

Luna 1 – Explorare & Fundamente

Obiectiv: Crearea culturii de AI în QA și alegerea unor procese clare unde AI aduce valoare.

- **[I] Săpt. 1–2: Training & awareness**

- Workshop intern: “Cum folosim AI în QA” (hands-on cu Playwright, Postman + ChatGPT/Copilot).
- Demonstrații: generare test cases din user stories, crearea de spec.ts pentru Playwright, auto-documentare test plan.

Plan detaliat sapt. 1 (setup):

- **Tool-uri:**
 - Playwright (JS/TS), instalat pe repo de test existent.
 - GitHub Actions (CI/CD).
 - OpenAI API / Copilot pentru generare cod.
- **Activități:**
 - Workshop hands-on: scriere test simplu de login în Playwright → apoi generare cu ChatGPT.
 - **Comparare:** test scris manual vs generat de AI.
- **Deliverable:** 1 test de login funcțional, validat de echipă.

Plan detaliat sapt. 2 (generare Test Cases)

- **Tool-uri:**
 - **Prompturi AI** → conversie din user stories (ex. JIRA / Confluence).
 - **Export .feature (Cucumber).**
- **Activități:**
 - QA scrie 2–3 user stories reale.
 - AI generează test cases → echipa le validează și corectează.
- **Deliverable:** 10 test cases generate și mapate în Playwright.
- **KPI:** Reducere timp de scriere cu min. 20%.

- **[II] Săpt. 3–4: Selectare procese pilot**

- **Alegi două zone clare:**
 - **Test design** → generare automată de test cases din cerințe.
 - **Test execution** → AI agent care rulează și compară loguri/UI.
- **Definire KPI inițiali:**
 - reducere cu 20% a timpului de creare test cases,
 - detectarea a min. 30% bug-uri regresive fără input manual.

Plan detaliat sapt. 3 – Regression Setup

- **Tool-uri:**
 - Playwright Test Runner.
 - pixelmatch (Node package pentru screenshot diff).

- **Activități:**
 - Setup screenshot baseline pentru 3 pagini.
 - Configurare AI care interpretează rezultatele (ChatGPT poate sumariza “ce e diferit”).
- **Deliverable:** 1 raport cu diferențele vizuale explicate în limbaj natural.

Plan detaliat sapt. 4 – Selectare Pilot Final

- **Tool-uri:** JIRA Dashboard.
 - **Activități:**
 - Alegerea 2 procese pilot:
 1. Generare test cases → Playwright.
 2. Regression automation cu AI diff.
 - **Deliverable:** Document “Pilot Scope” + KPI măsurați.
-

Luna 2 – Pilotare & Integrare

Obiectiv: Implementarea AI în task-uri repetitive și măsurarea impactului.

- **Săpt. 5–6: Test case generation**

- Integrare AI pentru conversie: user stories → test scenarios → .feature files → Playwright specs.
- QA-ul uman validează → înveți modelul pe feedback.

Plan detaliat Săpt. 5–6 – AI Test Case Generator

- **Tool-uri:**

- **Script custom:** JIRA API → extrage stories → trimite la OpenAI → primește .feature file.
- **Playwright** + Cucumber adaptor.

- **Activități:**

- **Rulezi pipeline:** story nou → test file generat → review uman.
- QA validează și marchează diferențele.

- **Deliverable:** 1 pipeline semi-automat “story → test”.

- **KPI:** Timp redus cu 25%.

- **Săpt. 7–8: Regression automation cu AI agent**

- **AI compară:**

- screenshots UI între versiuni,
- răspunsuri API (JSON diff),
- loguri și metrice de performanță.

- Se creează rapoarte auto-sumarizate pentru echipă.
- **Măsurare:** câte bug-uri regresive prinde AI vs QA manual !!??

Plan detaliat Săpt. 7–8 – Regression cu AI Agent

- **Tool-uri:**

- Playwright + pixelmatch.
- JSON diff (json-diff NPM).
- ChatGPT API → sumarizare loguri.

- **Activități:**

- Rulezi regresii săptămânale pe UI și API.
- AI produce raport sumarizat: “3 pagini afectate, diferențe de layout minore, 1 API răspuns schimbat”.

- **Deliverable:** Raport automat generat și trimis pe Slack/Teams.

- **KPI:** AI prinde >30% din regresii fără input manual.

Luna 3 – Optimizare & Scalare

Obiectiv: Crearea unui framework scalabil și justificarea ROI pentru extindere.

- **[I] Sapt. 9–10: Knowledge sharing**

- **AI agent** → auto-sumarizează test runs și scrie raport QA săptămânal.
- QA-urile fac knowledge sharing: ce prompturi, ce rezultate, unde a fost nevoie de corecții.

Plan detaliat sapt. 9 – AI Knowledge Sharing

- **Tool-uri:**
 - GitHub Wiki / Confluence.
 - ChatGPT API → auto-sumarizare test runs.
- **Activități:**
 - AI scrie rapoarte săptămânale: “Ce teste au rulat, ce bug-uri au fost găsite”.
 - QA validează și postează pe wiki.
- **Deliverable:** 2 rapoarte QA generate complet de AI.

Plan detaliat sapt. 10 – Performance Pilot

- **Tool-uri:**
 - k6 (performance testing).
 - AI → analizează grafice și extrage insight-uri.
- **Activități:**
 - Rulezi un test de load (ex: 1000 req/s).
 - AI detectează bottleneck și scrie sumar pentru echipă.
- **Deliverable:** Raport performance cu concluzii AI.

- **[II] Sapt. 11: Analiză KPI**

- Timp economisit pe test design & execution.
- Număr bug-uri detectate de AI vs manual.
- Costuri evitate (ex: externalizare testare regresivă).

Plan detaliat sapt. 11 – KPI Analysis

- **Tool-uri:**
 - JIRA Dashboard + Confluence report.
- **Activități:**
 - Compari timpul de scriere test cases, bug-uri detectate de AI vs manual.
 - Calculezi ore economisite.
- **Deliverable:** Raport ROI (time saved, bugs caught).

- **[II] Sapt. 12: Roadmap next phase**

- **Dacă ROI este pozitiv, extinzi către:**
 - performance testing asistat de AI,
 - testare securitate (scan automată + validare QA),
 - integrare AI în CI/CD pipelines.

Plan detaliat sapt. 12 – Roadmap Next Phase

- **Tool-uri:** Workshop intern.
- **Activități:**
 - Decizi ce scalezi:
 - Integrare AI și pe testare securitate,
 - Extindere regresie pe mobile,
 - Deployment AI agent direct în CI/CD.
- **Deliverable:** Roadmap 6 luni.

Rezultate așteptate după 3 luni

- Test case generation cu reducere cu **20–40% a timpului de scriere test cases.**
- Detectare **automată a regresiilor** în build-uri săptămânale
- Rapoarte QA auto-sumarizate → QA câștigă timp pentru testare exploratorie.
- O **echipă QA care știe să folosească AI practic**, nu doar teoretic.
- **Performance bottlenecks** detectate mai rapid.
- Raport ROI → justifică buget pentru AI extins.