

Kararsız testler: %0,5'ten %98'e

800 testin her biri %0,5 ihtimalle boşuna patlıyorsa süit neredeyse her koşulda kırmızı. Retry neden tedavi değil, kararsızlığı ne yakalar?

01 SÜİT BÜYÜDÜKÇE DAHA AZ GÜVENİLİR

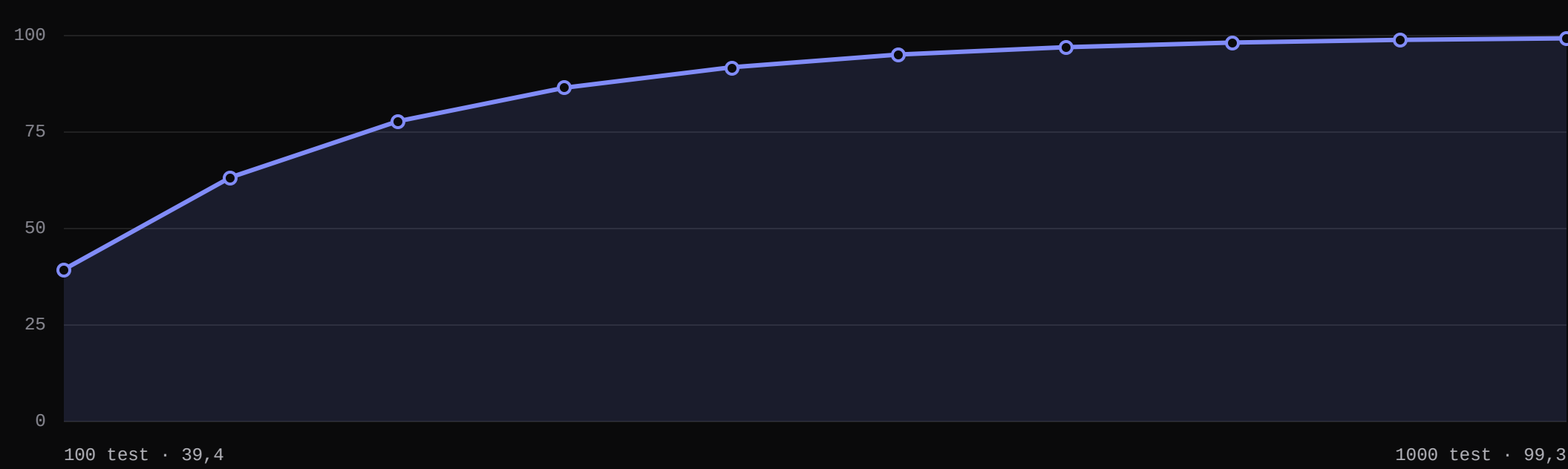
%98

Temiz bir çalıştırmada herhangi bir testin patlama ihtimali

%0,5 ihtimalle boşuna patlayan tek bir test görünmez. Aynısından 800 tane koy: $1 - 0,995^{800} \approx \%98$.

02 İHTİMALEER BİRİKİYOR

Test sayısı arttıkça yanlış kırmızı



Formülden hesaplandı: $1 - 0,995^n$. Ölçüm değil; yazıda yalnız $n = 800$ için $\approx \%98$ geçer.

03 PATLAMALAR BAĞIMSIZ DA DEĞİL

Yerelde 50 ms, yük altında timeout.

CI, paylaşımlı ve throttle'lı donanımda koşar; laptop'unda 50 ms'de biten async test, yük altında timeout'a düşer.

04 AI KARARSIZLIĞI İCAT ETMEDİ, SERİ ÜRETIME GEÇİRDİ

Doğru görünen dört kalıp



Sabit bir sleep

Async işi "beklemek" için: `time.Sleep(2 * time.Second)`, `await page.WaitForTimeout(2000)`, `cy.wait(2000)`.



Seed'siz rastgelelik

Her çalıştırmada farklı değer: `Math.random()`, `rand.Intn(100)`, sabit seed'i olmayan bir faker.



Kırılgan bir selector

DOM yapısına çakılı: `nth-child`, otomatik üretilmiş bir class adı. Markup azıcık kayınca kırılır.



Her şeye bir mock

Test bir sonucu değil bir etkileşimi doğrular.

05 RETRY BİR KOŞU BANDI, TEDAVİ DEĞİL

Hepsi reaktif



Retry

Gerçek CI dakikası yakar ve feedback loop'u uzatır; yeni hiçbir şey öğrenmemek için iki kez ödersin.



Quarantine

Sessizce büyüyen bir devre dışı test yığını. O yığın borçtur ve faiziyle büyür.



Ticket

Backlog gürültüsüne yaşanır ve sessizce önemini yitirir.

06 PATLAMAYI DEĞİL, ANTI-PATTERN'I YAKALA

Ne zaman devreye giriyor?

	Retry, quarantine, ticket	Commit sınırında statik tarama
ZAMAN	Test CI'ı zaten bozduktan sonra	✓ Test daha çalışmadan, diff'te
ETKİ	Belirtiyi gizler, bedelini sana keser	✓ Bilinen anti-pattern'leri işaretler
TEKRARLANABİLİRLİK	Yeniden çalıştırınca geçer, sonra yine patlar	✓ Aynı diff, aynı sonuç; model yok, network yok
KAPSAM	Testi deterministik yapmaz	✓ Bariz kararsızlığı CI'a hiç ulaşmadan durdurur

07 TESPİT EMNİYET AĞI, ASIL ÇÖZÜM TASARIM

Testi deterministik yaz

- ☐ Sabit beklemeleeri koşula dayalı beklemeyle değiştir. Saati değil, bir olguyu bekle.
- ☐ Her rastgele kaynağı seed'le; deterministik bir saat ve ID enjekte et.
- ☐ DOM pozisyonuna değil, role veya test id'ye göre seç.
- ☐ Logic'i değil, sınırı mock'la; en az bir gerçek integration yolunu kuru.



08 BU KALIP NE ZAMAN DOĞRU CEVAP OLMAKTAN ÇIKAR?

Statik kapının sınırları



Akıllı race'i yakalamaz

Bariz anti-pattern'leri yakalar, kendi kodunun içine gömülü bir race condition'ı değil.



Kuralları muhafazakâr tut

Gürültülü bir kapı yok sayılır; yok sayılan bir kapı ise hiç olmamasından kötüdür.



Sistem kararsızsa

Test edilen sistem deterministik olmadığı için kararsız olan bir test, sana sistem hakkında bir şey söylüyor.

Yazıldığı satırda

sade.dev

Kaynak: sade.dev, Kararsız Testler: Retry Çözüm Değil

