



# **Wdrożenie GitHub Copilot w software house (C / Java)**

Zwiększenie produktywności zespołów developerskich poprzez AI-assisted development

# Kontekst Organizacyjny: Nasze Wyzwania

Zrozumienie specyfiki naszej firmy jest kluczowe dla efektywnego wdrożenia nowych technologii.

## Charakterystyka organizacji:

- Software house / firma produktowa
- ~100 developerów
- Dominujące technologie:
  - C (embedded, performance, legacy)
  - Java (backend, systemy enterprise)
- Praca zespołowa: Git, code review, CI/CD
- Wysokie koszty pracy developera
- Presja na:
  - time-to-market
  - jakość kodu
  - utrzymanie legacy



## Wyzwania biznesowe:

- Powtarzalny boilerplate
- Długi onboarding nowych pracowników
- Wąskie gardła u doświadczonych seniorów
- Rosnące koszty utrzymania i rozwoju projektów

# Cel Wdrożenia GitHub Copilot

Naszym nadzędnym celem jest osiągnięcie mierzalnego wzrostu produktywności zespołów poprzez strategiczne wykorzystanie AI.

## Zwiększenie Efektywności

Optymalizacja pracy developerów poprzez automatyzację powtarzalnych zadań.

## Skrócenie Czasu Realizacji

Przyspieszenie cyklu rozwoju oprogramowania i dostarczania wartości.

## Lepsze Wykorzystanie Czasu Seniorów

Przeniesienie uwagi seniorów z boilerplate'u na złożone problemy i mentoring.

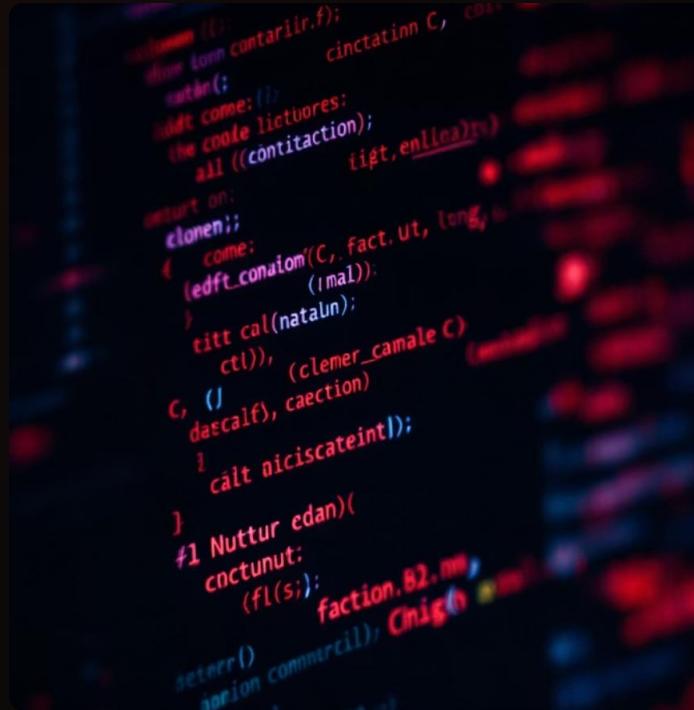
## Standaryzacja Jakości Kodu

Wsparcie w utrzymaniu wysokiej jakości i spójności kodu w całym projekcie.



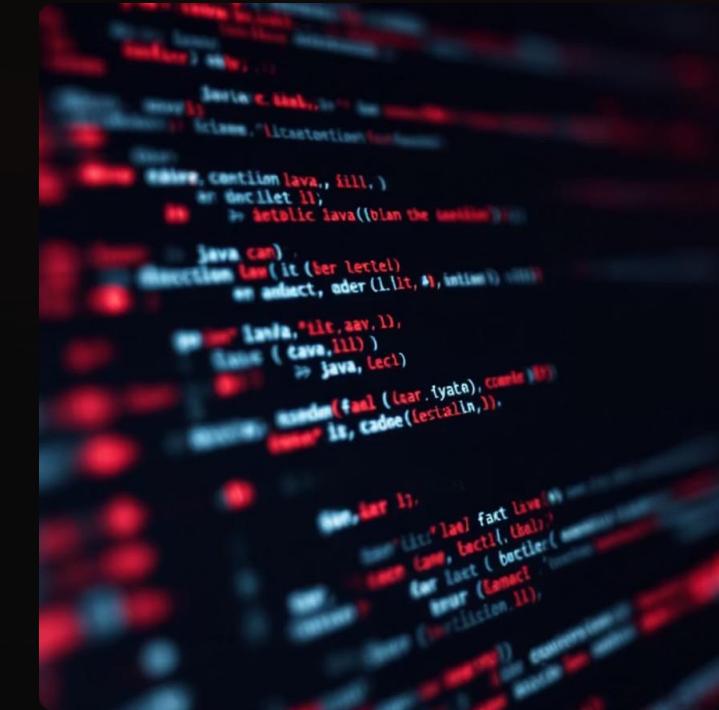
# Jak Copilot Wspiera Zespoły C i Java

GitHub Copilot oferuje dostosowane wsparcie, które adresuje specyficzne potrzeby każdej z naszych kluczowych technologii.



## Wsparcie dla C

- Generowanie bezpiecznych wzorców kodu
- Ułatwienie pracy z kodem legacy
- Redukcja błędów niskopoziomowych, w tym problemów z zarządzaniem pamięcią



## Wsparcie dla Java

- Szybkie tworzenie warstw aplikacji i komponentów
- Generowanie testów jednostkowych i integracyjnych
- Wsparcie w refaktoryzacjach i implementacji wzorców architektonicznych

# Inwestycja w Przyszłość: Skala i Koszty

Analiza kosztów wdrożenia GitHub Copilot w kontekście naszych zasobów ludzkich.

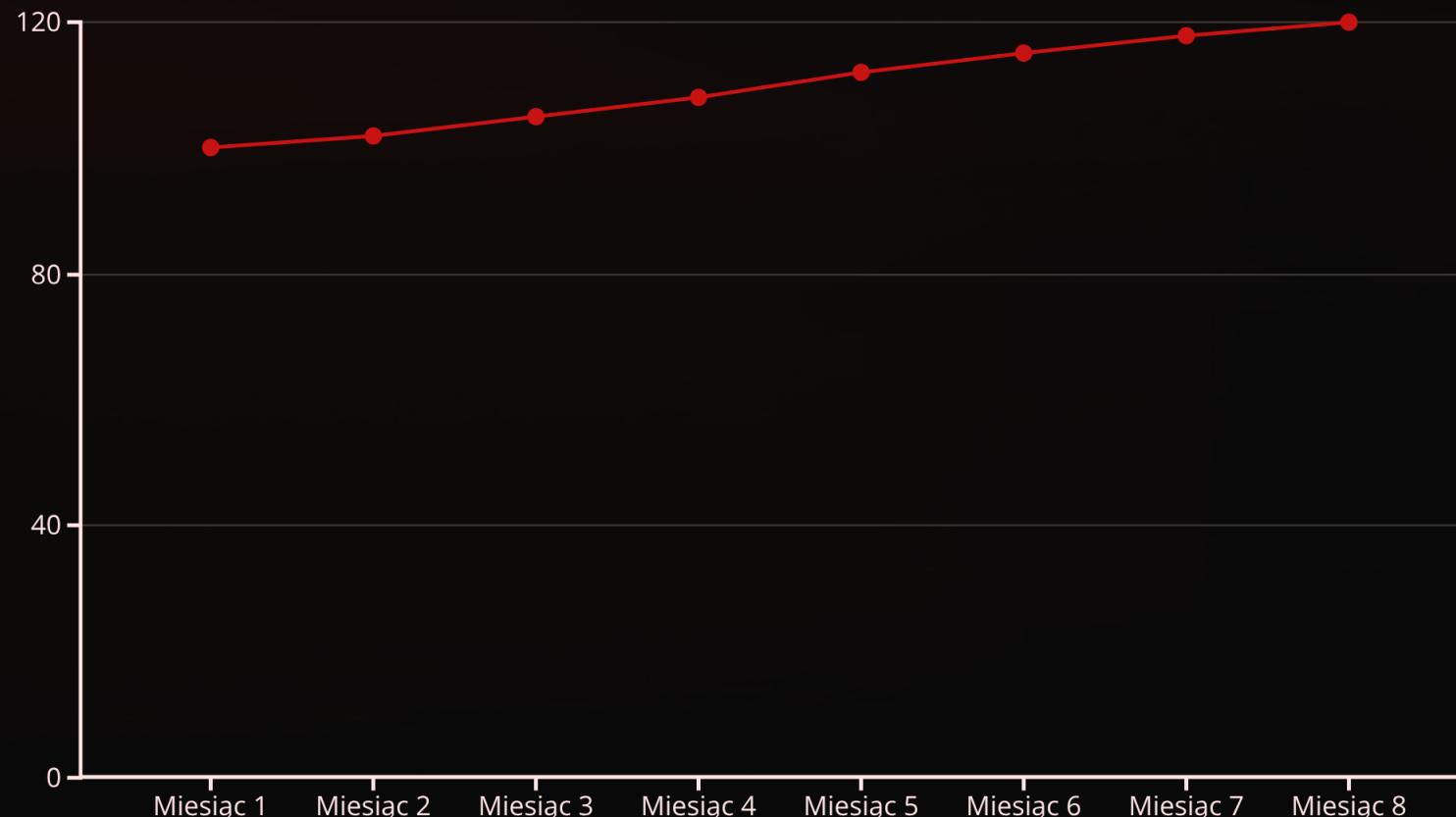


- Koszt licencji: ~100 zł / użytkownik / miesiąc
- Dla 100 developerów: ~10 000 zł / miesiąc
- Roczny koszt: ~120 000 zł / rok

W zestawieniu z całkowitymi kosztami pracy zespołu, inwestycja w Copilota stanowi ułamek. To mniej niż godzina pracy developera miesięcznie, co czyni ją łatwą do zaakceptowania na poziomie zarządczym, biorąc pod uwagę potencjalne korzyści.

# Wpływ na Produktywność: Mierzalne Korzyści

Oczekujemy znaczącego wzrostu efektywności, który przełoży się na realne oszczędności czasu i zasobów.



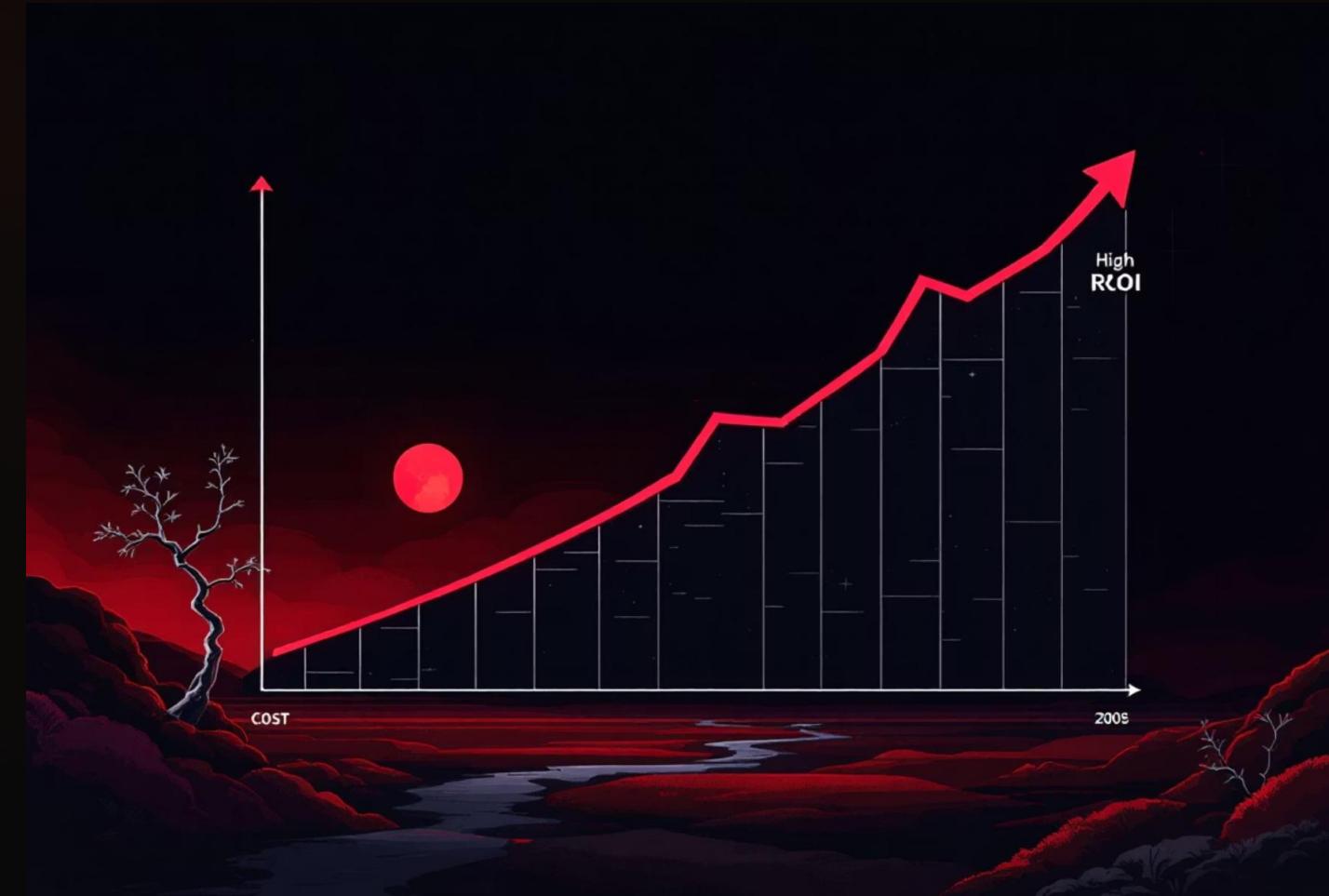
Wykres: Produktywność zespołu – konsekwentnie w górę i w prawo, odzwierciedlając ciągły wzrost po wdrożeniu Copilota.

## Kluczowe metryki:

- Szacowany wzrost produktywności: +10–20%
- Skrócenie czasu realizacji zadań projektowych
- Znaczne zmniejszenie czasu poświęcanego na kod powtarzalny i boilerplate
- Szybsze wdrażanie nowych funkcjonalności i rozwiązań

# ROI i Wartość Biznesowa

GitHub Copilot to inwestycja o szybkim zwrocie, która minimalizuje koszty i maksymalizuje wartość.



Wykres ROI: Koszt licencji pozostaje stały, podczas gdy generowana wartość biznesowa i oszczędności rosną wykładniczo.

## Niski Koszt Licencji

Relatywnie niewielka opłata miesięczna w porównaniu do potencjalnych korzyści.

## Wysoki Koszt Godziny Developerskiej

Każda zaoszczędzona godzina pracy developera przekłada się na realne oszczędności finansowe.

## Szybki Zwrot z Inwestycji

Już niewielka oszczędność czasu pokrywa koszt narzędzia, generując czysty zysk.

# Ryzyko Operacyjne: Dostępność Łącza

Świadomie zarządzamy ryzykami, aby zapewnić ciągłość pracy i stabilność systemu.

## Kluczowe zagadnienia:

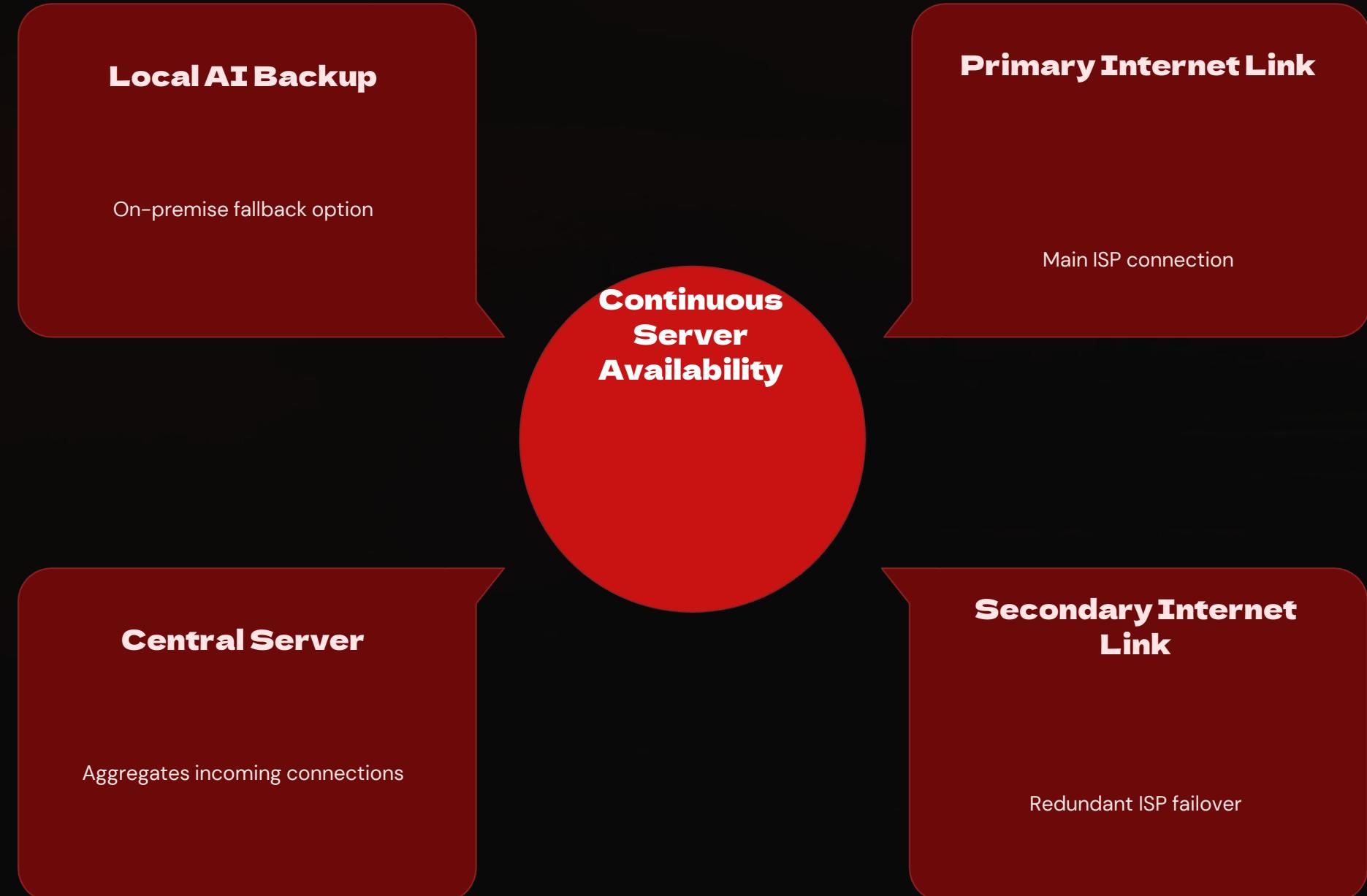
- Copilot wymaga stałego połączenia z internetem do funkcjonowania.
- Brak dostępu do sieci = brak wsparcia AI, co może wpływać na płynność pracy.
- Należy pamiętać, że core development (IDE, kompilatory) działa niezależnie od dostępności Copilota.

Ryzyko związane z dostępnością łączna jest nam znane i w pełni kontrolowalne. Zastosujemy odpowiednie mechanizmy, aby minimalizować jego wpływ na produktywność zespołów.



# Propozycja Zabezpieczenia: Plan Awaryjny

Zabezpieczamy się przed potencjalnymi przerwami w dostępie, aby utrzymać najwyższy poziom dostępności i produktywności.



Wdrożenie podwójnego łącza oraz rozwarcie lokalnych narzędzi AI jako opcji awaryjnej minimalizuje ryzyko przestoju.

# Podsumowanie i Następne Kroki

Wdrożenie GitHub Copilot to strategiczna inwestycja, która wzmocni naszą pozycję na rynku i przyspieszy rozwój.

## 1 Potwierdzenie Korzyści

Zwiększoną produktywność, skrócenie czasu realizacji zadań i optymalne wykorzystanie zasobów.

## 2 Niski Koszt, Wysoki Zwrot

Inwestycja o mierzalnym ROI, z minimalnym ryzykiem finansowym.

## 3 Minimalizacja Ryzyk

Aktywne zarządzanie ryzykiem dostępności łącza poprzez plany awaryjne.

## 4 Propozycja Dalszych Działań

Rozpoczęcie pilotażowego wdrożenia i monitorowanie metryk produktywności w wybranych zespołach.



# Nasz Zespół Projektowy

Poznaj osoby zaangażowane w tworzenie i rozwój tego projektu, w tym głównych autorów prezentacji.



**Jan  
Woźniak**

Numer albumu: 110986

Główny autor  
prezentacji, specjalista  
od wizualizacji.



**Arkadiusz  
Owczarek**

Numer albumu: 119205

Główny autor  
prezentacji, specjalista  
od wizualizacji.



**Adrian  
Wieczorek**

Numer albumu: 119202

Członek zespołu,  
zaangażowany w  
analizę danych.



**Klaudiusz  
Miniak**

Numer albumu: 110566

Członek zespołu,  
odpowiedzialny za  
badania.



**Krzysztof  
Mazerant**

Numer albumu: 119203

Członek zespołu,  
ekspert w dziedzinie  
technologii.



**Piotr  
Roman**

Numer albumu: 122217

Kierownik zespołu,  
wsparcie techniczne i  
programistyczne.