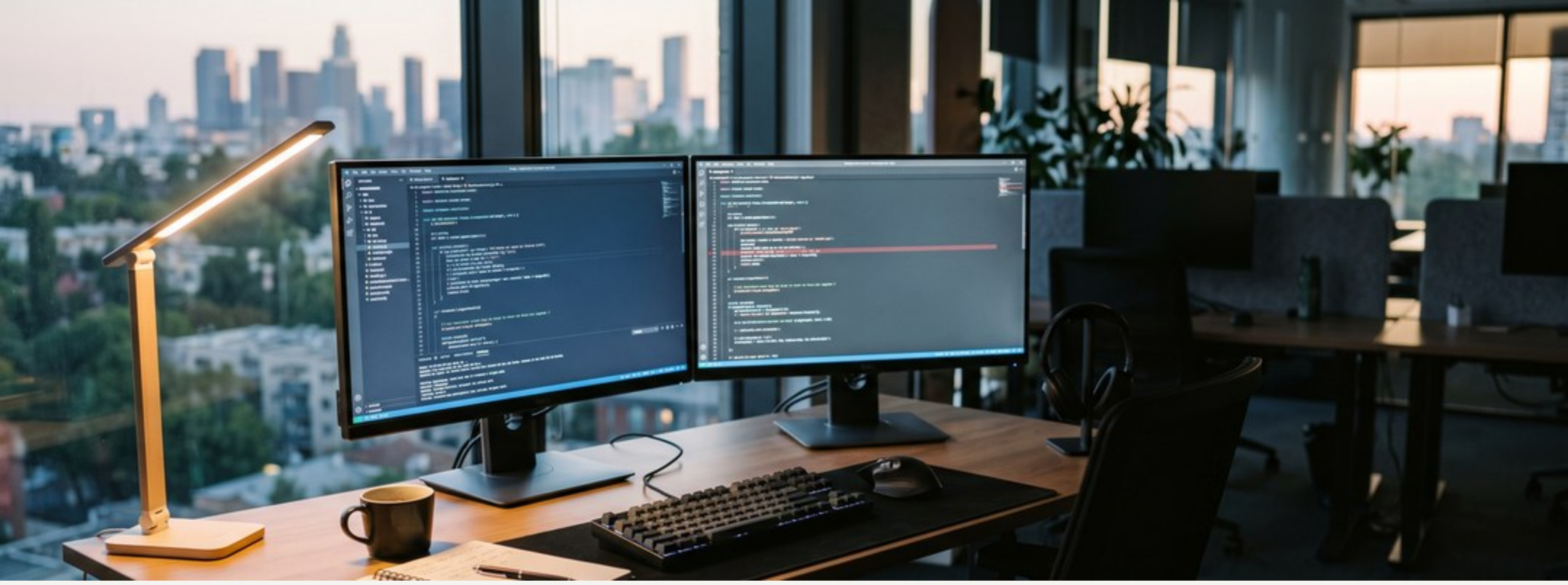
SNOK

Jacek Bugajski • CEO, SNOK

Kompiluje się, więc działa

ABAP w czasach,
gdy kod pisze asystent

O progu, który zniknął,
i o podpisie, który
przestał mieć pokrycie



01

Przez trzydzieści lat ABAP bronił się sam

Nie składnią, bo ta akurat wybacza wiele. Bronił się progiem wejścia: słownikiem danych, koncepcją uprawnień, logiką modułu i rytuałem transportów przez trzy systemy.

Ta wiedza nie leżała w dokumentacji. Leżała w głowach ludzi po kilku wdrożeniach i kilku nieudanych go-live. Przekazywało się ją przy biurku, nie z kursu.

I właśnie ten próg zniknął. Nie przez dekadę - w kilkanaście miesięcy.

02

Co się zmieniło pod maską

SAP udostępnił model wyszkolony wyłącznie na ABAP-ie. SAP-ABAP-1 uczył się na milionach linii kodu: obiektach standardowych, rozszerzeniach klientów, implementacjach BAdI, wzorcach ABAP Cloud. W materiałach SAP Community pada liczba ponad 250 milionów linii.

Zasila wyjaśnianie i generowanie kodu w Joule for Developers, wbudowanym w narzędzia w Eclipse i w SAP Business Application Studio. W 2026 doszło rozszerzenie ABAP dla Visual Studio Code.

To nie jest tylko historia o szybszym narzędziu dla ABAPera. Żeby napisać w SAP coś, co się kompiluje i przechodzi test funkcjonalny, nie trzeba już być ABAPerem.

Raport napisze konsultant funkcjonalny.
Napisze go analityk z zespołu danych.
Napisze go junior po trzech tygodniach

Kod się kompiluje

To niczego nie dowodzi

Kompilator w SAP nigdy nie sprawdzał tego,
co w tym systemie naprawdę groźne

2,74x

więcej znalezisk bezpieczeństwa miały pull requesty współtworzone z AI niż pisane wyłącznie przez ludzi.

Problemów ogółem - logiki, wydajności, utrzymywalności - było 1,7 raza więcej

CodeRabbit, State of AI vs Human Code Generation Report, grudzień 2025: 470 pull requestów z projektów otwartych. Badanie dostawcy narzędzia do przeglądu kodu

Kod z asystenta ma jedną cechę,
która usypia czujność najskuteczniej:
on działa

SNYK

03

Czego kompilator nigdy nie sprawdzał

Kontroli uprawnień

czy przed odczytem danych kadrowych stoi AUTHORITY-CHECK

Wstrzyknięć

czy zapytanie zbudowane dynamicznie z wartości od użytkownika da się rozmontować

Modułów RFC

czy udostępniony interfejs weryfikuje uprawnienia wywołującego

Zapisu do tabel

czy dane idą przez logikę aplikacji, czy prosto do tabeli

Kodowania wyjścia

czy pole trafiające do przeglądarki jest kodowane

Model tych rzeczy nie pominął złośliwie. On ich po prostu nie wie - nie widział Waszej koncepcji uprawnień ani macierzy rozdziału obowiązków

04

Trzy liczby, które robią z tego temat operacyjny

3–4x szybciej oddają kod deweloperzy pracujący z asystentem, a znalezisk bezpieczeństwa przybyło im dziesięciokrotnie w pół roku (Apiiro)

45% zadań programistycznych kończyło się kodem z podatnością (Veracode, ponad 100 modeli)

< 3 h tyle żyje niezabezpieczona aplikacja SAP w chmurze, zanim zostanie wykryta i zaatakowana (Onapsis)

Asystent nie pisze gorzej od człowieka. Pisze znacznie więcej, a liczba osób, które ten kod czytają, nie zmieniła się o jedną.

05

Clean Core podniósł stawkę

Przez lata kod Z siedział wewnątrz Waszych czterech ścian. Napisany byle jak, ale za zaporą sieciową, w systemie, do którego dostęp miało kilkadziesiąt osób.

Clean Core zmienił geografię. Rozszerzenia wychodzą z rdzenia na BTP i stają się aplikacjami w chmurze. Kierunek jest słuszny i sam go rekomenduję. Skutkiem ubocznym jest to, że ten sam kod, napisany z tą samą starannością, wisi teraz w internecie.

Ten sam wniosek przewijał się przez wspólny webinar Deloitte i Onapsis: Clean Core bez przesunięcia kontroli bezpieczeństwa na początek procesu jest półśrodkiem.

06

Kto się podpisze

W SAP mamy coś, czego nie ma większość świata IT: fizyczny, imienny moment odpowiedzialności. Ktoś zwalnia transport, ktoś inny importuje go na produkcję. Oba nazwiska zostają w logu.

Ten mechanizm powstał w czasach, gdy zwolnienie transportu znaczyło: przeczytałem, rozumiem, biorę to na siebie. Dziś coraz częściej znaczy: zaakceptowałem propozycję, której nie przeczytałem.

Rola ABAPera przesuwa się z pisania na walidację i to zmiana zdrowa. Kłopot w tym, że gdy zespoły oddawały pisanie maszynie, nikt nie dołożył im etatów na czytanie.

Podpis się nie zmienił.
Zmieniło się jego pokrycie

07

Dziś jest na to rada

Nie piszę tego, żeby postraszyć. Piszę, bo kontrolę da się wpiąć w trzy miejsca, których w SAP nie da się ominąć.

Kod przechodzi przez edytor, przez transport i przez system produkcyjny. W każdym z tych punktów może zapalić się czerwone światło - zanim zapali się u Waszego klienta.

W edytorze

podatność widać w tej samej minucie, w której powstała, a nie trzy dni przed wdrożeniem

W transporcie

skan zawartości paczki jest warunkiem importu, więc transport bez kontroli uprawnień nie jedzie dalej

Na produkcji

detekcja w czasie rzeczywistym wyłapuje próbę wykorzystania tego, co przeszło przez sito

W SNOK opieramy te trzy warstwy na SecurityBridge. Obserwuję ten produkt od początku istnienia firmy i to jedno z niewielu narzędzi bezpieczeństwa dla SAP, które rośnie razem z platformą.

Trzy rzeczy na poniedziałek

Sprawdźcie, kto dziś pisze u Was kod w SAP. Nie w strukturze organizacyjnej, tylko w systemie - lista autorów obiektów Z z ostatniego półrocza bywa zaskakująca.

Wpiszcie skan bezpieczeństwa w transport, nie w kalendarz. Kontrola, która odbywa się raz na kwartał, w praktyce nie odbywa się wcale.

Ustalcie, co znaczy Wasz podpis pod transportem. Jedno zdanie w polityce, po którym każdy zwalniający wie, czy potwierdza przeczytany kod, czy odbiór funkcjonalny.

Asystent nie stanie przed audytorem

A u Was - co dziś realnie znaczy podpis pod transportem?
Dyskusja pod postem

Jacek Bugajski · jacek.bugajski@snok.ai · snok.ai
Źródła: CodeRabbit XII 2025 (470 pull requestów, badanie dostawcy) · Veracode 2025 · Apiiro IX 2025 · Onapsis, badanie z VI 2026 i Threat Research · SAP i SAP Community

SNOK