BADANIE AI TESTING LAB, EDYCJA DRUGA

BOOKLET OPERATORA AI

15 zasad skutecznej pracy z modelem w testach

PODSTAWY WNIOSKÓW

Każda z zasad na tej liście odpowiada konkretnemu wzorcowi zaobserwowanemu w repozytoriach uczestników badania Sii Testing Lab. Szesnaście zespołów pracujących z narzędziami AI szukało dziewiętnastu celowo zasianych defektów w systemie do obsługi wniosków kredytowych. Sprawdziliśmy około 350 raportów błędów, konfrontując każdy z kodem, dokumentacją uczestnika i specyfikacją OpenAPI. Typowy zespół wykrył około sześciu z dziewiętnastu defektów, a dwóch najdroższych produkcyjnie nie wykrył nikt. Warsztat inżynierski był przy tym wyrównany, a narzędzia te same, więc różnice wynikały ze sposobu pracy. Praktyki, które powtarzały się w skutecznych zespołach i których brakowało u pozostałych, złożyły się na tę listę.

JAK KORZYSTAĆ Z BOOKLETU

Lista ma trzy zastosowania: jako punkty kontrolne w definicji ukończenia zadania testowego, jako pytania w przeglądzie kodu oraz jako materiał wdrożeniowy dla osoby, która zaczyna pracować z modelem. Zasady są niezależne od stacku technologicznego. Pierwsze jedenaście pogrupowaliśmy według pięciu bramek jakości opisanych w raporcie, a cztery ostatnie zbierają praktyki obowiązujące przez cały czas pracy.

BRAMKA 1

Niezależna wyrocznia poprawności

Oczekiwany wynik liczony ze specyfikacji, nigdy z zachowania aplikacji.

1. Buduj niezależną wyrocznię ze specyfikacji, nie na podstawie samego zachowania aplikacji.
2. Jeśli pełna formuła jest zbyt droga, użyj metamorficznego niezmiennika (metamorphic invariant).

BRAMKA 2

Asercje na właściwej warstwie

Oczekiwany wynik liczony ze specyfikacji, nigdy z zachowania aplikacji.

3. Dla wartości biznesowych dokonuj asercji przez API, nie tylko przez UI.

BRAMKA 3

Osobny przebieg adwersaryjny

Pokrycie potwierdza działanie. Wykrywanie błędów wymaga osobnego przebiegu.

4. Po podstawowym pokryciu uruchom osobny adversarial pass.
5. Zamieniaj każde ryzyko w konkretny scenariusz, który może udowodnić błąd.
6. Pisz testy tak, żeby mogły obalić system, a nie tylko potwierdzić happy path.

BRAMKA 4

Higiena odpowiedzi i bezpieczeństwo

Przegląd niezależny od testów domenowych, najtańszy do wdrożenia.

7. Zrób krótki security oraz response-hygiene sweep: pola wrażliwe w odpowiedziach, autoryzacja endpointów, spójność CORS, walidacja śmieci, zakres widoczności danych między rolami.

BRAMKA 5

Zweryfikuj, zanim uwierzysz

Zielony wynik jest twierdzeniem do sfalsyfikowania.

8. Nie pozwalaj AI osłabiać asercji, żeby test był zielony.
9. Każdy bug powinien mieć red-by-design repro albo test oczekiwanego niepowodzenia.
10. Traktuj czerwony test jako hipotezę o błędzie, zamiast automatycznie klasyfikować go jako flaky.
11. Weryfikuj ponownie wyniki z pomocą osobnego agenta, w osobnej sesji albo z udziałem człowieka.

PRAKTYKI PRZEKROJOWE

Dyscyplina pracy i pomiar

Obowiązują niezależnie od tego, którą bramkę właśnie przechodzisz.

12. Trzymaj stały kontekst dla AI w pliku typu CLAUDE.md, AGENTS.md albo MEMORY.md.
13. Zapisuj decyzje: co AI zaproponowało, co odrzuciłeś i dlaczego.
14. Nie buduj frameworka kosztem bug huntingu.
15. Nie używaj liczby testów jako głównej miary sukcesu.

Zanim uznasz test za gotowy, odwróć oczekiwaną wartość i uruchom go ponownie. Test, który przy błędnej wartości nadal przechodzi, niczego nie pilnuje. To najszybszy sposób, żeby sprawdzić, czy asercja ma zęby.

NA CZYM OPARLIŚMY TE WNIOSKI?

Pełne badanie wraz z danymi, wykresami i mechanizmami stojącymi za każdą bramką opisaliśmy w raporcie „Badanie AI Testing Lab, część II: AI nie wystarczy". Znajdziesz tam między innymi krzywą ryzyka pokazującą, że im droższy produkcyjnie defekt, tym rzadziej był wykrywany, oraz porównanie kosztu zgłoszeń, w którym ten sam wynik kosztował siedem razy więcej albo siedem razy mniej, zależnie od sposobu pracy.

Ile z Waszych „zielonych" testów **wykryłoby błąd, który trafi do produkcji?**

Przeprowadzimy analizę na Waszej aplikacji z wykorzystaniem tej samej metodologii, którą zastosowaliśmy w badaniu AI Testing Lab. Wynik pokaże, które obszary ryzyka są skutecznie zabezpieczone przez testy, a gdzie istnieją luki wymagające uwagi.

Skontaktuj się z ekspertami

Iuliia Toporova
Business Development Manager