Engineer

Тестування та QA

44 досягнення

Поєднайте частини системи

Роботи за темою «Тестування та QA» показують одну частину дієвої системи. Ми можемо допомогти пов'язати її з рештою вашої діяльності. Та частина, якої користувач не бачить: де зберігаються дані, які діють правила і як інші системи можуть із нею взаємодіяти.

Переглянути наші послуги Зв'язатися з нами →

Знайдіть схоже завдання

Відкрийте приклад, схожий на вашу ситуацію. Огляд пояснює рішення, що привели до результату, а не лише називає технологію. Спроєктувала й підтримувала 6 пайплайнів ETL/ELT — 6 пайплайнів ETL на трьох рушіях баз даних

Спроєктувала й підтримувала 6 пайплайнів ETL/ELT Зв'язатися з нами →

Оптимізувала CI/CD, заощадивши 4 000 годин

Оптимізувала процеси CI/CD, заощадивши 4 000 годин завдяки впровадженню автоматизації в пайплайнах розробки програмного забезпечення.

Перенесла API на Huma v2 зі 100% відповідністю

Перенесла HTTP API з Fiber на Huma v2 — 649 шляхів і 760 операцій — досягнувши та втримавши 100% відповідність між маршрутами, які реєструє сервер, і описом OpenAPI, який він публікує.

Згенерувала контракт API із самої бази даних

Згенерувала контракт API з бази даних назовні — OpenAPI, типізований TypeScript-клієнт на 44 076 рядків, 61 mock-обробник та обмеження, які застосовує UI, — з перевіркою на кожній ланці, що падає в разі розбіжностей.

Побудувала пайплайни CI/CD на GitHub Actions

Побудувала пайплайни CI/CD на GitHub Actions з distroless-образом продакшн-фронтенду та просуванням релізів між кількома середовищами.

Написала 578 цілей автоматизації go-task

Написала 578 цілей автоматизації go-task, що охоплюють нативний, Docker- та HTTPS-режими розробки, лінтинг, тестування, базу даних та розгортання.

Запровадила нуль попереджень у шести мовах

Встановила стандарт якості без жодних попереджень для шести мов — Go, TypeScript, SQL, Python, Shell і Markdown — дотримання якого забезпечували pre-commit hooks.

Побудувала тести для Go, SQL і браузера

Побудувала багаторівневий набір автоматизованих тестів — 981 тест Go, 543 фронтендні та браузерні специфікації, 494 поведінкові тести SQL — з мутаційним тестуванням, property-based тестами та обов'язковою перевіркою доступності.

Побудувала механізм із 268 перевірок репозиторію

Побудувала механізм перевірок репозиторію — 268 зареєстрованих перевірок коміту, 277 правил лінтингу та 15 власних правил ESLint — а також 146 тестів самих перевірок, щоб стандарт тримала збірка, а не код-рев'ю.

Тримала схему узгодженою впродовж 1 022 міграцій

Тримала схему узгодженою впродовж 1 022 міграцій за допомогою перевірки CI, яка збирає базу даних обома шляхами — чиста інсталяція та інсталяція плюс усі міграції — і падає, коли вони розходяться.

Посилила SSH до 24 перевірених директив

Посилила захист SSH: 24 директиви, кожну з яких перевіряє assert, і триетапна валідація — файл-кандидат, зібрана конфігурація, а потім зворотне зчитування самим демоном, — після того як таке зчитування виявило, що запущений сервер мовчки перевизначав дві з двадцяти чотирьох.

Довела, що блокування вторгнень працює щоразу

Довела працездатність усього ланцюжка блокування вторгнень під час кожного запуску посилення захисту: блокувала зарезервовану тестову адресу, зчитувала створене правило ядра й знімала блокування в гарантованому блоці очищення, тож jail, який перестав працювати, обриває запуск помилкою, а не звітує про справність.

Перевірила правила фаєрвола за позицією

Перевірила правила фаєрвола за позицією, а не лише за наявністю, зчитуючи пронумерований перелік правил і чинний ланцюжок фільтрації пакетів, бо наявне правило ще не означає, що до нього доходить хоч один пакет.

Домоглася, щоб check mode в Ansible казав правду

Домоглася, щоб check mode показував правду на всій платформі, після того як знайшла шість перевірок, що ухвалювали рішення за значенням, якого хост ніколи не повертав: модуль command в Ansible у режимі --check звітує про успіх, узагалі не виконуючи команду.

Додала секундну попередню перевірку секретів

Додала попередній плей (preflight), який приблизно за секунду проганяє той самий код, що й converge, на локальних секретах оператора, — після того як продакшн-запуск обірвався на дев'ятому завданні, застосувавшись лише наполовину, а налаштування swap уже було записано на робочий хост.

Посилила ізоляцію systemd для кожного юніта

Знизила оцінку відкритості (exposure) у пісочниці systemd для кожного юніта, який встановлює платформа: служба dead man's switch — з 9,6 UNSAFE до 1,5, відкритий в інтернет git-сервер — з 8,3 EXPOSED до 1,5, — і додала перевірку парсером під час converge, після того як знайшла у трьох шаблонах юнітів директиву з орфографічною помилкою, яку systemd мовчки ігнорував.

Побудувала перевірку комітів із 22 лінтерів

Побудувала перевірку перед комітом із 22 лінтерів, кожен з яких описується одним рядком, і ще п'ятьох, що заслуговують окремого абзацу, без рівня попереджень і без дозволу на вбудоване вимкнення правил, яка охоплює HTML, CSS, JavaScript, Python, YAML, Markdown, shell, посилання, орфографію, секрети й типографіку.

Страховка, що більше не забирає пів години

Скоротила браузерну перевірку якості сайту з 1 636 секунд до 615, запускаючи перевірки від найдовших до найкоротших через пул воркерів із максимум чотирма одночасними перевірками, після того як виміряла, що алфавітний порядок коштував 320 секунд проти 224.

Замінила 27 розмірів шрифту шкалою з 6 кроків

Замінила 27 фактичних розмірів шрифту, сусідні значення яких різнилися лише на 0,6%, шестиступеневою шкалою Major Third і написала лінтер, який відхиляє двадцять восьмий.

Перевірки доступності 32 сторінок в обох темах

Реалізувала перевірки WCAG 2.2 рівня AA на 32 репрезентативних сторінках в обох колірних темах із записом відповідності, перевіркою одного критерію AAA та явним зазначенням часткового результату для критерію 1.4.13.

Вади доступності, яких не бачить ваша автоматична перевірка

Знайшла й виправила 11 дефектів доступності, які браузер вважав справними: вісім посилань у підвалі, що лишалися в порядку табуляції попри вимкнену взаємодію через pointer-events, часову шкалу прокрутки (scroll timeline), яка обрізала колофон на трьох сторінках, і рушій правил, що виконував лише 70 зі своїх 105 правил.

Обмежила палітру сайту 28 кольорами

Обмежила систему кольорів 28 задокументованими кольорами й написала лінтер, який відхиляє значення, що застосоване, але не задокументоване, задокументоване, але не застосоване, хибно виміряне, переписане як літерал або розташоване в межах перцептивної відстані 0,02 від уже наявного.

Знайшла, що світлій темі бракує цілого шару

Виявила, що у світлій темі від самого її створення бракувало повносторінкового шару накладання, додавши перевірку, яка вимагає, щоб обидві теми малювали кожну багатошарову поверхню однаковою кількістю шарів.

Домоглася, щоб режим зменшеного руху працював

Домоглася, щоб режим зменшеного руху справді працював, після того як знайшла одинадцять селекторів, які й далі анімувалися за цього налаштування, бо універсальне правило transition-none поступається за специфічністю будь-якому правилу з класом.

Виправила мапу сайту з однією позначкою часу

Виправила мапу сайту, де 172 з 176 URL мали одну позначку часу зміни, взявши дату з історії git, після того як з'ясувала, що експорт переписує кожен файл під час кожного запуску.

Підключила до документації окремий лінтер

Підключила до документації окремий лінтер — ліміт рядків для кожного файлу, який можна лише знижувати, вимогу індексу та перевірку справжніх шляхів, — який знайшов шість із дев'яти планів, позначених виконаними, попри невідмічені пункти, і чотирнадцять завдань перевірки перед комітом, не згаданих у жодному документі.

Перевірила лінтером 10 242 рядки перевірки якості

Охопила форматувальником і лінтером 10 242 рядки JavaScript у перевірках якості, з'ясувавши, що це найбільший обсяг коду в репозиторії і єдиний, який ніщо не перевіряло, — виправила 13 зауважень і не вимкнула жодного.

Забезпечила генератору 981 тест і 92% покриття

Покрила генератор 981 тестовим випадком із мінімальним порогом покриття гілок 92%, де попередження вважаються помилками, і перевірила ідемпотентність, двічі запустивши весь пайплайн збірки з порожнього файлу та вимагаючи, щоб другий прохід нічого не змінював.

PDF/UA-1: пройдено 106 зі 106 перевірок профілю

Зафіксувала перевірку PDF/UA-1 для 9 документів: успішно пройдено 106 зі 106 налаштованих перевірок профілю доступності, тоді як архівний варіант не пройшов одне зі 146 правил.

Увімкнула всі правила лінтера Python як помилки

Увімкнула всі правила лінтера Python як помилки й довела 1 815 зауважень до нуля; кожен із небагатьох винятків має письмове обґрунтування, а два з них підкріплені перевіркою, а не коментарем.

Вбудувала кодувальник QR на 522 рядки

Перенесла в репозиторій власний кодувальник QR — код Ріда-Соломона над GF(256), фіксоване розташування модулів, вісім шаблонів маски — на 522 рядках, замість додавати третю залежність середовища виконання, і перевірила його, зчитавши готову матрицю незалежно написаним декодером.

Написала тести для самих перевірок

Написала тести для самих перевірок, з'ясувавши, що перевірка, яка отримує лише чисті вхідні дані, одного дня звітуватиме про чистоту, бо нічого не прочитала: підклала помилку правопису, щоб підтвердити, що перевірка орфографії її знаходить, і взяла діапазон ідентифікаторів із бази даних, а не з числа в тесті.

Виявила розбіжність між хуками й гейтом якості

Виявила, що хуки комітів і гейт якості виконують різні перевірки, хоча документ обіцяв, що вони однакові, порівнявши два переліки в тесті, — п'ять перевірок, які запускалися лише вручну, були саме тими, що читали текст CV.

Перевірила CI-хук git-сервера, знайшла 2 дефекти

Відрепетирувала CI-хук на боці git-сервера й знайшла два дефекти, яких не виявити читанням файлу: запасний варіант, що передавав на вхід хука аргумент, який неможливо розв'язати, і власну змінну середовища git, яка потрапляла слідом за гейтом у checkout і робила 19 тестів червоними.

Побудувала міжмовну перевірку чисел

Побудувала міжмовну перевірку контенту, яка падає, коли переклад губить число, вказане англійською, або коли мова вживає не притаманну їй нотацію, — і знайшла два данські описи без метрики та шістнадцять українських фрагментів, що цитували на англійський лад.

Обмежений потік подій у застосунку macOS

Обмежила буфери подій і журналювання після зафіксованого інциденту macOS зі зростанням на 41 МБ за секунду й 111 ГБ стиснених сторінок на машині з 36 ГБ; зафіксований повний тестовий запуск зменшив журнал ядра з 610 996 до 1 411 рядків.

Увімкнула сувору конкурентність Swift 6 поверх C

Перейшла на повну сувору перевірку конкурентності Swift 6 без акторів, з'єднавши блокувальний цикл подій C з головним актором через одного виробника, одного споживача й один порядок, — з'ясувавши, що окрема задача на кожну подію губить порядок, від якого залежить інтерфейс.

Написала парсер, що перевіряє кожен виклик FFI

Написала парсер, який читає справжній C-заголовок на 7 308 рядків і перевіряє кожне місце виклику, кожну константу переліку й те, що кожен клас-власник вказівника є final, після того як через написаний вручну заголовок-заглушку виклики трьох вилучених функцій компілювалися, лінкувалися й аварійно завершувалися.

Пришвидшила набір тестів до 8,9 секунди

Скоротила набір тестів від шести тестів, що перевищували ліміт у шістдесят секунд, до 135 успішних за 8,9 секунди, профілюючи головний потік і прибравши два виклики, у яких він перебував 3 989 із 4 017 вибірок.

Назвала всі 17 піктограм для зчитувачів екрана

Дала назви для зчитувачів екрана всім елементам керування, що складаються лише з піктограми, після того як виявила, що кнопка надсилання оголошувалася як «arrow up circle, button», і написала лінтер, який вимагає підпису в межах восьми рядків від піктограми.

Побудувала дизайн-систему macOS із 48 токенів

Побудувала дизайн-систему з 48 токенів на основі власного скляного матеріалу платформи й написала лінтер, який відхиляє магічне число, жорстко заданий розмір шрифту чи анімацію, що ігнорує налаштування зменшеного руху.

Задіяла невикористану половину ядра

Задіяла ту половину ядра обміну повідомленнями, якою застосунок ніколи не користувався, — передавання резервної копії, зникомі повідомлення, редагування та повторне надсилання, підтверджені запрошення, проксі та політику шифрування, — проганяючи кожен тест на справжній бібліотеці без заглушок.

Переписала помилки, щоб не винити користувачів

Переписала повідомлення про помилки застосунку за письмовим стандартом тону після того, як у разі відмови у вході застосунок звинувачував користувача в помилці введення, хоча провайдер насправді вимагав пароль для застосунку, і покрила це тестом, який називає провайдера.

Створила айдентику з 2 малюнків

Створила набори піктограм і фавіконів компанії на основі двох намальованих вручну зображень; кожен похідний ресурс перегенеровується скриптом, а перевірка падає, якщо похідний файл закомічено раніше за його джерело.

Змусила шість збирачів звітувати про пропущене

Змусила шість збирачів звітів повідомляти ту частину свого входу, яку вони так і не прочитали, — файли, що лишилися закритими, невстановлені зонди, нерозпізнані рядки журналу, — і підігнала пороги до 116 виміряних зразків, з'ясувавши, що сліпий збір і справний хост дають однакову сторінку.

Спрямуйте інженерні зусилля

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Розгляньте тему у своєму контексті

Розкажіть, як «Тестування та QA» стосується вашої системи. Ми допоможемо обрати відповідну послугу та визначити обсяг роботи.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/categories/testing/