PROGRAM SZKOLENIA DOBRE PRAKTYKI

Szkolenie Legacy Code

Przywrócisz do życia stary kod, zmieniając go bezpiecznie i małymi krokami, bez przepisywania od zera.

CZAS TRWANIA	POZIOM	FORMA
2 dni	**średniozaawansowany**	**warsztat dla zespołu, zdalnie lub stacjonarnie**

Kod, którego nikt nie chce dotykać, zwykle zarabia pieniądze firmy. Przepisanie go od zera kusi, ale trwa miesiącami i przynosi nowe błędy. Skuteczniejsza droga to stopniowa naprawa: najpierw zabezpieczasz obecne zachowanie testami, potem porządkujesz kod krok po kroku, a każda zmiana może od razu trafić na produkcję.

Na szkoleniu pracujesz na naprawdę zagmatwanym kodzie JavaScript. Poznasz techniki z książki „Working Effectively with Legacy Code" Michaela Feathersa w wersji dla współczesnego frontendu i Node.js. Uzupełnieniem jest szkolenie Clean Code.

PROGRAM

2 dni

- Czym jest legacy code?
  - Kod bez testów, kod bez właściciela
  - Przepisać czy naprawiać: koszty i ryzyka
- Od czego zacząć?
  - Hotspoty według Adama Tornhilla („Your Code as a Crime Scene")
  - Częstotliwość zmian z historii Gita a złożoność plików
  - `git log`, `git blame` i archeologia decyzji
  - Mapa zależności modułów
- Testy charakteryzujące
  - Zapisanie obecnego zachowania, także błędnego
  - Snapshoty, approval testing i golden master dla dużych wyjść
  - Szkic testów charakteryzujących z asystentem AI i jego weryfikacja
- Szwy (seams)
  - Wstrzykiwanie zależności w istniejący kod
  - Izolowanie czasu, sieci, losowości i globalnego stanu
  - Sprout i Wrap (metoda, klasa) dla nowych funkcji

- Bezpieczna refaktoryzacja
  - Małe kroki pod ochroną testów
  - Automatyczne refaktoryzacje w edytorze
  - Zmiana równoległa (parallel change): rozszerz, zmigruj, zwiń
  - Krótkie gałęzie i częste scalanie zamiast wielkiej przebudowy
- Wzorzec strangler fig (stopniowe zastępowanie)
  - Wymiana modułów i ekranów po kawałku
  - Feature flagi przy przełączaniu implementacji
- Migracja do TypeScript
  - Migracja plik po pliku: `allowJs`, `checkJs` i JSDoc
  - Start od `"strict": false`, bo TypeScript 6 włącza `strict` domyślnie
- Node.js, zależności i martwy kod
  - Node.js po końcu wsparcia (Node.js 20 od 2026-04-30, Node.js 22 od 2027-04-30), CommonJS do ESM z `require(esm)` jako pomostem
  - Porzucone biblioteki i aktualizacje z Dependabotem lub Renovate
  - Nieużywane pliki, eksporty i zależności (Knip), usuwanie bez strachu dzięki historii Gita

Szczegóły migracji do TypeScript 6 i 7 omawiam na szkoleniu TypeScript, a pracę z asystentami AI na szkoleniu AI dla programistów.

DLA KOGO?

- Dla programistów, którzy utrzymują kilkuletni kod bez testów.
- Dla liderów technicznych, którzy planują modernizację projektu.
- Dla zespołów rozważających przepisanie aplikacji od zera.

CO MUSISZ UMIEĆ?

- Swobodnie pisać kod w JavaScript.
- Pisać proste testy jednostkowe.
- Pracować z Gitem na co dzień.
- Korzystać z asystenta AI w edytorze (moduł o testach charakteryzujących).

CZEGO SIĘ NAUCZYSZ?

- Wskażesz na podstawie historii Gita miejsca, od których warto zacząć.
- Zabezpieczysz obecne zachowanie kodu testami charakteryzującymi.
- Wydzielisz zależność od czasu, sieci i globalnego stanu, żeby dało się napisać test.
- Wprowadzisz zmiany małymi krokami, bez długo żyjących gałęzi.
- Zastąpisz stary moduł nowym stopniowo, wzorcem strangler fig.
- Zaplanujesz migrację do TypeScript plik po pliku, bez zatrzymywania prac.
- Zaktualizujesz Node.js i zależności oraz usuniesz martwy kod.
- Wykorzystasz asystenta AI do szkiców testów charakteryzujących i sprawdzisz ich wynik.

OPINIE UCZESTNIKÓW

> *„Pod okiem Piotra nauczysz się pisać poprawny, czytelny kod i wyćwiczysz w sobie dobre praktyki.”*
>
> Małgorzata, Frontend Developer (WarsawJS MasterClass i WarsawJS Workshop)

TRENER

Piotr Kowalski

W branży IT od 2008 roku. Od 2014 jest mentorem dla wielu developerów. Jako trener spędził na sali szkoleniowej ponad 1000 godzin, a na LinkedIn ma 52 rekomendacje. Każde jego szkolenie to dawka solidnej wiedzy oraz humoru.

Więcej: piecioshka.pl/piotr-kowalski

ZORGANIZUJMY TO SZKOLENIE W TWOIM ZESPOLE

Program dopasuję do Waszego projektu i poziomu grupy. Napisz, z czym pracujecie, a odpowiem z propozycją terminu i zakresu.

KONTAKT	**piecioshka.pl/kontakt**
E-MAIL	**piecioshka+blog [małpka] gmail.com**
SZKOLENIE	**piecioshka.pl/szkolenia/legacy-code/**