Чекліст перед запуском AI-агентів

Що тихо розвалюється в проді

AI-агенти ламаються інакше, ніж звичайний софт. Тихо: тест зелений, демо працює, захисний крок існує на папері, і все виглядає добре, поки не перестає.

Кожен пункт нижче я спіймав, коли будував і доводив власну агентну платформу в травні й червні 2026. У кожного є дата, що сталося і перевірка, яку можна зробити на своїй системі цього тижня.

01 Зелений end-to-end тест може ховати мертву функцію.

Тест може перевіряти просту внутрішню логіку і пропускати з'єднання з продом. Усі перевірки зелені, а головна функція для реального користувача не спрацьовує.

СПІЙМАВ 4 червня 2026. Набір із 7 end-to-end тестів був зеленим, а кнопка «Схвалити / Відхилити» ніколи не доходила до оператора. Знайшло рев'ю коду перед мерджем.

ПЕРЕВІР Видали з'єднання між двома компонентами в проді. Якщо жоден тест не почервонів, тести декоративні.

02 Захисний крок може тихо пропускати все.

Крок, який мав блокувати ризиковані дії, може бути зроблений так, що через відсутню залежність він пропускає дію замість зупинки. Ззовні це виглядає як «працює».

СПІЙМАВ 4 червня 2026. У прод-збірці не було модуля правил, перевірка виходила раніше, і рішення «Відхилити» все одно виконувалось.

ПЕРЕВІР На реальній системі доведи, що відмова справді зупиняє дію.

03 «Автоматичне відновлення» може означати відновлення лише після акуратних збоїв.

Тести відновлення часто імітують збій кинутим винятком. Справжнє вбивство процесу (падіння, нестача пам'яті, вимкнене живлення) все одно може загубити роботу в процесі.

СПІЙМАВ 20 червня 2026. Після кинутого винятку робота відновлювалась правильно, а справжній SIGKILL губив незбережений результат, і всі побічні дії виконувались ще раз.

ПЕРЕВІР Вбий процес по-справжньому, а не кидай помилку всередині нього.

04 Налаштований таймаут може нічого не обмежувати.

Деякі бібліотеки приймають налаштування таймауту і за певних умов його ігнорують. Виклик, який ти вважаєш обмеженим, може висіти вічно і заклинити всю систему.

СПІЙМАВ 27 травня 2026. Клієнтська бібліотека для LLM прийняла timeout=4 і висіла понад 35 секунд на endpoint, який не відповідав. Завмирало близько 40% запусків.

ПЕРЕВІР Направ виклик на сервіс, який приймає з'єднання і ніколи не відповідає.

05 Один завислий зовнішній виклик може заморозити весь багатокроковий запуск.

Якщо кроки не ізольовані, одна залежність, що не відповідає, заморожує всю роботу замість того, щоб упав лише цей шматок.

СПІЙМАВ Травень 2026. Один завислий виклик моделі заморожував 12-хвилинні запуски, поки їх не вбивав watchdog. Під навантаженням близько третини викликів до хостованої моделі впирались у ліміт 180 секунд.

ПЕРЕВІР Чи може один крок упасти, повторитись або бути пропущеним без втрати вже зробленої роботи?

06 Дрібні розбіжності в даних тихо ламають те, що зберігається чи знаходиться.

Розбіжність часових поясів, неочікувана обгортка навколо відповіді моделі або не та метрика схожості змушують запис тихо не відбутися, а пошук повертати не те. Без жодної помилки.

СПІЙМАВ 25 травня 2026. Колонка з часовим поясом проти колонки без нього тихо пропускала записи пам'яті. Відстань L2 замість косинусної зламала пошук. Код, загорнутий у markdown-огорожі, валив тести і виглядав як проблема якості.

ПЕРЕВІР Подивись, що реально збереглось і знайшлось, а не лише на те, що «винятків не було».

07 Тест можна збудувати так, що він не може впасти.

Тест на стійкість можна побудувати так, що збій, який він мав ловити, недосяжний. Тоді зелений результат нічого не означає.

СПІЙМАВ 30 травня 2026. Перевірка стійкості до шуму була зеленою, бо набір відволікаючих прикладів наситився, і збій, який вона мала ловити, не міг статися.

ПЕРЕВІР Перш ніж довіряти тесту, змусь його один раз показати поганий результат.

08 Той самий код поводиться по-різному залежно від того, де він працює.

Залежність може змінювати свій вивід залежно від змінних середовища, які вона бачить. Перевірка, що проходить на ноутбуці, може впасти або тихо поводитись інакше в CI чи проді.

СПІЙМАВ 29 червня 2026. Тест проходив локально і падав у CI, бо pytest змінює свій вивід, коли бачить CI (CI=true).

ПЕРЕВІР Порівняй реальний вивід у різних середовищах, перш ніж називати тест нестабільним.

09 Більше знайденого контексту може зробити систему гіршою.

Додаткові приклади чи пам'ять не завжди допомагають. Вони можуть відволікти модель від задачі, яку вона розв'язала б сама.

СПІЙМАВ 29 травня 2026. На одному типі задач знайдена пам'ять знизила частку успіхів з 1,00 до 0,33. Підказка відволікла модель від простої задачі.

ПЕРЕВІР Перевір, чи доданий контекст не шкодить задачам, з якими система вже справляється.

10 Коли одна модель пише задачу, правильну відповідь і оцінку, вона впевнено помиляється.

Коли та сама модель або модель з тієї ж родини пише тест, очікувану відповідь і оцінювання, вона не погоджується сама з собою на граничних випадках.

СПІЙМАВ 27 травня 2026. Згенерована моделлю повторна перевірка мала інші вимоги до граничних випадків і завалила правильне рішення.

ПЕРЕВІР Прив'язуй оцінку до еталону, перевіреного незалежно.

11 Коли той самий тип виправлення повертається, перестань латати і поміряй.

Повторення схожих виправлень (підняти таймаут, підняти ще раз) спалює години, поки справжня причина лежить на шар глибше.

СПІЙМАВ 27 травня 2026. Три хибні здогадки і кілька змін таймауту, перш ніж дамп стеку показав читання з сокета взагалі без таймауту.

ПЕРЕВІР Після двох невдалих виправлень одного типу збери реальні логи, трейси і мінімальне відтворення.

Нічого екзотичного. Звичайна інженерна дисципліна для систем, які ламаються тихо, а не голосно. Тому це легко пропустити, поки про це не скаже прод.

Потрібна друга пара очей на твою AI-систему?

Експрес-аудит однієї системи: 3 робочі дні, від $1 500. Перший дзвінок на 30 хвилин безкоштовний.

rostyslav.stekh@gmail.com · linkedin.com/in/rostyslav-stekh