanban, και η Εικόνα 6<sup>1</sup> παρουσιάζει έναν τυπικό πίνακα Kanban που χρησιμοποιείται από τις ομάδες εργασίας. Δείχνει πώς οι δραστηριότητες μετακινούνται στα διάφορα στάδια της ροής, από το backlog μέχρι την τελική ολοκλήρωση, παρέχοντας διαφάνεια και διευκολύνοντας την αποτελεσματική συνεργασία.

### 4.3 Διαδικασία Ευέλικτης Ανάπτυξης (Agile Development Process)

Η Ευέλικτη Ανάπτυξη χαρακτηρίζεται από την επαναληπτική και σταδιακή προσέγγιση στην παράδοση λογισμικού. Η διαδικασία περιλαμβάνει συνήθως τις ακόλουθες βασικές φάσεις:

- **Σχεδιασμός (Planning):** Τα ευέλικτα έργα ξεκινούν με τον αρχικό σχεδιασμό, όπου η ομάδα συνεργάζεται για να καθορίσει το πεδίο εφαρμογής, τους στόχους και τις προτεραιότητες του έργου. Αυτό συχνά περιλαμβάνει τη δημιουργία ενός Product Backlog ή ενός καταλόγου επιθυμητών χαρακτηριστικών και λειτουργιών.
- **Ανάπτυξη (Development):** Κατά τη διάρκεια των επαναλήψεων ανάπτυξης (Sprints στο Scrum), η ομάδα συνεργάζεται για την υλοποίηση και τον έλεγχο των χαρακτηριστικών, εστιάζοντας στην παράδοση υψηλής ποιότητας, λειτουργικών τμημάτων λογισμικού. Συχνά

<sup>1</sup> Πηγή: <https://www.freepik.com/>

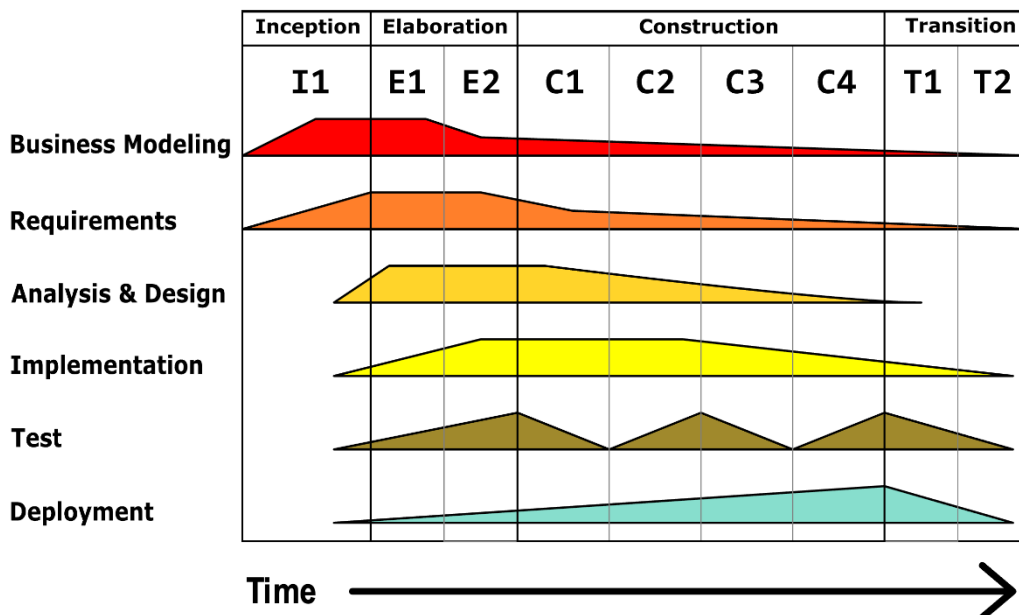
χρησιμοποιούνται πρακτικές συνεχούς ολοκλήρωσης και αυτοματοποιημένων δοκιμών για τη διασφάλιση της ποιότητας και της αξιοπιστίας του κώδικα.

- **Δοκιμές (Testing):** Οι δοκιμές ενσωματώνονται σε όλη τη διαδικασία ανάπτυξης, με ειδικές δραστηριότητες δοκιμών σε κάθε επανάληψη. Οι ευέλικτες ομάδες δίνουν προτεραιότητα στις έγκαιρες και συχνές δοκιμές για τον άμεσο εντοπισμό και την αντιμετώπιση ελαττωμάτων, διασφαλίζοντας ότι το λογισμικό πληροί τα πρότυπα ποιότητας.
- **Επιθεώρηση (Review):** Στο τέλος κάθε επανάληψης, η ομάδα διεξάγει μια επιθεώρηση Sprint (στο Scrum) ή μια παρόμοια συνάντηση για να επιδείξει την ολοκληρωμένη εργασία στους ενδιαφερόμενους και να συγκεντρώσει σχόλια. Αυτή η ανατροφοδότηση ενημερώνει τις επόμενες επαναλήψεις και βοηθά στην ιεράρχηση των μελλοντικών εργασιών.

Το Daily Standup ή Daily Scrum, είναι μια σύντομη, καθημερινή συνάντηση όπου τα μέλη της ομάδας συγχρονίζουν τις δραστηριότητές τους, συζητούν την πρόοδο και εντοπίζουν τυχόν εμπόδια ή δυσχέρειες. Συνήθως λαμβάνει χώρα μπροστά από έναν πίνακα Kanban ή Scrum, προωθώντας τη διαφάνεια και τη συνεργασία.

## Iterative Development

Business value is delivered incrementally in time-boxed crossdiscipline iterations.



Εικόνα 7: Κύκλος ζωής Ευέλικτης Ανάπτυξης

Η Εικόνα 7<sup>2</sup> περιγράφει την επαναληπτική φύση της ευέλικτης ανάπτυξης, επισημαίνοντας τις βασικές δραστηριότητες και τα παραδοτέα που σχετίζονται με κάθε φάση. Δίνει έμφαση στη συνεχή παράδοση

<sup>2</sup> Πηγή: Jakob Farian Krarup - Own work, CC0, <https://commons.wikimedia.org/w/index.php?curid=93364952>

αξίας στα ενδιαφερόμενα μέρη μέσω της σταδιακής ανάπτυξης και των συχνών βρόχων ανατροφοδότησης.

Με βάση το μοντέλο αυτό, διασφαλίζεται ότι το σύστημα εξελίσσεται με σταθερό και ελεγχόμενο τρόπο, λαμβάνοντας υπόψη τις ανάγκες και τα σχόλια των χρηστών σε κάθε στάδιο. Η ευέλικτη ανάπτυξη αποτελείται από **τέσσερις κύριες φάσεις** και **έξι βασικές πειθαρχίες (disciplines)**.

### Τέσσερις Φάσεις της ευέλικτης ανάπτυξης

#### 1. Έναρξη (Inception)

Σκοπός: Ορισμός του οράματος και του πεδίου εφαρμογής του έργου, καθώς και η κατανόηση των βασικών απαιτήσεων.

Δραστηριότητες: Καθορισμός των κρίσιμων απαιτήσεων, εκτίμηση των βασικών κινδύνων, ανάπτυξη ενός αρχικού προγραμματισμού και ενός επιχειρηματικού μοντέλου.

#### 2. Επιλογή (Elaboration)

Σκοπός: Ανάλυση και τεκμηρίωση των απαιτήσεων, σχεδιασμός της αρχιτεκτονικής του συστήματος και εξασφάλιση της τεχνικής βάσης.

Δραστηριότητες: Αναλυτική περιγραφή των απαιτήσεων, ανάπτυξη της αρχιτεκτονικής του συστήματος, διενέργεια λεπτομερών αναλύσεων και εκτίμηση του κόστους και των πόρων.

#### 3. Κατασκευή (Construction)

Σκοπός: Ανάπτυξη του συστήματος με βάση τις καθορισμένες απαιτήσεις και την αρχιτεκτονική.

Δραστηριότητες: Κωδικοποίηση, δοκιμή, ενσωμάτωση των στοιχείων του συστήματος και προετοιμασία για τη μετάβαση σε λειτουργία.

#### 4. Μετάβαση (Transition)

Σκοπός: Παράδοση του συστήματος στους τελικούς χρήστες και εξασφάλιση της ομαλής μετάβασης στη λειτουργία.

Δραστηριότητες: Εκπαίδευση των χρηστών, διενέργεια τελικών δοκιμών, διόρθωση προβλημάτων και διάθεση του συστήματος στην παραγωγή.

## Έξι Πειθαργίες (disciplines) της ευέλικτης ανάπτυξης

### 1. Διαχείριση Απαιτήσεων (Requirements)

Συλλογή, ανάλυση και τεκμηρίωση των απαιτήσεων του συστήματος. Αποτελεί τη βάση για τον προγραμματισμό και την ανάπτυξη.

### 2. Ανάλυση και Σχεδιασμός (Analysis and Design)

Ανάλυση των απαιτήσεων και σχεδιασμός της αρχιτεκτονικής του συστήματος. Περιλαμβάνει την καθορισμό των λειτουργιών και των δομών του συστήματος.

### 3. Υλοποίηση (Implementation)

Κωδικοποίηση του λογισμικού με βάση τις απαιτήσεις και τον σχεδιασμό. Περιλαμβάνει την ανάπτυξη του κώδικα και την ενσωμάτωση των διαφορετικών στοιχείων του συστήματος.

### 4. Δοκιμή (Testing)

Έλεγχος και δοκιμή του συστήματος για την εξασφάλιση της ποιότητας και της σωστής λειτουργίας. Περιλαμβάνει μονάδες, ολοκληρωμένες και συστηματικές δοκιμές.

### 5. Ανάπτυξη και Υποστήριξη (Deployment)

Διαχείριση της διάθεσης του συστήματος στους χρήστες και εξασφάλιση της σωστής λειτουργίας μετά την παράδοση. Περιλαμβάνει την εγκατάσταση, την εκπαίδευση και την υποστήριξη των χρηστών.

### 6. Διαχείριση Έργου (Project Management)

Παρακολούθηση και διαχείριση της προόδου του έργου. Περιλαμβάνει τον προγραμματισμό, την εκτίμηση κόστους, την παρακολούθηση των κινδύνων και την επικοινωνία με τους εμπλεκόμενους.

#### 4.3.1 Οφέλη της Μεθοδολογίας Ευέλικτης Ανάπτυξης

Οι ευέλικτες μεθοδολογίες έχουν φέρει επανάσταση στον τομέα της ανάπτυξης λογισμικού προσφέροντας μια ευέλικτη και προσαρμοστική προσέγγιση στη διαχείριση έργων και την παράδοση προϊόντων. Οι ευέλικτες μεθοδολογίες, που έχουν τις ρίζες τους στο ευέλικτο μανιφέστο και τις βασικές αρχές του, δίνουν προτεραιότητα στη συνεργασία με τους πελάτες, την επαναληπτική πρόοδο και τη συνεχή βελτίωση. Στο σημερινό δυναμικό και ταχέως εξελισσόμενο επιχειρηματικό περιβάλλον, η ευέλικτη μέθοδος έχει αναδειχθεί σε προτιμώμενη επιλογή για τις ομάδες ανάπτυξης λογισμικού και τους οργανισμούς που επιδιώκουν να ενισχύσουν την ανταπόκριση, την αποδοτικότητα και την ικανότητά τους να προσφέρουν αξία.

Με την υιοθέτηση των αρχών της ευέλικτης ανάπτυξης, οι ομάδες μπορούν να αποκομίσουν πολλά οφέλη, όπως αυξημένη ευελιξία για την προσαρμογή στις μεταβαλλόμενες απαιτήσεις, μεγαλύτερη

ταχύτητα παράδοσης του λογισμικού στην αγορά (speed to market), ενισχυμένη συνεργασία μεταξύ των μελών της ομάδας και των ενδιαφερόμενων μερών και παράδοση λογισμικού υψηλής ποιότητας.

Οι ευέλικτες μεθοδολογίες προσφέρουν πολυάριθμα οφέλη στις ομάδες και τους οργανισμούς ανάπτυξης λογισμικού, όπως:

#### 4.3.2 Ευελιξία (Flexibility)

- Οι ευέλικτες μεθοδολογίες προωθούν την προσαρμοστικότητα και την ευελιξία στην ανταπόκριση στις μεταβαλλόμενες απαιτήσεις και προτεραιότητες. Σε αντίθεση με τις παραδοσιακές προσεγγίσεις καταρράκτη, όπου οι αλλαγές είναι συχνά δύσκολο και δαπανηρό να εφαρμοστούν, η ευελιξία επιτρέπει στις ομάδες να προσαρμόζουν γρήγορα τα σχέδια και τις προτεραιότητές τους.
- Οι ευέλικτες ομάδες συμμετέχουν σε τακτικές επαναλήψεις ή Sprints, συνήθως διάρκειας 1-4 εβδομάδων, κατά τη διάρκεια των οποίων παραδίδουν σταδιακά νέα λειτουργικότητα (features) ή ενημερώσεις. Αυτή η επαναληπτική προσέγγιση επιτρέπει στις ομάδες να ενσωματώνουν ταχέως τα σχόλια των ενδιαφερομένων μερών και των πελατών, διασφαλίζοντας ότι το προϊόν παραμένει ευθυγραμμισμένο με τις εξελισσόμενες ανάγκες και τη δυναμική της αγοράς.
- Η ευελιξία των ευέλικτων μεθοδολογιών επιτρέπει στις ομάδες να αποδεχτούν την αβεβαιότητα και την πολυπλοκότητα, μετατρέποντας τις προκλήσεις σε ευκαιρίες για καινοτομία και βελτίωση. Με τη συνεχή βελτίωση και προσαρμογή των σχεδίων τους, οι ευέλικτες ομάδες μπορούν να διαχειρίζονται αποτελεσματικότερα τις αβεβαιότητες και να παρέχουν υπηρεσίες με υψηλή προστιθέμενη αξία με μεγαλύτερη συνέπεια.

#### 4.3.3 Ταχύτητα στην αγορά (Speed to Market)

- Οι ευέλικτες μεθοδολογίες δίνουν έμφαση στην ταχεία παράδοση λειτουργικού λογισμικού σε σύντομες επαναλήψεις. Με τον κατακερματισμό πολύπλοκων έργων σε μικρότερα, διαχειρίσιμα τμήματα, οι ευέλικτες ομάδες μπορούν να επιταχύνουν το χρόνο διάθεσης των προϊόντων και των λειτουργικότητά τους στην αγορά.
- Κάθε επανάληψη στην ευέλικτη ανάπτυξη παράγει ένα δυνητικά ολοκληρωμένο (shippable) βήμα προϊόντος, επιτρέποντας στις ομάδες να επικυρώνουν τη λειτουργικότητα και να συλλέγουν σχόλια από τους χρήστες νωρίς και συχνά. Αυτή η έγκαιρη επικύρωση επιτρέπει στις ομάδες να εντοπίζουν και να αντιμετωπίζουν τα προβλήματα νωρίτερα, μειώνοντας τον κίνδυνο δαπανηρών διορθώσεων ή καθυστερήσεων αργότερα στη διαδικασία ανάπτυξης.
- Η επαναληπτική προσέγγιση της ευέλικτης ανάπτυξης επιτρέπει επίσης στις ομάδες να ιεραρχούν την λειτουργικότητα με βάση την επιχειρηματική αξία και την ανατροφοδότηση των πελατών, διασφαλίζοντας ότι η πιο πολύτιμη λειτουργικότητα θα παραδοθεί πρώτη. Αυτό το

επαναληπτικό μοντέλο παράδοσης βοηθά τους οργανισμούς να παραμείνουν ανταγωνιστικοί στις ταχέως εξελισσόμενες αγορές, ανταποκρινόμενοι γρήγορα στις μεταβαλλόμενες ανάγκες των πελατών και στις τάσεις της αγοράς.

#### 4.3.4 Συνεργασία (Collaboration)

- Οι ευέλικτες μεθοδολογίες προωθούν μια κουλτούρα συνεργασίας και ομαδικής εργασίας μεταξύ διαλειτουργικών ομάδων, συμπεριλαμβανομένων των προγραμματιστών, των δοκιμαστών, των σχεδιαστών και των ιδιοκτητών προϊόντων. Με την κατάργηση των silo και την προώθηση της ανοιχτής επικοινωνίας, οι ευέλικτες ομάδες μπορούν να αξιοποιήσουν τις διαφορετικές δεξιότητες και προοπτικές των μελών της ομάδας για την αποτελεσματικότερη επίλυση προβλημάτων.
- Οι ευέλικτες τελετές (ceremonies), όπως οι καθημερινές συναντήσεις αναμονής, οι συνεδρίες σχεδιασμού sprint και οι ανασκοπήσεις sprint, παρέχουν τακτικές ευκαιρίες για συνεργασία και ευθυγράμμιση. Αυτές οι τελετές (ceremonies) επιτρέπουν στις ομάδες να συζητούν την πρόοδο, να εντοπίζουν τις εξαρτήσεις και να λαμβάνουν αποφάσεις σε συνεργασία, οδηγώντας σε καλύτερο συντονισμό και κοινή κατανόηση.
- Η ευελιξία δίνει επίσης έμφαση στη στενή συνεργασία με τα ενδιαφερόμενα μέρη (και τους πελάτες) καθ' όλη τη διάρκεια της διαδικασίας ανάπτυξης. Με την έγκαιρη και συχνή συμμετοχή των ενδιαφερόμενων μερών, οι ευέλικτες ομάδες μπορούν να διασφαλίσουν ότι το προϊόν ανταποκρίνεται στις ανάγκες και τις προσδοκίες τους.
- Αυτή η ενεργός συμμετοχή βοηθά στην οικοδόμηση εμπιστοσύνης και ευθυγράμμισης μεταξύ της ομάδας ανάπτυξης και των ενδιαφερόμενων μερών, οδηγώντας σε πιο επιτυχημένα αποτελέσματα.

#### 4.3.5 Ποιότητα (Quality)

- Οι ευέλικτες μεθοδολογίες δίνουν προτεραιότητα στην ποιότητα καθ' όλη τη διάρκεια της διαδικασίας ανάπτυξης, αντί να την αντιμετωπίζουν ως δευτερεύουσα σκέψη.
- Η συνεχής ενσωμάτωση (Continuous Integration), οι αυτοματοποιημένες δοκιμές και οι τακτικοί βρόχοι ανατροφοδότησης αποτελούν αναπόσπαστα μέρη της ευέλικτης ανάπτυξης, διασφαλίζοντας ότι η ποιότητα ενσωματώνεται στο προϊόν από την αρχή.
- Οι πρακτικές συνεχούς ενσωμάτωσης στην ευέλικτη ανάπτυξη περιλαμβάνουν τη συχνή ένταξη των αλλαγών στον κώδικα και την αυτόματη εκτέλεση δοκιμών για τον έγκαιρο εντοπισμό και την αντιμετώπιση ελαττωμάτων. Αυτή η προληπτική προσέγγιση στη διασφάλιση της ποιότητας βοηθά στον ταχύτερο εντοπισμό των προβλημάτων, μειώνοντας την πιθανότητα συσσώρευσης σφαλμάτων και τεχνικών λαθών με την πάροδο του χρόνου.

- Οι ευέλικτες ομάδες δίνουν επίσης προτεραιότητα στην αριστεία και τον καλό σχεδιασμό, εστιάζοντας στη δημιουργία καθαρού, συντηρήσιμου κώδικα που μπορεί να προσαρμόζεται στις μεταβαλλόμενες απαιτήσεις. Επενδύοντας εκ των προτέρων στην ποιότητα και δίνοντας έμφαση σε βιώσιμες πρακτικές ανάπτυξης, οι ευέλικτες ομάδες μπορούν να παραδώσουν λογισμικό υψηλότερης ποιότητας που ανταποκρίνεται στις προσδοκίες των πελατών και οδηγεί σε μεγαλύτερη ικανοποίηση.

Συνολικά, η ευελιξία, η ταχύτητα στην αγορά, η συνεργασία και η εστίαση στην ποιότητα που είναι εγγενείς στις ευέλικτες μεθοδολογίες επιτρέπουν στις ομάδες ανάπτυξης λογισμικού και στους οργανισμούς να παρέχουν προστιθέμενη αξία πιο αποτελεσματικά και να προσαρμόζονται στις μεταβαλλόμενες συνθήκες της αγοράς και στις ανάγκες των πελατών

## 4.4 Διαδικασία Ευέλικτης Ανάπτυξης Γραφικών Διεπαφών Χρήστη (Agile UX)

### 4.4.1 Εισαγωγή

Η μεθοδολογία Agile UX συνδυάζει τις αρχές του Agile λογισμικού με τις πρακτικές της σχεδίασης εμπειρίας χρήστη (User Experience Design - UX). Σκοπός είναι η δημιουργία προϊόντων, υπηρεσιών ή συστημάτων που ανταποκρίνονται στις ανάγκες των χρηστών με ευελιξία και ταχύτητα, μέσα από επαναλαμβανόμενους κύκλους ανάπτυξης και ανατροφοδότησης.

### 4.4.2 Βασικές Αρχές του Agile

Οι βασικές αρχές του Agile περιλαμβάνουν:

- **Επαναλαμβανόμενες Εργασίες:** Ανάπτυξη σε μικρούς, επαναλαμβανόμενους κύκλους (sprints).
- **Συνεργασία:** Στενή συνεργασία μεταξύ των ομάδων ανάπτυξης και των χρηστών /«πελατών».
- **Ανταπόκριση στην Αλλαγή:** Δυνατότητα προσαρμογής στις μεταβαλλόμενες απαιτήσεις.
- **Προτεραιότητα στον Χρήστη:** Εστίαση στις ανάγκες και την εμπειρία του χρήστη.

### 4.4.3 Διαδικασία Agile UX

- Η διαδικασία του Agile UX συνδυάζει τις αρχές του Agile development με τις πρακτικές της σχεδίασης εμπειρίας χρήστη (UX) για να δημιουργήσει προϊόντα που ικανοποιούν τις ανάγκες των χρηστών. Η Agile UX προσφέρει ευελιξία και προσαρμοστικότητα, επιτρέποντας στις ομάδες να ανταποκρίνονται άμεσα στις αλλαγές και στα σχόλια των χρηστών. Ακολουθεί μια σύντομη περιγραφή των βασικών σταδίων και αρχών της διαδικασίας:
- **Δημιουργία Ομάδας:** Στην αρχή, δημιουργείται μια διεπιστημονική ομάδα που περιλαμβάνει σχεδιαστές UX, προγραμματιστές, αναλυτές επιχειρήσεων και άλλους ενδιαφερόμενους. Η

συνεργασία είναι κεντρική στην Agile UX και ενισχύεται μέσω συνεχούς επικοινωνίας και συνεργασίας.

- **Κατανόηση των Χρηστών και των Αναγκών τους:** Η διαδικασία ξεκινά με την κατανόηση των χρηστών και των αναγκών τους. Αυτό επιτυγχάνεται μέσω ερευνών χρηστών, συνεντεύξεων, και ανάλυσης δεδομένων. Η δημιουργία personas και user stories βοηθά στην εστίαση στις πραγματικές ανάγκες των χρηστών.
- **Δημιουργία Απαιτήσεων και Προδιαγραφών:** Οι απαιτήσεις του προϊόντος καταγράφονται σε ένα backlog, το οποίο περιέχει τις user stories και τις προδιαγραφές. Αυτές οι απαιτήσεις διαμορφώνονται με βάση τα ευρήματα της έρευνας χρηστών και προτεραιοποιούνται ανάλογα με την αξία τους για τον χρήστη.
- **Σχεδίαση και Πρωτοτυποποίηση:** Η σχεδίαση γίνεται με επαναληπτικό τρόπο, ξεκινώντας από χαμηλής πιστότητας σκίτσα και wireframes, και προχωρώντας σε υψηλής πιστότητας πρωτότυπα. Τα πρωτότυπα δοκιμάζονται με χρήστες για να συλλεχθούν σχόλια και να βελτιωθούν οι σχεδιαστικές λύσεις.
- **Ανάπτυξη:** Η ανάπτυξη γίνεται σε σύντομους κύκλους, γνωστούς ως σπριντ, που διαρκούν από μία έως τέσσερις εβδομάδες. Σε κάθε σπριντ, η ομάδα εργάζεται πάνω σε συγκεκριμένες user stories, ολοκληρώνοντας μικρές, λειτουργικές εκδόσεις του προϊόντος.
- **Συνεχής Ανατροφοδότηση και Βελτίωση:** Η ανατροφοδότηση είναι κρίσιμη στην Agile UX. Οι σχεδιαστές και οι προγραμματιστές λαμβάνουν συνεχή σχόλια από τους χρήστες και τα ενδιαφερόμενα μέρη, τα οποία χρησιμοποιούνται για τη βελτίωση του προϊόντος σε κάθε σπριντ.
- **Δοκιμή και Επικύρωση:** Τα προϊόντα δοκιμάζονται διεξοδικά μέσω usability testing, A/B testing, και άλλων μεθόδων για να διασφαλιστεί ότι πληρούν τις απαιτήσεις των χρηστών και είναι εύκολα στη χρήση. Οι δοκιμές πραγματοποιούνται καθ' όλη τη διάρκεια της ανάπτυξης για την άμεση διόρθωση των προβλημάτων.
- **Παράδοση και Κυκλοφορία:** Η Agile UX επιτρέπει την τακτική παράδοση λειτουργικών εκδόσεων του προϊόντος, προσφέροντας αξία στους χρήστες σε σύντομα χρονικά διαστήματα. Η κυκλοφορία γίνεται σταδιακά, επιτρέποντας την άμεση αντίδραση στις ανατροφοδοτήσεις των χρηστών.
- **Επαναληψιμότητα:** Η διαδικασία της Agile UX είναι επαναληπτική. Μετά την παράδοση, η ομάδα συνεχίζει να βελτιώνει το προϊόν βάσει νέων δεδομένων και ανατροφοδοτήσεων, εξασφαλίζοντας ότι το προϊόν παραμένει σχετικό και χρήσιμο για τους χρήστες.

#### 4.4.4 Σύνοψη

Η Agile UX ενσωματώνει τη φιλοσοφία της συνεχούς βελτίωσης και της ευελιξίας, διασφαλίζοντας ότι τα προϊόντα εξελίσσονται διαρκώς για να καλύπτουν τις μεταβαλλόμενες ανάγκες και προσδοκίες των

χρηστών. Η υιοθέτηση της μεθοδολογίας Agile UX μπορεί να βελτιώσει σημαντικά τη διαδικασία ανάπτυξης προϊόντων, προσφέροντας καλύτερη εμπειρία χρήστη και αυξημένη ικανοποίηση των πελατών. Συνδυάζοντας τις αρχές του Agile με τις πρακτικές του UX, οι ομάδες μπορούν να δημιουργήσουν προϊόντα που είναι όχι μόνο λειτουργικά αλλά και εξαιρετικά αποδεκτά από τους χρήστες. Τα πλεονεκτήματα της Διαδικασίας Agile UX συνοψίζονται στον παρακάτω πίνακα.

Πίνακας 5: Τα Πλεονεκτήματα της Διαδικασίας Agile UX

Όφελος	Περιγραφή
Ταχύτητα και Ευελιξία	Επιτρέπει <b>γρήγορες προσαρμογές</b> στις αλλαγές και τις ανάγκες των χρηστών.
Καλύτερη Εμπειρία Χρήστη	Η <b>συνεχής ανατροφοδότηση</b> από τους χρήστες βοηθά στη δημιουργία προϊόντων που καλύπτουν καλύτερα τις ανάγκες τους
Μείωση Κόστους	Ο εντοπισμός και η επίλυση προβλημάτων <b>σε πρώιμο στάδιο</b> μειώνουν το κόστος ανάπτυξης

#### 4.5 Εφαρμογή των Μεθοδολογιών Ευέλικτης Ανάπτυξης στο πλαίσιο του έργου

Η ανάπτυξη της μετα-πλατφόρμας έξυπνων πόλεων που θα ενοποιεί και θα εναρμονίζει υφιστάμενες κάθετες υλοποιήσεις απαιτεί μια μεθολογία ευέλικτης ανάπτυξης (Agile Development) προκειμένου να διασφαλιστεί η προσαρμοστικότητα, η συνεργασία και η αποτελεσματικότητα σε ένα πολυ-εταρικό περιβάλλον. Η Ευέλικτη Ανάπτυξη επιτρέπει την τμηματική ανάπτυξη της πλατφόρμας, με συνεχείς βελτιώσεις και ενσωμάτωση ανατροφοδότησης από τους εμπλεκόμενους φορείς.

##### 1. Διαχείριση Απαιτήσεων με Sprint:

- Διαχωρισμός σε User Stories: Η μετατροπή των απαιτήσεων σε user stories επιτρέπει τη σαφή περιγραφή του τι χρειάζεται να αναπτυχθεί. Το Agile backlog περιλαμβάνει όλα τα user stories, τα οποία ιεραρχούνται ανάλογα με τις ανάγκες των χρηστών και τις επιχειρηματικές ή τεχνικές προτεραιότητες.
- Sprint Planning με Εργαλεία Διαχείρισης: Για την οργάνωση των sprint, η χρήση εργαλείων όπως το Jira ή το Azure DevOps προσφέρει δυνατότητες backlog management, καθορισμού sprint goals και παρακολούθησης της προόδου με burndown charts.
- Επαναληπτική Προσέγγιση: Κάθε sprint δίνει τη δυνατότητα για ανατροφοδότηση από τους ενδιαφερόμενους (stakeholders) και γρήγορη τροποποίηση των απαιτήσεων, ελαχιστοποιώντας τον κίνδυνο μεγάλης παρέκκλισης από τις αρχικές προδιαγραφές.

## 2. Διαλειτουργικότητα & Ενοποίηση:

- **API-first Αρχιτεκτονική:** Η υλοποίηση κάθε υπηρεσίας με τη φιλοσοφία “API-first” διασφαλίζει ότι τα δεδομένα είναι προσβάσιμα και αξιοποιήσιμα από διαφορετικούς καταναλωτές (π.χ. mobile εφαρμογές, web interfaces, άλλες πλατφόρμες). Χρησιμοποιούνται RESTful APIs ή GraphQL, ενώ για πιο σύνθετες ανάγκες real-time επικοινωνίας μπορεί να επιλεγεί WebSockets ή gRPC.
- **Μικρο-υπηρεσίες (Microservices):** Η διαχωρισμένη υλοποίηση σε microservices επιτρέπει την ανεξάρτητη ανάπτυξη και κλιμάκωση (scaling) επιμέρους λειτουργιών. Κάθε κάθετη εφαρμογή μπορεί να γίνει “αυτόνομο” microservice, το οποίο ενσωματώνεται ομαλά στο κεντρικό σύστημα μέσω κατάλληλων APIs ή event-driven μηχανισμών (Kafka, RabbitMQ).
- **Open Standards & Interoperability:** Χρήση προτύπων όπως το NGSI (Next Generation Service Interface) του FIWARE ή τοMQTT για IoT συσκευές. Έτσι, προωθείται η επικοινωνία μεταξύ ετερογενών συστημάτων με κοινά πρωτόκολλα και φορμά ανταλλαγής δεδομένων (JSON, XML, AVRO κ.λπ.).

## 3. Συνεργατική Ανάπτυξη:

- **Κοινά Αποθετήρια Κώδικα:** Η φιλοξενία του κώδικα σε πλατφόρμες όπως το GitHub ή το GitLab επιτρέπει την παράλληλη ανάπτυξη από πολλαπλές ομάδες. Με πολιτικές Branching (π.χ. Gitflow) και Pull Requests, διασφαλίζεται η ποιότητα και η επίλυση των merge conflicts.
- **Επικοινωνία & Συντονισμός:** Εργαλεία όπως το Slack, το Microsoft Teams ή το Mattermost, σε συνδυασμό με καθημερινά stand-up meetings, διευκολύνουν την αμεσότητα στην επίλυση τεχνικών προβλημάτων.
- **Κοινή “Γλώσσα” μέσα στο Οικοσύστημα:** Ο σχεδιασμός κοινών Data Models και Shared Libraries (π.χ. κοινές βιβλιοθήκες για ασφάλεια ή μοντέλα δεδομένων) βοηθά διαφορετικούς εταίρους να αναπτύσσουν λογισμικό με συνέπεια και κοινές βάσεις.

## 4. Πρωτότυπα και MVP (Minimum Viable Product):

- **Ταχεία Δημιουργία Πρωτοτύπου (Prototyping):** Η ανάπτυξη πρώιμων πρωτοτύπων επιτρέπει τη γρήγορη αξιολόγηση ιδεών – π.χ. η χρήση χαμηλής πιστότητας wireframes για το UX ή η δημιουργία sandbox APIs για δοκιμές ολοκλήρωσης.
- **Ανάπτυξη MVP:** Στο πρώτο στάδιο, επιλέγονται οι λειτουργίες που παρέχουν τη βασική αξία (core functionality) για τους χρήστες. Αυτό μπορεί να περιλαμβάνει ένα κεντρικό API gateway για την πρόσβαση στα δεδομένα, μία βασική εφαρμογή διαχείρισης (dashboard) και τη συνδεσιμότητα με ένα-δυο βασικούς αισθητήρες/υπηρεσίες.

- **Συνεχής Ανατροφοδότηση:** Μετά την παράδοση του MVP, γίνεται συλλογή σχολίων από εταίρους, χρήστες και ομάδες ανάπτυξης. Αυτά τα σχόλια τροφοδοτούν τον σχεδιασμό των επόμενων sprints, προσθέτοντας σταδιακά νέες δυνατότητες και βελτιώνοντας την εμπειρία χρήστη (UX).

#### 5. Δοκιμές και Συνεχής Ενσωμάτωση:

- **Περιβάλλον Ενσωμάτωσης (Integration Environment):** Υιοθετείται ένα περιβάλλον όπου κάθε commit ή pull request περνά από αυτοματοποιημένες δοκιμές (unit tests, integration tests, code quality checks). Εργαλεία όπως το Jenkins, το GitLab CI ή το GitHub Actions μπορούν να υλοποιήσουν τέτοιους μηχανισμούς.
- **Συνεχής Παράδοση (Continuous Delivery):** Μετά την επιτυχημένη εκτέλεση των δοκιμών, οι νέες εκδόσεις του λογισμικού μπορούν να αναπτυχθούν αυτόματα σε staging περιβάλλοντα. Η υλοποίηση σε containerized μορφή (π.χ. Docker) και η ορχήστρωση με Kubernetes διευκολύνουν την αυτοματοποιημένη ανάπτυξη σε κλίμακα.
- **DevSecOps Προσέγγιση:** Επιπλέον, ενσωματώνονται πρακτικές ασφαλείας (security scans, vulnerability assessments) στη ροή του CI/CD, εξασφαλίζοντας ότι κάθε αλλαγή στον κώδικα ελέγχεται και από πλευράς ασφάλειας πριν φτάσει στο production περιβάλλον.

#### 6. Δυναμική Διαχείριση Προτεραιοτήτων:

- **Κατηγοριοποίηση σε Must, Should, Could, Won't:** Η μέθοδος MoSCoW βοηθά στο να εστιάζει η ομάδα στις πιο κρίσιμες λειτουργίες. Η κατηγοριοποίηση των απαιτήσεων διασφαλίζει ότι τα “Must have” ολοκληρώνονται πρώτα, ενώ οι υπόλοιπες απαιτήσεις υλοποιούνται σταδιακά, ανάλογα με τη διαθεσιμότητα πόρων.
- **Ευθυγράμμιση με Επιχειρησιακούς Στόχους:** Με συχνά review και refinement των προτεραιοτήτων, η ανάπτυξη συμβαδίζει με τις πραγματικές ανάγκες των πόλεων, των επιχειρήσεων, των ερευνητικών κέντρων και των πολιτών.

#### 7. Ευέλικτος Σχεδιασμός Χρήστη (Agile UX):

- **Συνεχείς Δοκιμές Χρηστικότητας:** Χρήση A/B testing ή usability sessions για να αξιολογηθεί η ευκολία χρήσης και η κατανόηση της πλατφόρμας. Εργαλεία όπως το Figma ή το Sketch υποστηρίζουν ταχεία επαναληπτική βελτίωση των UI/UX σχεδίων.
- **Iterative UX Sprints:** Σε παράλληλο με την ανάπτυξη του λογισμικού, η ομάδα UX εκτελεί δικούς της μίνι-κύκλους (design sprints), παραδίδοντας wireframes, mockups ή prototypes για έγκαιρο feedback.

- Design Systems & Style Guides: Η ύπαρξη ενιαίων style guides ή design systems (π.χ. Material Design, Bootstrap-based guidelines) εξασφαλίζει ομοιομορφία στα UI components, επιταχύνει την ανάπτυξη και βελτιώνει τη συνολική εμπειρία χρήσης.

Συνοψίζοντας, η παραπάνω τεχνολογική προσέγγιση εστιάζει στην ασφάλεια, την κλιμάκωση, τη συνεχή ενσωμάτωση και την ευελιξία της ανάπτυξης. Μέσα από την αρχιτεκτονική microservices, τις πρακτικές DevOps/DevSecOps και την έμφαση στο Agile UX, η μετα-πλατφόρμα έξυπνων πόλεων μπορεί να ενσωματώνει πληθώρα υπηρεσιών και δεδομένων, να διασφαλίζει υψηλό επίπεδο αξιοπιστίας και να εξελίσσεται γρήγορα βάσει των αναγκών των χρηστών και των στρατηγικών στόχων των εμπλεκόμενων φορέων.

Με αυτό τον τρόπο επιτυγχάνεται η ομαλή ενοποίηση και η αξιοποίηση των κάθετων λύσεων, ενώ παράλληλα δημιουργείται ένα περιβάλλον συνεχούς καινοτομίας για επιχειρηματίες, ερευνητές και πολίτες.

.

## Σύνοψη

Αυτό το παραδοτέο, Π1.1, παρέχει μια λεπτομερή περιγραφή της μεθοδολογίας προσδιορισμού των απαιτήσεων που ακολουθήθηκε στα πλαίσια του έργου καθώς και τις απαιτήσεις που προέκυψαν. Επιπλέον, περιγράφει την ευέλικτη μεθοδολογία ανάπτυξης λογισμικού που θα χρησιμοποιηθεί για την υλοποίηση της πλατφόρμας.

Η περιγραφή της μεθοδολογίας προσδιορισμού των απαιτήσεων περιλαμβάνει την αναλυτική παρουσίαση της διαδικασίας που ακολουθήθηκε για την «ανακάλυψη» και την αξιολόγηση των απαιτήσεων μαζί με την απαραίτητη σχετική ορολογία. Οι τελικές απαιτήσεις της πλατφόρμας περιγράφονται αναλυτικά σε μορφή πίνακα. . Για την ιεράρχηση των απαιτήσεων χρησιμοποιήθηκε η μέθοδος MoSCoW, η οποία κατηγοριοποιεί τις απαιτήσεις με βάση την αναγκαιότητα υλοποίησης. Για κάθε μια απαίτηση αναφέρονται περαιτέρω λεπτομέρειες σχετικά με την κατηγορία της (πρωτεύουσα και δευτερεύουσα), η προτεραιότητα της, η περιγραφή και το σκεπτικό πίσω από την απαίτηση καθώς και ένα κριτήριο αποδοχής. Πιο συγκεκριμένα, προέκυψαν συνολικά 38 απαιτήσεις, εκ των οποίων 18 είναι λειτουργικές και 20 μη-λειτουργικές. Επιπλέον, κρίθηκε ότι η ικανοποίηση των περισσότερων απαιτήσεων (52.8%) κρίνεται απολύτως αναγκαία (MUST), το 34.2% αυτών είναι υψηλής προτεραιότητας (SHOULD) και το 13.2% αυτών είναι επιθυμητό να ικανοποιηθεί, ενώ δεν υπάρχουν απαιτήσεις οι οποίες δε θα ικανοποιηθούν σε αυτή την έκδοση του συστήματος (WON'T).

Τέλος το παραδοτέο περιγράφει την ευέλικτη μεθοδολογία ανάπτυξης λογισμικού που θα χρησιμοποιηθεί για την υλοποίηση της πλατφόρμας και τα οφέλη της.

## Αναφορές

- [1] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile software development methods: Review and analysis,” *VTT Publications*, no. 478. 2002.
- [2] K. M. Adams, “Non-functional Requirements in Systems Analysis and Design, Topics in Safety, Risk, Reliability and Quality Design Methodologies,” in *Design Methodologies*, vol. 28, 2015.
- [3] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. 2000.
- [4] International Institute of Business Analysis, *The Guide to the Business Analysis Body of Knowledge*. 2008.
- [5] I. Sommerville, “Software engineering 9th Edition,” *ISBN-10*, vol. 137035152, p. 18, 2011.