Panic coding w czasach AI

Jak pracować odpowiedzialnie z generatywnym kodem

BRAVE AI community

BRAVE

AI_devs

Partner wydarzenia:

TIDIO

Zanim przejdziemy dalej

Aleksandra Zalewska
Szczecin, Polska

Senior Frontend Developer
@ Siili Solutions

BRAVE AI_devs Partner wydarzenia: TIDIO

BRAVE AI community

To przyśpieszy Twoja pracę

AI pod ręką.
Deadline za plecami.
"Zrób to z AI".

Problem nie jest w technologii.

BRAVE AI_devs Partner wydarzenia: TIDIO

BRAVE AI community

Co generatywne AI faktycznie zmieniło

- Nie zmieniło tego, że:
  ktoś odpowiada za wynik

- Zmieniło to, że:
  wynik pojawia się zanim zapadnie decyzja

Co robimy pod presją

Gdy:

- czasu jest mało
- oczekiwania są wysokie
- odpowiedzi są natychmiastowe

Nazwijmy to wprost

Panic coding
to moment, w którym

- akceptujemy wynik bez decyzji
- ufamy, bo „działa"
- odkładamy odpowiedzialność na później
- oddajemy ster

To jest psychologia, nie technologia.

Moment, który wszyscy znamy

AI napisało mi 400 linii kodu. W 12 sekund. I… działa.

BRAVE AI_devs Partner wydarzenia: TIDIO

BRAVE AI community

Narzędzie jako katalizator

AI nie powoduje paniki.
AI ją przyspiesza.

BRAVE AI_devs Partner wydarzenia: TIDIO

BRAVE AI community

Gdzie dokładnie tracimy kontrolę

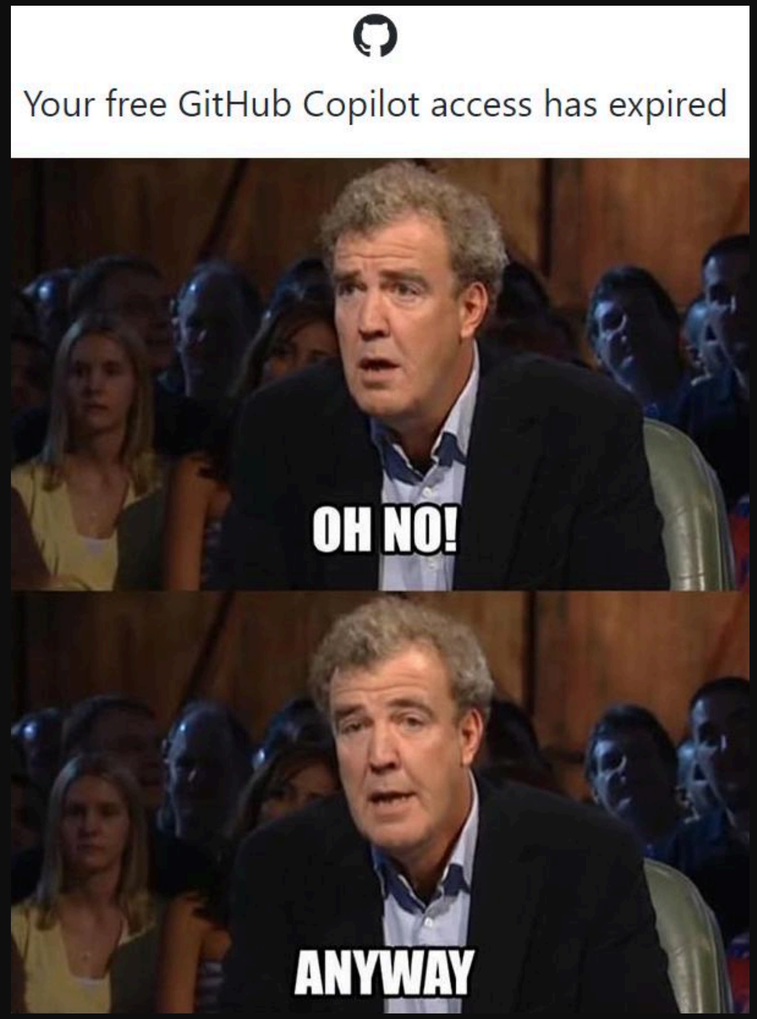
Nie w kodzie.
Tylko w momencie, gdy:

- przestajemy pytać „dlaczego”
- przestajemy sprawdzać
- przestajemy brać odpowiedzialność

Znane narzędzie, znany problem

GitHub Copilot

- ↗ większość z nas go ma
- ↗ wielu z nas próbowało
- ↗ wielu się odbiło



Historia, która kończy się źle

Zadanie rekrutacyjne

- AI pod ręką (można korzystać)
- mam czas
- przecież można to zrobić “lepiej”

Gdzie dokładnie była porażka

Kod działał “za dobrze”

- nie był mój
- nie umiałam go obronić
- pojawił się “red flag”

Problemy, które AI mogło wprowadzić bez weryfikacji:

🔴 **1. Nadmiar bibliotek i zależności**

```
// AI mogło dodać bez sensu:
import { format } from 'date-fns';
import { List } from 'immutable';
import 'bootstrap';
import 'chart.js';
// ... gdy używane są tylko fragmenty
```

🔴 **2. Marnotrawstwo zasobów obrazów**

```
// W imageService.js - AI generuje zbędne ukryte obrazy:
imageSizes.forEach((size) => {
  const hiddenImg = document.createElement('img');
  hiddenImg.style.display = 'none'; // UKRYTE!
  hiddenImg.src = `https://placehold.co/${size}x${size}`;
  // Ładuje obrazy które nigdy nie są pokazane
});
```

🔴 **3. Zbędne event listenery**

```
window.addEventListener('scroll', () => {
  const scrollTop = window.pageYOffset || document.documentElement.scrollTop;
  console.log('Scroll position:', scrollTop); // Tylko console.log!
});

window.addEventListener('resize', () => console.log('Window resized'));
```

Dlaczego to skończy się źle

Bo generatywny kod:

- nie zna kontekstu
- nie zna kosztu błędu
- nie bierze odpowiedzialności

Przykłady promptów, które mogły doprowadzić do problemów:

Prompt 1: "Dodaj system obrazów"

```
"Dodaj system preładowania obrazów dla lepszej wydajności.
Utwórz funkcję która będzie ładować obrazy w różnych rozmiarach."
```

❌ **Rezultat:** AI stworzyło zbędne ukryte obrazy w różnych rozmiarach

Prompt 2: "Dodaj event listenery"

```
"Dodaj event listenery dla scroll i resize żeby aplikacja była responsywna"
```

❌ **Rezultat:** AI dodało listenery które tylko logują do konsoli

Prompt 3: "Ulepsz wydajność"

```
"Dodaj więcej bibliotek do obsługi dat, immutable data i charts"
```

❌ **Rezultat:** AI dodało ciężkie biblioteki dla prostych operacji

Prompt 4: "Stwórz system analityki"

```
"Stwórz zaawansowany system trackingu z retry logic i batch processing"
```

❌ **Rezultat:** Ogromny plik `JS analytics.js` z nieużywanymi funkcjami

Co zmienia wszystko

Odpowiedzialność zaczyna się przed generate

Problem → Plan → Decyzje → Kod

Przykład jak powinno wyglądać odpowiedzialne podejście:

✅ **Prompt z weryfikacją:**

```
"Przeanalizuj czy te biblioteki są rzeczywiście potrzebne.
Pokaż mi które części kodu ich używają. Jeśli date-fns
jest używane tylko do jednej funkcji, zastąp natywnym JS."
```

✅ **Prompt z konkretami:**

```
"Pokaż mi wszystkie miejsca gdzie są dodawane event listenery.
Sprawdź czy każdy z nich ma konkretną funkcjonalność,
nie tylko console.log."
```

✅ **Prompt z budżetem wydajności:**

```
"Bundle nie może przekraczać 200KB. Usuń wszystkie
funkcje i biblioteki które nie są bezpośrednio używane
w aktywnym kodzie."
```

Odpowiedzialny workflow z AI

Jak to wygląda w praktyce

AI:

- generuje warianty
- przyspiesza eksplorację

Człowiek:

- wybiera
- odpowiada

Brutalna prawda o generatywnym kodzie

Generatywny kod

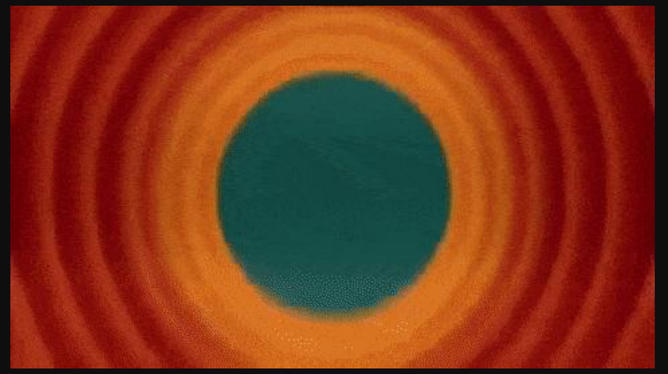
nie zabiera odpowiedzialności
My ją oddajemy.

BRAVE AI_devs Partner wydarzenia: TIDIO

BRAVE AI community

Q&A

W którym momencie generatywny kod decyduje za Ciebie?



Dziękuję za uwagę