



AI IDZIE PO TWOJĄ PRACĘ...

... ale dowiedz się jak być nie do zastąpienia!

Uważam, że:

**AI na pewno
zabierze mi pracę**

**Istnieje ryzyko,
że AI zabierze
mi pracę**

**Nie ma szans, że
AI zabierze mi
pracę.**

Nie umiem
programować

Nie mam wykształcenia
informatycznego

Nie jestem księciem z bajki
ani synem króla nafty.
Nie jestem nawet
dziennikarzem TVNu.

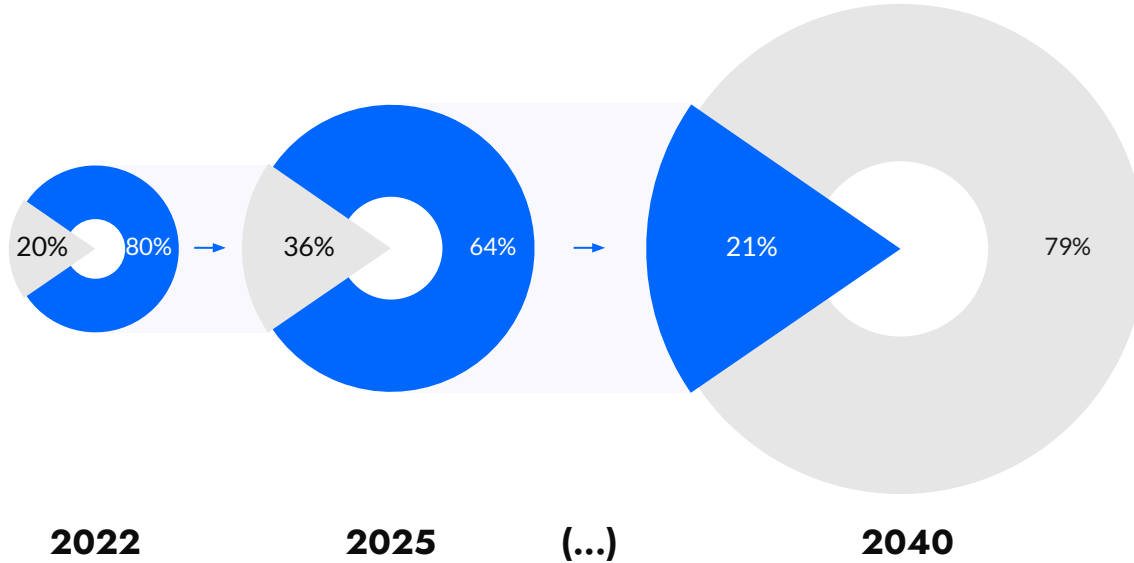
Wystarczająco zupełnie wystarczy

Moja historia w branży IT zaczęła się od tego, że - blisko dekadę temu - umiałem zlokalizować ikonkę Photoshopa na pulpicie komputera. I tak, w 2016 roku założyłem działalność gospodarczą, a trzy lata temu Spółkę Unlimitech, gdzie koncentrujemy się głównie na migracjach i wdrożeniach systemów e-commerce. Nasze przychody rok do roku utrzymują się na poziomie około 2 mln. zł.

Zasada pareto w obszarze programowania AI

Ile to jest wystarczająco?

■ Człowiek ■ Maszyna



ROK 1900

Gdzie jest
samochód?



ROK 1913

Gdzie jest
koń?



...koniowi opowiadasz?

Nie tylko konie

01

Przemysł wielorybiczny w szczytowym momencie zatrudniał około 70 000 osób. Z czasem olej wielorybi zastąpiła ropa naftowa i w końcu energia elektryczna - do dziś milionowy rynek zatrudnienia.

02

W latach 60-tych XIX wieku w samych stanach zjednoczonych zatrudnionych było ok. 20 tys. **telegrafistów**. Później pojawiły się telefony stacjonarne, a dziś przemysł telefonii komórkowej to podobnie jak przemysł elektryczny - milionowy rynek pracy.

03

Przemysł lodowy zdominował rynek do 1930 roku, zatrudniając blisko 100 000 osób w 1900 roku, by zniknąć ostatecznie w wyniku rozwoju chłodnictwa elektrycznego, tworząc w tej dziedzinie nowe miejsca pracy w produkcji urządzeń chłodniczych.

*Za każdym razem kiedy myślisz, że coś możesz
lub czegoś nie możesz zrobić - masz rację.*

Henry Ford

*Człowiek rozsądny dostosowuje się do świata.
Człowiek nierozsądny uporczywie stara się dostosować świat do
siebie. Dlatego wszelki postęp zależy od ludzi nierozsądnych.*

George Bernard Shaw

Podsumowując

Jak być nie do zastąpienia w dobie AI

01

Zachować otwarty umysł na możliwości, które ze sobą niesienie rozwój generatywnej sztucznej inteligencji czy sztucznej inteligencji ogółem.

02

?

03

?

AI w pracy programistów

Wykorzystywanie narzędzi AI

61,8%

Wykorzystuje AI w procesie programowania.

13,8%

Planuje zacząć wykorzystywać AI.

24,4%

Nie zamierza wykorzystywać AI w swojej pracy.

Sentyment do narzędzi AI

72%

Ma dobre zdanie na temat narzędzi AI wspierających programowanie.

18,7%

Nie ma zadania na ten temat.

6,4%

Źle lub bardzo źle ocenia narzędzia AI wspierające programowanie.

Zaufanie do narzędzi AI

39,3%

Profesjonalistów mniej lub
bardziej ufa w kod
generowany przez AI.

22,9%

Raczej nie ufa w kod
generowany przez AI.

10,6%

Skrajne zaufanie (2,7%
respondentów głęboko ufa
AI, 7,9% w ogóle nie ufa).

Uważam, że:

**AI na pewno
zabierze mi pracę**

**Istnieje ryzyko,
że AI zabierze
mi pracę**

**Nie ma szans, że
AI zabierze mi
pracę.**

AI w pracy programistów

Obawa przed AI

69,6%

Profesjonalistów nie obawia się utraty pracy na rzecz AI.

27,1%

Profesjonalistów nie ma pewności co do zagrożenia.

15%

Profesjonalistów obawia się o swoją pracę.

Dlaczego nie należy ignorować zagrożenia?

27%

Tylko w ciągu ostatnich dwóch lat rynek zatrudnienia programistów w USA zmniejszył się o 27%.*

300 mln

Już w 2023 prognozy zakładały, że automatyzacja oraz AI będą miały wpływ na 300 mln miejsc pracy.**

2 mln

Replit Agent umożliwił swoim użytkownikom "wypromptowanie" 2 mln aplikacji w 6 miesięcy.***

* Washington Post - More than a quarter of computer-programming jobs just vanished. What happened? (14 Marzec 2025)

** Joseph Briggs, Devesh Kodnani - The Potentially Large Effects of Artificial Intelligence on Economic Growth (26 Marzec 2023)

*** Kyle Poyar - Inside Replit's journey to making 2 million apps in 6 months (19 Marzec 2025)

Podsumowując

Jak być nie do zastąpienia w dobie AI

01

Zachować otwarty umysł na możliwości, które ze sobą niesie rozwój generatywnej sztucznej inteligencji i sztucznej inteligencji ogółem.

02

Aktywnie wykorzystywać GenAI w swojej pracy, usprawniać i automatyzować work-flow.

03

?

Kontekst iteracji pracy (commit)

GitHub Copilot

- Dołożenie kontekstu sąsiednich zakładki w GitHub Copilot zwiększyło skuteczność **generowanego kodu o 5%**,
- Zastosowanie techniki Fill-In-The-Middle w Copilot pozwoliło uzyskać dodatkowe **+10% do skuteczności generowanego kodu**,
- GenAI na poziomie pojedynczego pliku potrafi dość dobrze dopasować składnię i wywołania funkcji do otaczającego kodu, o ile ten kontekst mieści się w jego oknie pamięci

Write a unit test for this function



GitHub Copilot

```
import unittest
import datetime

class TestParseExpenses(unittest.TestCase):
    def test_parse_expenses_with_valid_input(self):
        expenses_string = '2023-01-02 -34.01 USD\n2023-01-03 -12.50 USD'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD'),
                           (datetime.datetime(2023, 1, 3), -12.50, 'USD')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_empty_input(self):
        expenses_string = ''
        expected_output = []
        self.assertEqual(parse_expenses(expenses_string), expected_output)

if __name__ == '__main__':
    unittest.main()
```

The code assumes that the datetime module is imported. The test cases cover different scenarios such as valid input, empty input, input with comments, invalid date format, and invalid value format.

Ask a question or type '/' for commands

ChatGPT

- **GPT4** rozwiązuje poprawnie ok. 68% zadań programistycznych z zestawu HumanEval,
- Bez kontekstu między-plikowego skuteczność realizacji bardziej złożonych zadań jest zatrważająco niska (np. GPT-3.5 EM tylko 4.88% w Python),
- GPT-4 bez dodatkowego kontekstu radzi sobie dobrze z zadaniami izolowanymi (około 35% poprawnych rozwiązań w kategorii zadań zależnych tylko od jednego pliku), ale przy zadaniach wymagających wiedzy o całym projekcie jego skuteczność spadła do **~19%**,

 ChatGPT:

Jasne! Oto przykładowe testy jednostkowe:

python

```
import unittest

class TestCalculateTotal(unittest.TestCase):
    def test_multiple_items_with_quantity(self):
        items = [
            {"price": 10, "qty": 2},
            {"price": 5, "qty": 3}
        ]
        self.assertEqual(calculate_total(items), 10*2 + 5*3)

    def test_missing_quantity_defaults_to_one(self):
        items = [
            {"price": 15},
            {"price": 7, "qty": 2}
        ]
        self.assertEqual(calculate_total(items), 15 + 7*2)

    def test_empty_list(self):
        self.assertEqual(calculate_total([]), 0)

    def test_invalid_structure(self):
        # Tutaj AI nie wie, jak aplikacja ma obsługiwać błąd
```

Zapytaj o cokolwiek



 Wyszukaj

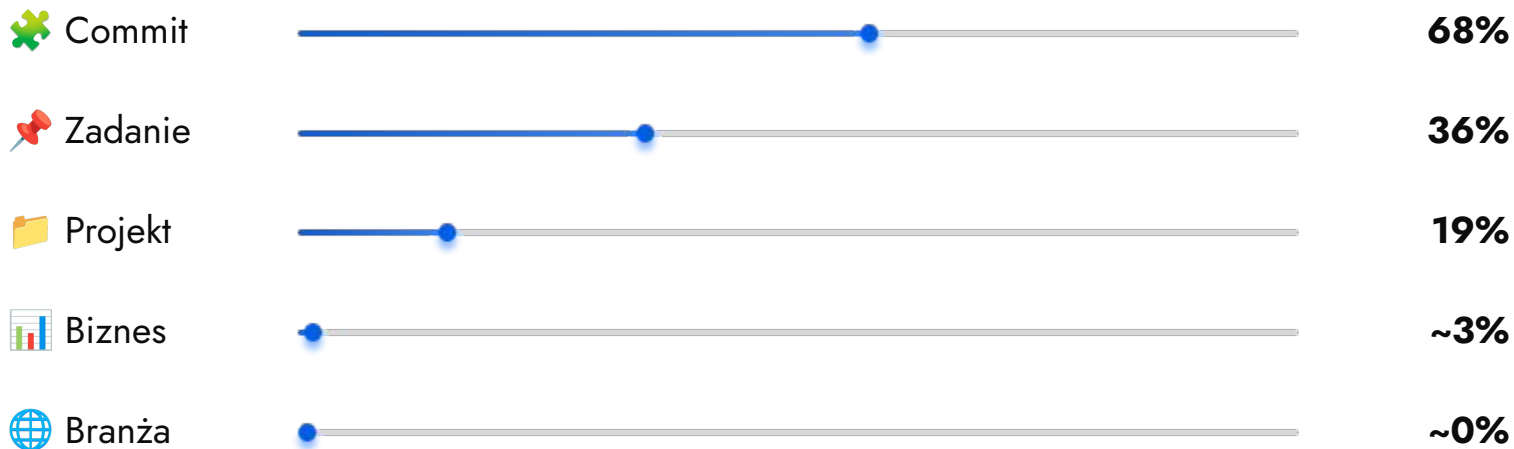
 Zbadaj głęboko

 Stwórz obraz



Kontekst vs Kontent

Skuteczność kontekstowa



AI dobrze radzi sobie z zadaniami, ale nie rozumie projektu, nie czuje biznesu, nie ogarnia branży. Jeśli Twoja wartość kończy się na wykonaniu — masz problem. Jeśli ogarniasz dlaczego i dla kogo — jesteś niezastąpiony.

Podsumowując

Jak być nie do zastąpienia w dobie AI

01

Zachować otwarty umysł na możliwości, które ze sobą niesie rozwój generatywnej sztucznej inteligencji i sztucznej inteligencji ogółem.

02

Aktywnie wykorzystywać GenAI w swojej pracy, usprawniać i automatyzować work-flow.

03

Rozwijać umiejętności krytycznego myślenia, analizy biznesowej, myślenia kontekstowego, pisanie kodu pozostawić LLMom :>

Epilog

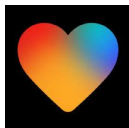
Programiści w erze *vibe codingu*



Replit Agent umożliwia tworzenie aplikacji i stron internetowych poprzez naturalne polecenia tekstowe, bez potrzeby pisania kodu, co obniża próg wejścia do świata tworzenia oprogramowania.



Devin AI umożliwia tworzenie aplikacji i stron internetowych poprzez naturalne polecenia tekstowe, bez potrzeby pisania kodu, a także samodzielne rozwiązywanie problemów programistycznych, co może zrewolucjonizować sposób tworzenia oprogramowania.



Zinus



- Dwóch inżynierów stworzyło z pomocą Replit Agent działającą aplikację QA w 1,5 miesiąca (zamiast 3), oszczędzając ponad **\$140 tys. budżetu i 50% czasu developmentu.**
- Zamiast specyfikacji – rozmowy z AI. Programiści opisywali potrzebę, a Agent generował interfejsy (React + Material UI) i backend – całość doszlifowana iteracyjnie.
- Działająca appka z AI do analizy rozmów + dashboard, wdrożona jednym kliknięciem, która zastąpiła zewnętrzne narzędzie, dając dodatkowe **\$40k oszczędności rocznie.**



Nubank



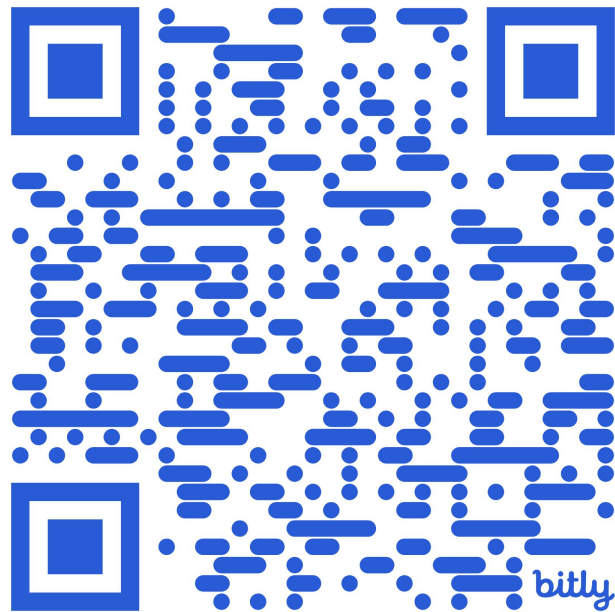
- Nubank musiał zmodernizować monolityczny system ETL z 6 mln linii kodu. Zadanie zaplanowane na ponad **+ 100 inżynierów i 18 miesięcy**.
- Po fine-tuningu Devina na ich kodzie, AI samodzielnie refaktoryzowała komponenty, generowała skrypty pomocnicze i poprawiała własne błędy. Programiści głównie weryfikowali pull requesty i zajmowali się krytycznymi częściami procesu.
- Projekt **przyspieszony 8–12x**, **koszty refaktoryzacji spadły 20-krotnie**, a AI zastąpiła tysiące godzin ręcznej pracy – wszystko z zachowaniem kontroli jakości.



Szansa czy zagrożenie?

Wpływ na branżę software prawdopodobnie będzie coraz większy - rola programistów ewoluuje w kierunku projektowania, nadzorowania i integracji, podczas gdy coraz większa część kodu - zwłaszcza tego odtwórczego - będzie pisana przez sztuczną inteligencję.

Jak wskazują statystyki, firmy, które już wdrożyły AI w proces tworzenia, odnotowują znaczny wzrost wydajności i skrócenie cykli rozwojowych.



Kompetencje

1. Potrafi wyrazić *każdy* problem techniczny i każdy cel biznesowy w języku naturalnym tak precyzyjnie, że AI generuje dokładnie to, czego potrzebuje.
2. Zna różnicę między „napisz funkcję” a „stwórz proces walidacji danych z logiką branżową X i regułami Y”.
3. Nie buduje aplikacji sam - buduje je wspólnie z AI. Tworzy architekturę tak, by Agent mógł wygenerować 80% kodu, a on dopieszcza tylko krytyczne fragmenty.
4. Zamiast pisać endpointy REST, zleca ich wygenerowanie AI, skupiając się na integracjach, UX i bezpieczeństwie.
5. Nie myśli „jak napisać funkcję”, tylko „jaki problem klienta rozwiązuję, jak będzie skalowalny, jak wpłynie na revenue, UX i support”.

Nie myśli językami
programowania.
Myśli rezultatami.

Nie przywiązuje się do
konkretnego stacku - rozumie
zasady programowania, zna
wzorce projektowe.

Postawa i mindset

1. Tworzy nietypowe rozwiązania łącząc API, modele językowe, workflow automatyzacji i interfejsy użytkownika w sposób nieszablonowy.
2. Nie dąży do perfekcji. Zleca AI wygenerowanie prototypu, testuje go z użytkownikiem, refinuje w locie.
3. Rozumie, że 10 iteracji z AI w 1 dzień to więcej niż 1 tydzień tradycyjnego developmentu.
4. Nie boi się zmian. Nie zamyka się w stacku. Jest typem programisty, który w zna i rozumie podstawy stacków, które wykorzystuje, dostosowuje rozwiązania do problemów.
5. Zamiast zaczynać od kodu, zaczyna od mapy: co ma powstać, kto to będzie używał, gdzie może się rozsypać.

Nie skupia się na tym, jak coś zakodować, tylko dlaczego to ma działać tak, a nie inaczej.

Rozumie produkt, użytkownika, rynek, model biznesowy i to, co chce osiągnąć właściciel firmy.

To nie programista
- klepacz kodu.
To programista - strateg.

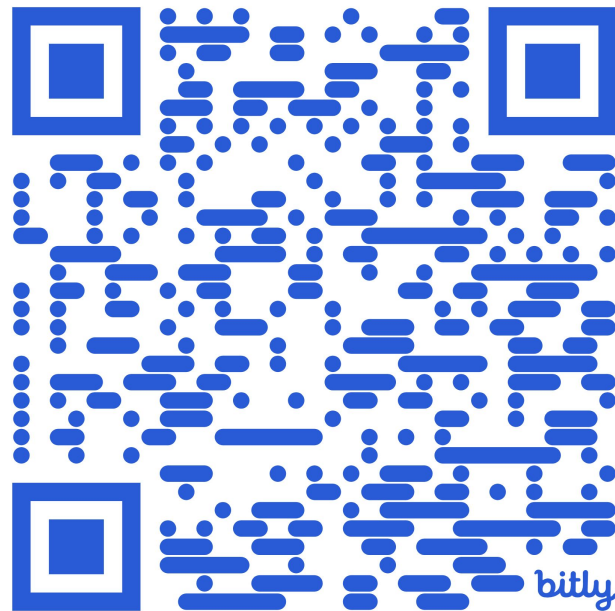
Wiedza i doświadczenie

1. Zna ograniczenia modeli LLM: gdzie halucynują, gdzie się mylą, jak je dopalać przez fine-tuning i RLHF.
2. Używa narzędzi typu Replit Agent, Devin AI, Vercel v0, AWS App Studio tak naturalnie, jak dawni programiści używali edytora Vim.
3. Tworzy nietypowe rozwiązania łącząc API, modele językowe, workflow automatyzacji i interfejsy użytkownika w sposób nieszablonowy.
4. Gdy trzeba coś „low-level”, wchodzi tylko tam, gdzie AI nie ogarnia.
5. Nie skupia się na tym co sam potrafi, umie czy wie, koncentruje się na tym co robić, żeby jego AI pracowało bardziej wydajnie, skutecznie i bezbłędnie.

Dla niego "Zrób login" nie
znaczy "wstaw formularz".
Znaczy: "Jaki poziom UX?
Jakie dane warto śledzić?
Czy to będzie bariera, czy
wartość?"

Zamiast pisać 300 linii kodu,
pisze jeden precyzyjny
prompt i dostaje gotową
funkcjonalność.
Ale nie bierze jej w ciemno.

**Czy Ty jesteś nie do
zastąpienia?**



Dzięki za uwagę
i powodzenia - w życiu nie do zastąpienia.



Alek Gołaszewski

CEO Unlimitech

T: +(48) 724 787 012

M: alek@unlimitech.pl

WWW: unlimitech.pl