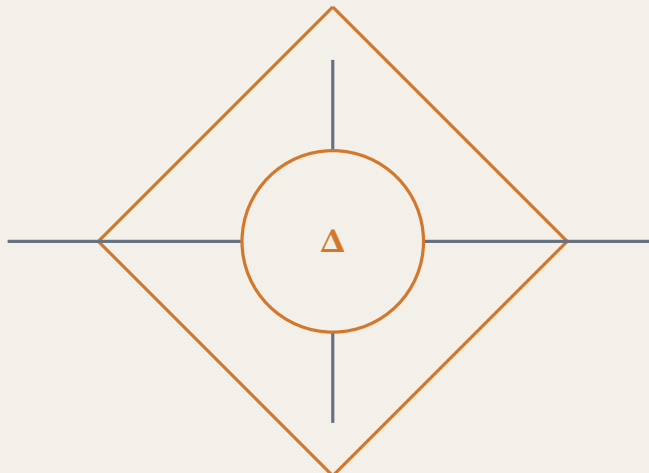


Δημιουργία Apps με Codex

Build, test και ship με σωστή μεθοδολογία



Πώς να χρησιμοποιήσεις αυτό το βιβλίο

Η έκδοση έχει σχεδιαστεί ως εφαρμοσμένο πρόγραμμα μάθησης. Στόχος δεν είναι η παθητική ανάγνωση, αλλά η μετατροπή κάθε κεφαλαίου σε συγκεκριμένη απόφαση, έλεγχο ή παραδοτέο.

01 Κατανόηση

Διάβασε πρώτα τον μηχανισμό, τους βασικούς όρους και τα όρια της μεθόδου.

02 Εφαρμοσε

Χρησιμοποίησε δικά σου δεδομένα ή ένα ασφαλές υποθετικό σενάριο στο εργαστήριο.

03 Έλεγε

Δοκίμασε κανονικές και δυσμενείς συνθήκες. Κατέγραψε παραδοχές και πηγές.

04 Αποφάσισε

Όρισε ενέργεια, υπεύθυνο, προθεσμία και ημερομηνία επανεξέτασης.

GNOSERA

KNOWLEDGE FOR THE NEXT ERA · BY INDIGO Με επιφύλαξη παντός δικαιώματος.

Ο σχεδιασμός, η επιμέλεια, η διάρθρωση και η παρούσα ψηφιακή έκδοση αποτελούν αποκλειστική εκδοτική παραγωγή της INDIGO GROUP MON. IKE. Απαγορεύεται η αντιγραφή, αναδημοσίευση, μεταπώληση, δημόσια ανάρτηση, τροποποίηση ή διανομή ολόκληρου ή μέρους του έργου χωρίς προηγούμενη έγγραφη άδεια του εκδότη.

Η αγορά παρέχει μία προσωπική, μη μεταβιβάσιμη άδεια χρήσης στον αρχικό αγοραστή. Η άδεια δεν περιλαμβάνει δικαίωμα εμπορικής εκμετάλλευσης, αναπαραγωγής ή παραχώρησης σε τρίτους.

ΣΗΜΑΝΤΙΚΗ ΣΗΜΕΙΩΣΗ

Εκπαιδευτικό επιχειρηματικό περιεχόμενο. Δεν αποτελεί εξαιρουμένη νομική, φορολογική ή επενδυτική συμβουλή.

Πώς χρησιμοποιείται αυτό το βιβλίο

Το βιβλίο είναι σχεδιασμένο ως εφαρμοσμένο πρόγραμμα. Δεν αρκεί να διαβάσεις τις έννοιες: πρέπει να ορίσεις πραγματική ροή, baseline, κριτήριο ποιότητας, υπεύθυνο και απόφαση. Κάθε κεφάλαιο κλείνει με εργαστήριο και checklist.

1 · ΚΑΤΑΝΟΗΣΗ

Ξεχώρισε τον μηχανισμό, τα δεδομένα, τον άνθρωπο και τα όρια.

2 · ΕΦΑΡΜΟΓΗ

Δούλεψε πάνω σε πραγματική ροή της δικής σου εταιρείας.

3 · ΕΛΕΓΧΟΣ

Δοκίμασε κανονικές, ακραίες και αποτυχημένες περιπτώσεις.

4 · ΑΠΟΦΑΣΗ

Κλιμάκωσε μόνο όταν αξία, ποιότητα και κίνδυνος είναι μετρήσιμα.

ΚΑΝΟΝΑΣ ΠΗΓΩΝ

Μεταβαλλόμενοι ισχυρισμοί ελέγχθηκαν έως 16/08/2026. Επίσημες πηγές, κανονιστικά κείμενα και τεχνική τεκμηρίωση υπερισχύουν εμπορικών summaries. Πριν από νομική, φορολογική, εργασιακή ή κανονιστική απόφαση απαιτείται επανέλεγχος από αρμόδιο επαγγελματία.

Περιεχόμενα

01	Από την ιδέα σε ελέγξιμη προδιαγραφή εφαρμογής	6
02	Επιλογή περιβάλλοντος: desktop, CLI, IDE και cloud	14
03	Ανάγνωση και χαρτογράφηση ενός υπάρχοντος codebase	22
04	AGENTS.md: κανόνες, όρια και μόνιμο project context	30
05	Task briefs και prompts που οδηγούν σε σωστή υλοποίηση	38
06	Plan mode, διάσπαση εργασίας και checkpoints	46
07	Ασφαλείς αλλαγές κώδικα, patches και έλεγχος diff	54
08	Testing, linting, type checks και definition of done	62

Περιεχόμενα

09	Συστηματικό debugging και root-cause analysis	70
10	Frontend εφαρμογές, responsive UI και visual QA	78
11	Git, branches, commits, pull requests και code review	86
12	Permissions, sandboxing, approvals, secrets και ασφαλής εκτέλεση	94
13	Skills, plugins και MCP: επεκτάσεις χωρίς λειτουργικό χάος	102
14	Local, cloud, remote και CI αυτοματισμοί	110
15	Capstone: από specification σε production release και maintenance plan	118

Από την ιδέα σε ελέγξιμη προδιαγραφή εφαρμογής

Να μετατρέψεις μια ασαφή επιχειρηματική ιδέα σε συγκεκριμένο πρόβλημα, χρήστες, ροές, δεδομένα, περιορισμούς και κριτήρια αποδοχής πριν ζητήσεις παραγωγή κώδικα.

Η επαγγελματική αξία αυτού του κεφαλαίου βρίσκεται στη σύνδεση ανάμεσα στο τεχνικό έργο και στην επιχειρησιακή ευθύνη. Το ζητούμενο «Από την ιδέα σε ελέγξιμη προδιαγραφή εφαρμογής» δεν αντιμετωπίζεται ως μεμονωμένο tip. Εντάσσεται σε αλυσίδα αποφάσεων όπου το αποτέλεσμα πρέπει να είναι κατανοητό, αναπαραγώγιμο και ασφαλές. Για αυτό θα παράγεις συγκεκριμένα artifacts, θα ελέγξεις τους συνηθέστερους κινδύνους και θα συνδέσεις την ολοκλήρωση με μετρήσιμα στοιχεία.

ΚΕΝΤΡΙΚΗ ΑΡΧΗ

Το Codex αποδίδει καλύτερα όταν το ζητούμενο περιγράφεται ως ελέγξιμη αλλαγή και όχι ως σύνθημα. Η προδιαγραφή δεν χρειάζεται να είναι τεράστια· πρέπει όμως να ξεχωρίζει τι είναι απαίτηση, τι είναι υπόθεση και τι δεν επιτρέπεται να αλλάξει.

ΤΙ ΘΑ ΠΑΡΑΔΩΣΕΙΣ

- Problem statement μιας παραγράφου
- User journey με happy path και failure paths
- Feature specification
- Definition of done
- Κατάλογος εκτός scope

Τα παραδοτέα αυτά δημιουργούν κοινή γλώσσα ανάμεσα σε ιδιοκτήτη προϊόντος, developer, reviewer και υπεύθυνο λειτουργίας. Αν κάποιος λείπει, κατέγραψε ρητά ποια απόφαση δεν μπορεί ακόμη να ληφθεί και ποια πληροφορία χρειάζεται για να κλείσει το κενό.

ΜΕΤΡΗΣΙΜΟ ΑΠΟΤΕΛΕΣΜΑ

Ποσοστό acceptance criteria που περνά · Αριθμός αλλαγών scope ανά κύκλο · Χρόνος από brief σε ελεγμένο vertical slice

1.1 Το σωστό νοητικό μοντέλο

Το Codex αποδίδει καλύτερα όταν το ζητούμενο περιγράφεται ως ελέγξιμη αλλαγή και όχι ως σύνθημα. Η προδιαγραφή δεν χρειάζεται να είναι τεράστια· πρέπει όμως να ξεχωρίζει τι είναι απαίτηση, τι είναι υπόθεση και τι δεν επιτρέπεται να αλλάξει.

Στο συγκεκριμένο κεφάλαιο, ο στόχος δεν είναι να απομνημονεύσεις ορολογία. Είναι να μπορείς να εξηγήσεις ποια απόφαση παίρνει ο άνθρωπος, ποια εργασία ανατίθεται στο Codex, ποια στοιχεία θεωρούνται πηγή αλήθειας και ποια απόδειξη θα σε κάνει να αποδεχθείς το αποτέλεσμα.

Η πρακτική εφαρμογή ξεκινά με το εξής ερώτημα: «Τι πρέπει να παραμένει αληθινό πριν και μετά την αλλαγή;». Από εκεί προκύπτουν τα όρια, τα tests και το score. Αν δεν μπορείς να απαντήσεις, το task χρειάζεται ακόμη discovery και όχι βιαστική παραγωγή κώδικα.

Για να γίνει η αρχή λειτουργική, σύνδεσέ την με τα παραδοτέα «Problem statement μιας παραγράφου» και «User journey με happy path και failure paths». Το πρώτο εξηγεί τι επιχειρούμε να πετύχουμε· το δεύτερο δείχνει πού συναντά τον πραγματικό κώδικα και τη λειτουργία. Η απόσταση μεταξύ των δύο είναι το πεδίο όπου εμφανίζονται συνήθως οι κρυφές απαιτήσεις.

Έπειτα όρισε μέτρηση. Στο συγκεκριμένο θέμα, ένα χρήσιμο σήμα είναι «Ποσοστό acceptance criteria που περνά». Το metric δεν αντικαθιστά το review, αλλά αναγκάζει την ομάδα να ορίσει τι θεωρεί βελτίωση και τι θα αποτελούσε λόγο επιστροφής στο σχέδιο.

ΠΑΡΑΔΕΙΓΜΑ

Μια μικρή εμπορική εταιρεία θέλει εφαρμογή outlet. Αντί για «φτιάξε app», η ομάδα ορίζει login, κατάλογο, φίλτρα, καλάθι, checkout, ιστορικό παραγγελιών, ρόλους, αποτυχίες πληρωμής και μετρήσιμα κριτήρια για κάθε ροή.

ΕΡΩΤΗΣΗ ΕΛΕΓΧΟΥ

Μπορεί ένας ανεξάρτητος reviewer να καταλάβει το πρόβλημα, το επιτρεπτό score και την απόδειξη επιτυχίας χωρίς να ζητήσει προφορικές διευκρινίσεις;

1.2 Workflow εφαρμογής

Η διαδρομή για το «Από την ιδέα σε ελέγξιμη προδιαγραφή εφαρμογής» αποτελείται από πέντε ελέγξιμα βήματα. Κάθε βήμα πρέπει να αφήνει ορατό παραδοτέο και να έχει συνθήκη ολοκλήρωσης. Αν ένα βήμα αποτύχει, δεν προχωράμε μηχανικά στο επόμενο.

01 · ΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΧΡΗΣΤΗ

Κατέγραψε input, owner και επιθυμητό output πριν εκτελεστεί.

02 · ΧΑΡΤΟΓΡΑΦΗΣΗ ΒΑΣΙΚΗΣ ΔΙΑΔΡΟΜΗΣ ΚΑΙ ΕΞΑΙΡΕΣΕΩΝ

Σύνδεσε το βήμα με συγκεκριμένο path, command ή evidence.

03 · ΚΑΤΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ, ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΠΕΡΙΟΡΙΣΜΩΝ

Έλεγε κανονική περίπτωση, εξαίρεση και αποτυχία.

04 · ΣΥΝΤΑΞΗ ACCEPTANCE CRITERIA

Αποθήκευσε την απόδειξη ώστε να μπορεί να γίνει review.

05 · ΜΙΚΡΟ VERTICAL SLICE ΓΙΑ ΤΗΝ ΠΡΩΤΗ ΑΠΟΔΕΙΞΗ

Αποφάσισε αν προχωράς, επανασχεδιάζεις ή σταματάς.

Κανόνας: καμία επιτυχία δεν θεωρείται πραγματική αν δεν μπορεί να επαναληφθεί στο συμφωνημένο περιβάλλον με τα ίδια verification steps.

1.3 Παραδοτέα και task brief

Τα παραδοτέα μετατρέπουν τη συζήτηση σε ελέγξιμη εργασία. Δεν χρειάζεται κάθε task να παράγει όλα τα παρακάτω, αλλά πρέπει να αφήνει αρκετό evidence ώστε ο επόμενος άνθρωπος να συνεχίσει χωρίς να ξανακάνει όλο το discovery.

- Problem statement μιας παραγράφου
- User journey με happy path και failure paths
- Feature specification
- Definition of done
- Κατάλογος εκτός scope

REUSABLE PROMPT

Μελέτησε το repository χωρίς αλλαγές. Εντόπισε τα σημεία που επηρεάζονται από τη λειτουργία [X]. Επέστρεψε σχέδιο υλοποίησης, κινδύνους, ερωτήσεις που εμποδίζουν ασφαλή εκτέλεση και ελέγχους αποδοχής. Μην γράψεις κώδικα ακόμη.

ΠΡΙΝ ΤΟ ΣΤΕΙΛΕΙΣ

- Έχει συγκεκριμένο outcome;
- Δείχνει τα σωστά paths ή evidence;
- Ορίζει όρια και προστατευμένα σημεία;
- Περιλαμβάνει verification και stop condition;

1.4 Case εφαρμογής

Μια μικρή εμπορική εταιρεία θέλει εφαρμογή outlet. Αντί για «φτιάξε app», η ομάδα ορίζει login, κατάλογο, φίλτρα, καλάθι, checkout, ιστορικό παραγγελιών, ρόλους, αποτυχίες πληρωμής και μετρήσιμα κριτήρια για κάθε ροή.

ΑΝΑΛΥΣΗ ΑΠΟΦΑΣΗΣ

- Τι γνωρίζουμε: η υπάρχουσα συμπεριφορά, οι χρήστες, τα συστήματα και τα διαθέσιμα tests.
- Τι πρέπει να επαληθευτεί: οι κρυφές εξαρτήσεις, τα edge cases και οι επιχειρησιακοί κανόνες που δεν είναι γραμμένοι.
- Τι δεν πρέπει να γίνει: επέκταση scope, διαγραφή δεδομένων ή αλλαγή production χωρίς evidence και rollback.
- Ποια είναι η απόδειξη: focused diff, passing checks, πραγματικό scenario και review των κινδύνων.

Η κρίσιμη διάκριση είναι ανάμεσα σε αλλαγή που απλώς φαίνεται σωστή και σε αλλαγή που είναι αποδεδειγμένα κατάλληλη για το συμφωνημένο περιβάλλον. Εδώ πρέπει να εξεταστούν ειδικά οι κίνδυνοι «Η λύση προηγείται του προβλήματος» και «Κρυφές απαιτήσεις εμφανίζονται μετά την υλοποίηση». Η αντιμετώπιση δεν είναι γενική προειδοποίηση: μετατρέπεται σε test, permission boundary, review step ή rollback ανάλογα με τη φύση του κινδύνου.

DECISION MEMO

Απόφαση: προχωράμε μόνο όταν τα artifacts «Problem statement μιας παραγράφου · User journey με happy path και failure paths · Feature specification» είναι διαθέσιμα και τα metrics «Ποσοστό acceptance criteria που περνά · Αριθμός αλλαγών score ανά κύκλο · Χρόνος από brief σε ελεγμένο vertical slice» μπορούν να μετρηθούν. Διαφορετικά, το task παραμένει discovery.

Το case δεν είναι συνταγή για αντιγραφή. Είναι παράδειγμα του τρόπου με τον οποίο χωρίζουμε πραγματικά στοιχεία, υποθέσεις, ελέγχους και ευθύνη.

1.5 Κίνδυνοι και quality gates

Οι παρακάτω κίνδυνοι δεν είναι θεωρητικές εξαιρέσεις. Είναι οι συνηθέστεροι τρόποι με τους οποίους μία φαινομενικά σωστή αλλαγή αποτυγχάνει σε review, staging ή production.

ΚΙΝΔΥΝΟΣ 1

Η λύση προηγείται του προβλήματος

ΚΙΝΔΥΝΟΣ 2

Κρυφές απαιτήσεις εμφανίζονται μετά την υλοποίηση

ΚΙΝΔΥΝΟΣ 3

Δεν υπάρχει κοινός ορισμός του ολοκληρωμένου

ΚΙΝΔΥΝΟΣ 4

Αλλάζει ταυτόχρονα προϊόν, design και υποδομή

QUALITY GATES

- Έχει παραχθεί: Problem statement μιας παραγράφου
- Υπάρχει test για την κύρια αποτυχία
- Το diff ελέγχθηκε για μη σχετικές αλλαγές
- Μετράται: Ποσοστό acceptance criteria που περνά
- Υπάρχει owner και rollback για production επίπτωση

1.6 Εργαστήριο

Γράψε specification δύο σελίδων για μία πραγματική λειτουργία: στόχος, χρήστης, πριν/μετά, ροή, δεδομένα, edge cases, μη λειτουργικές απαιτήσεις, acceptance criteria και εκτός scope.

ΥΠΟΧΡΕΩΤΙΚΗ ΠΑΡΑΔΟΣΗ

- Problem statement μιας παραγράφου
- User journey με happy path και failure paths
- Feature specification
- Evidence για: Ποσοστό acceptance criteria που περνά
- Σύνοψη αναφορά: τι επιβεβαιώθηκε, τι απέτυχε, τι παραμένει άγνωστο

ΠΡΩΤΟΚΟΛΛΟ ΕΚΤΕΛΕΣΗΣ

Χρόνος εργασίας: 60-90 λεπτά. Πρώτα κάνε read-only discovery και συγκέντρωσε evidence. Έπειτα δημιούργησε το μικρότερο παραδοτέο που αποδεικνύει την ιδέα. Εκτέλεσε τουλάχιστον μία κανονική και μία αποτυχημένη περίπτωση. Τέλος, μέτρησε «Ποσοστό acceptance criteria που περνά» και κατέγραψε αν το αποτέλεσμα δικαιολογεί επόμενο checkpoint.

ΚΡΙΤΗΡΙΟ ΕΠΙΤΥΧΙΑΣ

Το εργαστήριο θεωρείται ολοκληρωμένο μόνο όταν ένας δεύτερος άνθρωπος μπορεί να επαναλάβει τα βήματα, να δει τα ίδια στοιχεία και να καταλήξει στην ίδια απόφαση χωρίς προφορική καθοδήγηση.

STOP CONDITION

Σταμάτησε αν εμφανιστεί: η λύση προηγείται του προβλήματος ή αν δεν υπάρχει ασφαλής τρόπος να επαληθευτεί το αποτέλεσμα.

1.7 Checklist και αυτοαξιολόγηση

- ☐ Μπορώ να εξηγήσω με δικά μου λόγια: Να μετατρέψεις μια ασαφή επιχειρηματική ιδέα σε συγκεκριμένο πρόβλημα, χρήστες, ροές, δεδομένα, περιορισμούς και κριτήρια αποδοχής πριν ζητήσεις παραγωγή κώδικα.
- ☐ Έχω δημιουργήσει: Problem statement μιας παραγράφου και User journey με happy path και failure paths.
- ☐ Έχω ξεχωρίσει facts, assumptions, unknowns και απαγορευμένες ενέργειες.
- ☐ Έχω ελέγξει τον κίνδυνο: Η λύση προηγείται του προβλήματος.
- ☐ Μπορώ να μετρήσω: Ποσοστό acceptance criteria που περνά.
- ☐ Υπάρχει test ή χειροκίνητη διαδικασία που αποδεικνύει την κύρια ροή και την κύρια αποτυχία.
- ☐ Το τελικό diff είναι εντός score και δεν αντικαθιστά υπάρχουσες αλλαγές άλλου χρήστη.
- ☐ Η αναφορά παράδοσης δηλώνει καθαρά τι έγινε, τι ελέγχθηκε και ποιο ρίσκο παραμένει.

ΤΕΛΙΚΗ ΕΡΩΤΗΣΗ

Αν το αποτέλεσμα επηρεάσει πραγματικό πελάτη, δεδομένα ή χρήματα, ποια συγκεκριμένη απόδειξη θα δείξεις για να δικαιολογήσεις ότι ήταν ασφαλές να κυκλοφορήσει;

ΚΕΦΑΛΑΙΟ 02

Επιλογή περιβάλλοντος: desktop, CLI, IDE και cloud

Να επιλέγεις τη σωστή επιφάνεια Codex ανάλογα με το repository, τον τύπο εργασίας, την ανάγκη αλληλεπίδρασης, τη διάρκεια και το επίπεδο ελέγχου.

Η επαγγελματική αξία αυτού του κεφαλαίου βρίσκεται στη σύνδεση ανάμεσα στο τεχνικό έργο και στην επιχειρησιακή ευθύνη. Το ζητούμενο «Επιλογή περιβάλλοντος: desktop, CLI, IDE και cloud» δεν αντιμετωπίζεται ως μεμονωμένο tip. Εντάσσεται σε αλυσίδα αποφάσεων όπου το αποτέλεσμα πρέπει να είναι κατανοητό, αναπαραγώγιμο και ασφαλές. Για αυτό θα παράγεις συγκεκριμένα artifacts, θα ελέγξεις τους συνηθέστερους κινδύνους και θα συνδέσεις την ολοκλήρωση με μετρήσιμα στοιχεία.

ΚΕΝΤΡΙΚΗ ΑΡΧΗ

Δεν υπάρχει μία επιφάνεια για όλες τις εργασίες. Η CLI ταιριάζει στη στενή επαφή με terminal και scripts, το IDE στη συνεχή εργασία δίπλα στον κώδικα, το desktop στη διαχείριση project και πολλών ενεργειών και το cloud σε απομονωμένες, μεγαλύτερες αναθέσεις.

ΤΙ ΘΑ ΠΑΡΑΔΩΣΕΙΣ

- Πίνακας επιλογής περιβάλλοντος
- Project setup checklist
- Κατάλογος απαιτούμενων εργαλείων
- Πολιτική πρόσβασης
- Handoff note

Τα παραδοτέα αυτά δημιουργούν κοινή γλώσσα ανάμεσα σε ιδιοκτήτη προϊόντος, developer, reviewer και υπεύθυνο λειτουργίας. Αν κάποιος λείπει, κατέγραψε ρητά ποια απόφαση δεν μπορεί ακόμη να ληφθεί και ποια πληροφορία χρειάζεται για να κλείσει το κενό.

ΜΕΤΡΗΣΙΜΟ ΑΠΟΤΕΛΕΣΜΑ

Χρόνος setup ανά εργασία · Ποσοστό runs που ξεκινούν χωρίς missing dependency
· Αριθμός μη αναμενόμενων approval requests

2.1 Το σωστό νοητικό μοντέλο

Δεν υπάρχει μία επιφάνεια για όλες τις εργασίες. Η CLI ταιριάζει στη στενή επαφή με terminal και scripts, το IDE στη συνεχή εργασία δίπλα στον κώδικα, το desktop στη διαχείριση project και πολλών ενεργειών και το cloud σε απομονωμένες, μεγαλύτερες αναθέσεις.

Στο συγκεκριμένο κεφάλαιο, ο στόχος δεν είναι να απομνημονεύσεις ορολογία. Είναι να μπορείς να εξηγήσεις ποια απόφαση παίρνει ο άνθρωπος, ποια εργασία ανατίθεται στο Codex, ποια στοιχεία θεωρούνται πηγή αλήθειας και ποια απόδειξη θα σε κάνει να αποδεχθείς το αποτέλεσμα.

Η πρακτική εφαρμογή ξεκινά με το εξής ερώτημα: «Τι πρέπει να παραμένει αληθινό πριν και μετά την αλλαγή;». Από εκεί προκύπτουν τα όρια, τα tests και το score. Αν δεν μπορείς να απαντήσεις, το task χρειάζεται ακόμη discovery και όχι βιαστική παραγωγή κώδικα.

Για να γίνει η αρχή λειτουργική, σύνδεσέ την με τα παραδοτέα «Πίνακας επιλογής περιβάλλοντος» και «Project setup checklist». Το πρώτο εξηγεί τι επιχειρούμε να πετύχουμε· το δεύτερο δείχνει πού συναντά τον πραγματικό κώδικα και τη λειτουργία. Η απόσταση μεταξύ των δύο είναι το πεδίο όπου εμφανίζονται συνήθως οι κρυφές απαιτήσεις.

Έπειτα όρισε μέτρηση. Στο συγκεκριμένο θέμα, ένα χρήσιμο σήμα είναι «Χρόνος setup ανά εργασία». Το metric δεν αντικαθιστά το review, αλλά αναγκάζει την ομάδα να ορίσει τι θεωρεί βελτίωση και τι θα αποτελούσε λόγο επιστροφής στο σχέδιο.

ΠΑΡΑΔΕΙΓΜΑ

Για μικρή αλλαγή UI χρησιμοποιείται IDE με άμεσο preview. Για αναλυτικό έλεγχο 200 αρχείων χρησιμοποιείται cloud task σε ξεχωριστό worktree. Για migration βάσης δεδομένων επιλέγεται local περιβάλλον με περιορισμένα δικαιώματα και ρητές εγκρίσεις.

ΕΡΩΤΗΣΗ ΕΛΕΓΧΟΥ

Μπορεί ένας ανεξάρτητος reviewer να καταλάβει το πρόβλημα, το επιτρεπτό score και την απόδειξη επιτυχίας χωρίς να ζητήσει προφορικές διευκρινίσεις;

2.2 Workflow εφαρμογής

Η διαδρομή για το «Επιλογή περιβάλλοντος: desktop, CLI, IDE και cloud» αποτελείται από πέντε ελέγξιμα βήματα. Κάθε βήμα πρέπει να αφήνει ορατό παραδοτέο και να έχει συνθήκη ολοκλήρωσης. Αν ένα βήμα αποτύχει, δεν προχωράμε μηχανικά στο επόμενο.

01 · ΚΑΤΑΤΑΞΗ ΤΗΣ ΕΡΓΑΣΊΑΣ ΣΕ ΈΡΕΥΝΑ, ΑΛΛΑΓΗ, ΈΛΕΓΧΟ Ή ΑΥΤΟΜΑΤΟΠΟΊΗΣΗ

Κατέγραψε input, owner και επιθυμητό output πριν εκτελεστεί.

02 · ΈΛΕΓΧΟΣ ΠΡΟΣΒΑΣΗΣ ΣΕ ΑΡΧΕΊΑ, ΔΊΚΤΥΟ ΚΑΙ SECRETS

Σύνδεσε το βήμα με συγκεκριμένο path, command ή evidence.

03 · ΕΠΙΛΟΓΗ LOCAL Ή CLOUD EXECUTION

Έλεγε κανονική περίπτωση, εξαίρεση και αποτυχία.

04 · ΟΡΙΣΜΟΣ APPROVAL POLICY

Αποθήκευσε την απόδειξη ώστε να μπορεί να γίνει review.

05 · ΚΑΤΑΓΡΑΦΗ ΑΠΟΤΕΛΈΣΜΑΤΟΣ ΣΤΟ REPOSITORY

Αποφάσισε αν προχωράς, επανασχεδιάζεις ή σταματάς.

Κανόνας: καμία επιτυχία δεν θεωρείται πραγματική αν δεν μπορεί να επαναληφθεί στο συμφωνημένο περιβάλλον με τα ίδια verification steps.

2.3 Παραδοτέα και task brief

Τα παραδοτέα μετατρέπουν τη συζήτηση σε ελέγξιμη εργασία. Δεν χρειάζεται κάθε task να παράγει όλα τα παρακάτω, αλλά πρέπει να αφήνει αρκετό evidence ώστε ο επόμενος άνθρωπος να συνεχίσει χωρίς να ξανακάνει όλο το discovery.

- Πίνακας επιλογής περιβάλλοντος
- Project setup checklist
- Κατάλογος απαιτούμενων εργαλείων
- Πολιτική πρόσβασης
- Handoff note

REUSABLE PROMPT

Πριν ξεκινήσεις, έλεγξε το διαθέσιμο περιβάλλον, τις εντολές build/test, τα δικαιώματα και τα αρχεία οδηγιών. Δήλωσε τι λείπει και ποια είναι η ασφαλέστερη διαδρομή εκτέλεσης.

ΠΡΙΝ ΤΟ ΣΤΕΙΛΕΙΣ

- Έχει συγκεκριμένο outcome;
- Δείχνει τα σωστά paths ή evidence;
- Ορίζει όρια και προστατευμένα σημεία;
- Περιλαμβάνει verification και stop condition;

2.4 Case εφαρμογής

Για μικρή αλλαγή UI χρησιμοποιείται IDE με άμεσο preview. Για αναλυτικό έλεγχο 200 αρχείων χρησιμοποιείται cloud task σε ξεχωριστό worktree. Για migration βάσης δεδομένων επιλέγεται local περιβάλλον με περιορισμένα δικαιώματα και ρητές εγκρίσεις.

ΑΝΑΛΥΣΗ ΑΠΟΦΑΣΗΣ

- Τι γνωρίζουμε: η υπάρχουσα συμπεριφορά, οι χρήστες, τα συστήματα και τα διαθέσιμα tests.
- Τι πρέπει να επαληθευτεί: οι κρυφές εξαρτήσεις, τα edge cases και οι επιχειρησιακοί κανόνες που δεν είναι γραμμένοι.
- Τι δεν πρέπει να γίνει: επέκταση scope, διαγραφή δεδομένων ή αλλαγή production χωρίς evidence και rollback.
- Ποια είναι η απόδειξη: focused diff, passing checks, πραγματικό scenario και review των κινδύνων.

Η κρίσιμη διάκριση είναι ανάμεσα σε αλλαγή που απλώς φαίνεται σωστή και σε αλλαγή που είναι αποδεδειγμένα κατάλληλη για το συμφωνημένο περιβάλλον. Εδώ πρέπει να εξεταστούν ειδικά οι κίνδυνοι «Cloud task χωρίς επαρκές setup» και «Τοπική εργασία με υπερβολικά δικαιώματα». Η αντιμετώπιση δεν είναι γενική προειδοποίηση: μετατρέπεται σε test, permission boundary, review step ή rollback ανάλογα με τη φύση του κινδύνου.

DECISION MEMO

Απόφαση: προχωράμε μόνο όταν τα artifacts «Πίνακας επιλογής περιβάλλοντος · Project setup checklist · Κατάλογος απαιτούμενων εργαλείων» είναι διαθέσιμα και τα metrics «Χρόνος setup ανά εργασία · Ποσοστό runs που ξεκινούν χωρίς missing dependency · Αριθμός μη αναμενόμενων approval requests» μπορούν να μετρηθούν. Διαφορετικά, το task παραμένει discovery.

Το case δεν είναι συνταγή για αντιγραφή. Είναι παράδειγμα του τρόπου με τον οποίο χωρίζουμε πραγματικά στοιχεία, υποθέσεις, ελέγχους και ευθύνη.

2.5 Κίνδυνοι και quality gates

Οι παρακάτω κίνδυνοι δεν είναι θεωρητικές εξαιρέσεις. Είναι οι συνηθέστεροι τρόποι με τους οποίους μία φαινομενικά σωστή αλλαγή αποτυγχάνει σε review, staging ή production.

ΚΙΝΔΥΝΟΣ 1

Cloud task χωρίς επαρκές setup

ΚΙΝΔΥΝΟΣ 2

Τοπική εργασία με υπερβολικά δικαιώματα

ΚΙΝΔΥΝΟΣ 3

Διαφορετικές εκδόσεις εργαλείων

ΚΙΝΔΥΝΟΣ 4

Αποτελέσματα που δεν μεταφέρονται καθαρά στο κύριο branch

QUALITY GATES

- Έχει παραχθεί: Πίνακας επιλογής περιβάλλοντος
- Υπάρχει test για την κύρια αποτυχία
- Το diff ελέγχθηκε για μη σχετικές αλλαγές
- Μετράται: Χρόνος setup ανά εργασία
- Υπάρχει owner και rollback για production επίπτωση

2.6 Εργαστήριο

Πάρε πέντε πραγματικές εργασίες και επέλεξε επιφάνεια Codex για καθεμία. Τεκμηρίωσε γιατί, ποια πρόσβαση χρειάζεται, τι μπορεί να εκτελεστεί αυτόματα και ποια ενέργεια απαιτεί άνθρωπο.

ΥΠΟΧΡΕΩΤΙΚΗ ΠΑΡΑΔΟΣΗ

- Πίνακας επιλογής περιβάλλοντος
- Project setup checklist
- Κατάλογος απαιτούμενων εργαλείων
- Evidence για: Χρόνος setup ανά εργασία
- Σύνοψη αναφορά: τι επιβεβαιώθηκε, τι απέτυχε, τι παραμένει άγνωστο

ΠΡΩΤΟΚΟΛΛΟ ΕΚΤΕΛΕΣΗΣ

Χρόνος εργασίας: 60-90 λεπτά. Πρώτα κάνε read-only discovery και συγκέντρωσε evidence. Έπειτα δημιούργησε το μικρότερο παραδοτέο που αποδεικνύει την ιδέα. Εκτέλεσε τουλάχιστον μία κανονική και μία αποτυχημένη περίπτωση. Τέλος, μέτρησε «Χρόνος setup ανά εργασία» και κατέγραψε αν το αποτέλεσμα δικαιολογεί επόμενο checkpoint.

ΚΡΙΤΗΡΙΟ ΕΠΙΤΥΧΙΑΣ

Το εργαστήριο θεωρείται ολοκληρωμένο μόνο όταν ένας δεύτερος άνθρωπος μπορεί να επαναλάβει τα βήματα, να δει τα ίδια στοιχεία και να καταλήξει στην ίδια απόφαση χωρίς προφορική καθοδήγηση.

STOP CONDITION

Σταμάτησε αν εμφανιστεί: cloud task χωρίς επαρκές setup ή αν δεν υπάρχει ασφαλής τρόπος να επαληθευτεί το αποτέλεσμα.