

Optymalizacja SDLC w modelu Agentic AI – z wbudowanym governance i ciągłą walidacją na każdym etapie wytwarzania oprogramowania

AI Delivery Operating System – PractIQ na Red Hat OpenShift

Jak zbudować powtarzalny i skalowalny model Agentic AI w całym SDLC, rozwiązując problem governance i walidacji na każdym etapie wytwarzania oprogramowania?



"AI nie skaluje się poprzez dodawanie kolejnych narzędzi, ale budowanie modelu operacyjnego, który reguluje sposób współpracy ludzi i agentów AI – w nowych projektach, istniejących systemach i modernizacji starszych systemów".

Streszczenie dla kadry zarządzającej

Działy IT dużych organizacji klasy enterprise inwestują w sztuczną inteligencję — ale większość z nich nie jest w stanie przełożyć tych inwestycji na systemowe, przewidywalne rezultaty. Narzędzia AI zwiększają produktywność pojedynczych inżynierów, ale nie rozwiązują głębszego problemu: braku modelu jednolitego modelu, który umożliwia efektywną współpracę ludzi i agentów AI w skali całej organizacji.

Problem ma jednak dwa powiązane ze sobą wymiary. Pierwszy to brak governance i standardów dla nowej rzeczywistości: organizacje nie wiedzą, jak wdrożyć AI w sposób kontrolowany, powtarzalny i skalowalny — jak zdefiniować reguły pracy agentów, standaryzować workflowy i zapewnić spójność wyników między projektami i zespołami.

Drugi to spuścizna dwóch dekad pracy w zwinnych metodykach. Era Agile przyniosła szybkość i iteracyjność — ale kosztem architektury, dokumentacji i standaryzacji. Organizacje zbudowały setki aplikacji bez spójnej dokumentacji technicznej, z wiedzą architektoniczną rozproszoną i nieskodyfikowaną oraz z narastającym długiem technologicznym. Gdy AI wchodzi do procesu wytwórczego, ten dług staje się krytyczną barierą: agenci AI nie mogą efektywnie pracować bez kontekstu — bez wiedzy o tym, co system robi, jak jest zbudowany i jakie ma zależności. Organizacje, które chcą wdrożyć AI systemowo, muszą najpierw odbudować to, co przez lata było pomijane.

Bez jasnych standardów, reguł, ustrukturyzowanej wiedzy, dokumentacji procesów gotowych do wykorzystania przez AI — sztuczna inteligencja pozostaje eksperymentem, a nie strategicznym narzędziem zapewniającym przewagę konkurencyjną.

Niniejszy dokument opisuje nową kategorię rozwiązań — AI Delivery Operating System — i przedstawia, w jaki sposób PractIQ firmy Inteca realizuje tę koncepcję w praktyce, działając na platformie Red Hat OpenShift. Jest on adresowany do architektów enterprise, dyrektorów technicznych i liderów IT odpowiedzialnych za strategię adopcji AI w dużych organizacjach.

Kluczowe wnioski są następujące. Po pierwsze, główną barierą systemowej adopcji AI w działach IT nie jest brak technologii, lecz brak governance i operacyjnego modelu pracy z agentami AI. Po drugie, istniejące rozwiązania rynkowe adresują wyłącznie fragmenty tego problemu — brakuje platformy, która integruje governance, standaryzację procesów SDLC, orkiestrację agentów i automatyzację praktyk IT w jednym miejscu. PractIQ z Red Hat OpenShift wypełnia te luki w trzech scenariuszach: dla nowych projektów (Greenfield), dla istniejących systemów wymagających odtworzenia dokumentacji i architektury (Brownfield) oraz dla modernizacji i migracji legacy (Modernization) — dostarczając kompletne środowisko do systemowej adopcji AI niezależnie od punktu startowego organizacji.

CZĘŚĆ I: KONTEKST RYNKOWY I NOWA KATEGORIA

Jak zmienia się wytwarzanie oprogramowania

Wytwarzanie oprogramowania przeszło w ciągu ostatnich trzech dekad trzy fundamentalne transformacje. Każda z nich wymagała nie tylko nowych narzędzi, ale przede wszystkim nowego modelu operacyjnego – nowego sposobu myślenia o tym, jak ludzie, procesy i technologie współpracują ze sobą.

Pierwsza transformacja to przejście od modeli kaskadowych do metodyk zwinnych. Waterfall zakładał, że wymagania można zdefiniować w całości na początku projektu, a następnie realizować je sekwencyjnie przez miesiące. Agile zakwestionował ten paradygmat – wprowadzając iteracyjność, krótkie sprinty, bliską współpracę z klientem i gotowość do zmiany. Nowy model operacyjny wymagał nowych ról, nowych ceremonii i nowego sposobu myślenia o planowaniu. Organizacje, które go przyjęły, zyskały szybkość i elastyczność. Organizacje, które go zignorowały, stopniowo traciły zdolność do konkurencyjności.

Druga transformacja to rozszerzenie paradygmatu Agile na obszar wdrożeń – DevOps. Metodyki zwinne rozwiązały problem planowania i iteracji w wytwarzaniu oprogramowania, ale pozostawiły nierozwiązany problem zarządzania konfiguracją i wdrożeń – jak sprawnie i bezpiecznie dostarczać oprogramowanie na środowiska produkcyjne? DevOps odpowiedział na to pytanie, wprowadzając automatyzację CI/CD, infrastrukturę jako kod, kulturę współpracy między development i operations oraz ciągłe monitorowanie. Cykle delivery skróciły się z tygodni do dni i godzin. Jednak jeden element pozostał niezmieniony: to ludzie pisali kod, tworzyli dokumentację, projektowali architekturę i wykonywali testy.

Trzecia transformacja – ta, której jesteśmy dziś świadkami – zmienia właśnie ten element. Sztuczna inteligencja przestaje być narzędziem wspomagającym pracę inżynierów i staje się aktywnym uczestnikiem procesu twórczego. Agenci AI analizują wymagania, tworzą dokumentację techniczną, projektują architekturę, generują kod i wykonują testy. Według badań Gartner, do 2028 roku 75% inżynierów oprogramowania będzie korzystać z narzędzi AI do kodowania, w

porównaniu z mniej niż 10% w 2023 roku. McKinsey szacuje, że generatywna AI może zautomatyzować do 30% aktywności w typowych projektach wytwarzania oprogramowania.

Dane te opisują jednak wyłącznie skalę zjawiska — nie odpowiadają na pytanie, które jest dziś kluczowe dla liderów IT: jak sprawić, żeby ta zmiana przyniosła systemowe korzyści organizacji, a nie tylko lokalną poprawę produktywności wybranych inżynierów?

Historia daje jednoznaczną odpowiedź: każda z poprzednich transformacji wymagała nowego modelu operacyjnego. Agile wymagał nowych ról, ceremonii i podejścia do planowania. DevOps wymagał nowych narzędzi, kultury współpracy i automatyzacji pipeline'ów. Era AI nie jest wyjątkiem — wymaga nowego podejścia do governance, standaryzacji i zarządzania współpracą ludzi i agentów. Organizacje, które zbudują ten model jako pierwsze, uzyskają przewagę analogiczną do tej, którą uzyskały firmy, które jako pierwsze wdrożyły DevOps.

Dlaczego adopcja AI staje w miejscu

Działy IT dużych organizacji klasy enterprise stoją dziś przed realnym wyzwaniem: jak wdrożyć AI w sposób systemowy, rozwiązać nawarstwiające się przez lata problemy z governance i sprawić, aby AI działała powtarzalnie i przewidywalnie w skali całej organizacji?

Wbrew pozorom, bariera nie jest technologiczna. Narzędzia AI są dostępne, modele językowe osiągają imponujące wyniki, a koszt ich użycia systematycznie maleje. Problem leży gdzie indziej. Obserwacje z projektów prowadzonych przez Inteca w środowiskach enterprise wskazują na pięć powiązanych ze sobą wzorców, które blokują systemową adopcję.

Fragmentaryczna adopcja i brak standardów. Różne zespoły adoptują AI niezależnie od siebie, używając różnych narzędzi, różnych modeli i różnych podejść. Powstają izolowane wyspy efektywności — ale brak wspólnych standardów uniemożliwia replikowanie sukcesów w skali organizacji. IDC raportuje, że 64% organizacji wskazuje brak spójnych standardów i procesów jako główną barierę w skalowaniu inicjatyw AI.

Architektura i dane nieprzygotowane na AI. Agenci AI potrzebują kontekstu — aktualnej dokumentacji technicznej, spójnej architektury systemów, ustrukturyzowanej wiedzy domenowej. Tymczasem w większości dużych organizacji dokumentacja jest przestarzała lub fragmentaryczna, architektura niespójna, a wiedza rozproszona między ludźmi i systemami. Bez tzw. AI-ready knowledge base agenci działają w próżni — generują kod niezgodny z architekturą, powtarzają błędy i wymagają ciągłej korekty ze strony inżynierów.

Agentic coding bez governance. Agentic coding — model, w którym agenci AI samodzielnie wykonują sekwencje zadań programistycznych — zwiększa szybkość wytwarzania, ale jednocześnie wprowadza nowy rodzaj nieprzewidywalności. Bez jasno zdefiniowanych reguł, standardów kodowania i mechanizmów walidacji każdy projekt wygląda inaczej. Jakość kodu generowanego przez AI jest zmienna, koszty nieoczekiwanie rosną, a ryzyko długu technologicznego wzrasta zamiast maleć.

Brak skalowalności. To, co działa w jednym zespole pilotażowym, rzadko replikuje się w skali całego działu IT. Sukcesy pozostają lokalne, bo nie istnieje platforma, która pozwoliłaby standaryzować i dystrybuować sprawdzone praktyki pracy z AI między projektami, zespołami i lokalizacjami. Według raportu Forrester z 2024 roku, tylko 17% organizacji, które wdrożyły pilotaże AI w IT, zdołało skutecznie przeskalować je poza pierwotny zespół.

Dług odziedziczony po epoce Agile. Era Agile przyniosła szybkość i iteracyjność — ale kosztem architektury, dokumentacji i standaryzacji. Przez dwie dekady organizacje budowały setki aplikacji bez spójnej dokumentacji technicznej, z wiedzą architektoniczną rozproszoną i nieskodyfikowaną oraz z narastającym długiem technologicznym. Gdy AI wchodzi do procesu wytwórczego, ten dług staje się krytyczną barierą. Jak ujmuje to Craig Adam w *"Agile is Out, Architecture is Back"* (Medium, 2025): *"Agile taught us speed. AI requires architecture."* Agenci AI nie mogą efektywnie pracować bez kontekstu — bez wiedzy o tym, co system robi, jak jest zbudowany i jakie ma zależności. Organizacje, które chcą wdrożyć AI, muszą najpierw odbudować to, co utraciły w erze Agile.

Konsekwencje biznesowe tych problemów są wymierne. Dłuższy time-to-market mimo rosnących inwestycji w AI. Niespójna jakość produktów i rosnące koszty poprawek. Niemożność wykazania ROI z inicjatyw AI. Oraz — być może najpoważniejsza konsekwencja długoterminowa — rosnące ryzyko utraty kontroli nad procesem wytwórczym w momencie, gdy AI staje się jego kluczowym uczestnikiem.

Dlaczego istniejące rozwiązania nie wystarczają

Rynek narzędzi AI dla działów IT jest dziś rozwinięty i dynamicznie rosnący. Jednak analiza dostępnych rozwiązań ujawnia strukturalną lukę: każde z nich adresuje wybrany fragment problemu, żadne nie oferuje systemowego podejścia do governance i automatyzacji całego SDLC.

Narzędzia AI dla developerów — takie jak GitHub Copilot, Cursor, Claude Code czy Codex — skutecznie zwiększają produktywność pojedynczych programistów przy pisaniu kodu. Nie oferują jednak żadnego mechanizmu standaryzacji, governance ani zarządzania współpracą w skali zespołu czy organizacji. Są narzędziami dla jednostek, nie dla organizacji.

Platformy automatyzacji klasy enterprise — UiPath, SS&C Blue Prism — koncentrują się na automatyzacji procesów biznesowych i administracyjnych. Nie są zaprojektowane z myślą o SDLC, nie rozumieją specyfiki pracy inżynierów i nie oferują mechanizmów governance dla agentów AI pracujących nad kodem, architekturą czy dokumentacją techniczną.

Platformy AI governance — IBM Watsonx jako wiodący przykład — oferują zaawansowane narzędzia do zarządzania cyklem życia modeli AI, monitorowania i zapewnienia zgodności. Nie dotyczą jednak warstwy procesowej — nie standaryzują sposobu pracy z AI, nie automatyzują SDLC i nie zapewniają orkiestracji agentów w procesie wytwórczym.

Platformy agentic AI — takie jak Langchain — skupiają się na orkiestracji agentów i automatyzacji zadań. Brakuje im jednak warstwy standaryzacji praktyk IT, przygotowania dokumentacji i architektury pod AI oraz podejścia end-to-end do całego SDLC.

Wynik tej fragmentaryzacji jest przewidywalny: organizacje, które próbują złożyć kompletne rozwiązanie z dostępnych narzędzi punktowych, napotykają na problemy z integracją, spójnością i governance. Nikt nie zarządza całością. Brakuje warstwy spinającej — rozwiązania, które organizuje sposób pracy działu IT w erze AI: standaryzuje praktyki, zarządza agentami, automatyzuje SDLC i zapewnia governance na każdym etapie procesu wytwórczego.

Nowa kategoria: AI Delivery Operating System

Tak jak DevOps zdefiniował nowy model operacyjny w erze ciągłej integracji i ciągłego dostarczania era AI wymaga własnej kategorii platformowej. Proponujemy nazwać ją AI Delivery Operating System — w skrócie AI-DOS.

AI Delivery Operating System to platforma operacyjna, która standaryzuje, zarządza i automatyzuje sposób pracy działów IT z AI w całym cyklu życia oprogramowania. Analogia z systemem operacyjnym jest tu celowa: tak jak OS zarządza zasobami komputera i umożliwia działanie aplikacji w przewidywalny, kontrolowany sposób, AI-DOS zarządza współpracą ludzi, agentów AI, danych i procesów — zapewniając, że całość działa spójnie, bezpiecznie i skalowalnie.

AI Delivery Operating System integruje cztery warstwy, które dotychczas funkcjonowały niezależnie od siebie. Warstwa governance definiuje standardy i reguły pracy z agentami AI — co mogą robić, jak powinny działać i jak są walidowane. Warstwa procesowa dostarcza ustrukturyzowane workflowy SDLC oparte na sprawdzonych praktykach branżowych — od analizy wymagań po testy akceptacyjne. Warstwa orkiestracji agentów zarządza współpracą ludzi i AI w ramach workflowów — przypisuje role, zarządza kontekstem i zapewnia human-in-the-loop control. Warstwa automatyzacji wykonuje praktyki IT w sposób powtarzalny i mierzalny — generując artefakty, kod, dokumentację i testy zgodne ze standardami organizacji.

Organizacje, które wdrożą AI Delivery Operating System, zyskują zdolność, której żadne narzędzie punktowe nie jest w stanie zapewnić: systemową, skalowalną adopcję AI — od pojedynczego zespołu pilotażowego po całą organizację, z zachowaniem spójności, jakości i kontroli na każdym etapie.

CZĘŚĆ II: ARCHITEKTURA I INTEGRACJA Z RED HAT OPENSIFT

PractIQ jako AI Delivery Operating System

PractIQ firmy Inteca jest pierwszą platformą klasy enterprise, która w pełni realizuje koncepcję AI Delivery Operating System. Zaprojektowana na podstawie wieloletnich doświadczeń Inteca w projektach wdrożeń SDLC governance dla sektorów finansowego, bankowego, ubezpieczeniowego i przemysłowego, PractIQ dostarcza gotowe środowisko operacyjne dla działów IT gotowych na systemową adopcję AI.

Platforma opiera się na koncepcji praktyki — rozumianej nie jako sztywna sekwencja kroków procesowych, lecz jako powtarzalny, ukierunkowany na efekt sposób działania, który może być realizowany przez ludzi, agentów AI lub ich kombinację. To podejście — *practice-first engineering* — stanowi fundament filozofii PractIQ i odróżnia go od platform zorientowanych na automatyzację zadań.

PractIQ oferuje dwa komplementarne modele działania. Model Core dostarcza gotowe, sprawdzone praktyki IT dostosowane do konkretnych technologii i frameworków — organizacja może zacząć pracę natychmiast, bez konieczności projektowania procesów od zera. Model Forge pozwala zaprojektować i zautomatyzować własne praktyki, dopasowane do jej specyfiki, standardów wewnętrznych i ekosystemu narzędzi. Oba modele mogą funkcjonować równolegle, co umożliwia przyrostowe wdrożenie — od gotowych praktyk do w pełni spersonalizowanego środowiska.

Platforma dostarcza pięć gotowych praktyk SDLC, które razem tworzą zintegrowany, end-to-end system automatyzacji procesu wytwórczego.

AI Analysis to praktyka wspomaganej przez AI analizy wymagań biznesowych i technicznych. Agenci AI strukturyzują, walidują i uzupełniają wymagania względem bazy wiedzy i wzorców branżowych, zapewniając kompletność i spójność już na starcie projektu. Praktyka eliminuje niekompletne lub sprzeczne wymagania — jedno z najczęstszych źródeł kosztownych błędów na późniejszych etapach SDLC.

AI Architecture standaryzuje projektowanie architektury w złożonym krajobrazie architektonicznym przy wsparciu agentów AI. Automatycznie generuje dokumentację architektoniczną w standardach C4 Model, kracając czas tworzenia modeli architektury o ponad 90% przy jednoczesnym zachowaniu spójności i zgodności ze standardami organizacji.

AI Application Design automatyzuje tworzenie i walidację dokumentacji technicznej i funkcjonalnej dla pojedynczego komponentu. Przekształca dokumentację z jednorazowego, ręcznego zadania w powtarzalną, skalowalną praktykę. Wytworzone modele są przygotowane do dalszego wykorzystania przez agentów AI developmentu i testów — stając się elementem żywej bazy wiedzy organizacji.

AI Development standaryzuje i kontroluje proces agentic coding. Definiuje role agentów, reguły działania i workflow współpracy z inżynierami — eliminując losowość wyników i zapewniając spójną jakość generowanego kodu niezależnie od projektu czy zespołu. Praktyka wspiera szeroki zakres technologii: Angular, Spring Boot, React, .NET i inne.

AI Quality Assurance osadza ciągle zapewnienie jakości w całym SDLC. Automatyzuje tworzenie i uruchamianie testów weryfikacyjnych E2E dla całych ścieżek użytkownika i integracji między komponentami — z pełnym traceability do wymagań.

Wszystkie praktyki działają w modelu human-in-the-loop: agenci AI wykonują zadania, ale człowiek zachowuje pełną kontrolę i możliwość ingerencji na każdym etapie. To kluczowy element architektury PractIQ — AI działa według standardów, nie improwizuje.

Architektura techniczna PractIQ

PractIQ to modularny system pięciu warstw, zaprojektowany z myślą o środowiskach enterprise — z pełną kontrolą, traceability i możliwością dopasowania do istniejącego ekosystemu IT organizacji.

Human-in-the-Loop Layer zapewnia ciągłą kontrolę człowieka nad działaniami agentów AI. Warstwa udostępnia interfejsy konwersacyjne dostępne przez

przeglądarkę internetową oraz integrację z Visual Studio Code, umożliwiając inżynierom naturalną interakcję z systemem podczas pracy z produktami praktyk. Kluczową funkcją tej warstwy jest możliwość akceptacji, modyfikacji lub cofania działań automatycznych proponowanych przez agentów — w trybie nadzorowanym, który gwarantuje, że człowiek pozostaje odpowiedzialny za ostateczny kształt artefaktów.

Knowledge Layer gromadzi, organizuje i udostępnia wiedzę domenową i techniczną niezbędną do realizacji praktyk. Warstwa zawiera definicje praktyk, reguły postępowania i elementy wiedzy — szablony, wzorce architektoniczne, standardy organizacyjne i przykłady. Dzięki tej warstwie agenci AI nie operują na ogólnej wiedzy modelu językowego, lecz na wiedzy dopasowanej do realiów konkretnej organizacji. Warstwa umożliwia zarówno wykorzystanie praktyk dostarczanych przez Inteca, jak i budowanie własnego repozytorium wiedzy organizacyjnej.

Data Context Layer zapewnia pełny kontekst organizacyjny dla agentów AI poprzez integracje z kluczowymi systemami IT. Warstwa integruje się z JIRA w celu odczytu i modyfikacji backlogu, zadań i błędów; z Enterprise Architect w celu dostępu do diagramów architektury i modeli domenowych; oraz z narzędziami dokumentacyjnymi takimi jak Confluence czy Notion. Na podstawie danych z tych systemów warstwa buduje graf wiedzy, który umożliwia agentom podejmowanie decyzji zgodnych z rzeczywistym stanem projektu i organizacji.

LLM Execution Layer odpowiada za wykonywanie zadań opartych na generatywnej AI. Warstwa wspiera wiele modeli językowych — OpenAI GPT, Anthropic Claude, Gemini oraz modele lokalne — i dynamicznie dobiera model do zadania. Architektura warstwy pozwala na rozszerzenie o modele fine-tuned na danych organizacyjnych z wykorzystaniem adapterów LoRA, co optymalizuje koszty przy zachowaniu jakości wyników i zapewnia suwerenność danych w środowiskach o podwyższonych wymaganiach bezpieczeństwa.

Orchestration Layer to centralny komponent integrujący pozostałe warstwy i zarządzający przepływem danych oraz zadań w ramach praktyk. Warstwa

odpowiada za konfigurację i budowę serwerów narzędzi (MCP Servers), orkiestrację przepływu między warstwami, zarządzanie pamięcią konwersacji agentów, ekstrakcję wiedzy z systemów domenowych (Knowledge Ingestion Pipeline) oraz monitoring i logowanie wszystkich działań. Warstwa orkiestracji jest również punktem wejścia dla przyszłej integracji A2A (Agent-to-Agent) – protokołu umożliwiającego współpracę między agentami z różnych platform i organizacji.

Integracja z Red Hat OpenShift

Red Hat OpenShift stanowi rekomendowane środowisko produkcyjne dla PractIQ – zapewniając enterprise-grade bezpieczeństwo, skalowalność i gotowość produkcyjną wymaganą przez duże organizacje klasy enterprise.

Dlaczego OpenShift. Wdrożenie PractIQ w środowisku enterprise wymaga platformy, która spełnia wymagania korporacyjne w zakresie bezpieczeństwa, compliance, zarządzania dostępem i skalowalności – bez konieczności budowania tej infrastruktury od zera. Red Hat OpenShift jako wiodąca platforma Kubernetes klasy enterprise dostarcza wszystkie te elementy w jednym, zintegrowanym środowisku. Dla organizacji, które już korzystają z OpenShift jako platformy aplikacyjnej, wdrożenie PractIQ oznacza rozszerzenie istniejącego ekosystemu o warstwę AI governance i automatyzacji SDLC – bez konieczności wdrażania nowej infrastruktury.

Jak PractIQ działa na OpenShift. Wszystkie komponenty PractIQ są konteneryzowane i wdrażane jako workloady OpenShift, zarządzane przez natywne mechanizmy platformy. Integracja obejmuje szereg produktów Red Hat, które razem tworzą kompletne środowisko uruchomieniowe dla PractIQ.

Red Hat OpenShift stanowi fundament platformy – zapewnia orkiestrację kontenerów, automatyczne skalowanie, izolację środowisk oraz wbudowane mechanizmy bezpieczeństwa i compliance. Orchestration Layer z serwerami MCP działa jako zestaw mikroservisów zarządzanych przez OpenShift, korzystając z wbudowanych mechanizmów service discovery, load balancingu i automatycznego restartowania.

Red Hat OpenShift AI jest wykorzystywany do uruchamiania i zarządzania modelami językowymi bezpośrednio w klastrze — w tym modeli lokalnych i modeli fine-tuned z adapterami LoRA). Zapewnia to pełną suwerenność danych i eliminuje konieczność wysyłania wrażliwych danych organizacyjnych do zewnętrznych API. OpenShift AI dostarcza również środowisko do fine-tuningu modeli na danych organizacyjnych, co optymalizuje jakość i koszty przy intensywnym użyciu.

Red Hat OpenShift Dev Spaces zapewnia środowisko programistyczne zintegrowane z platformą — umożliwiając inżynierom pracę z agentami AI PractIQ bezpośrednio w przeglądarce, bez konieczności lokalnej konfiguracji środowisk. Jest to szczególnie istotne w organizacjach z rozproszonymi zespołami lub wymaganiami bezpieczeństwa wykluczającymi pracę na lokalnych maszynach.

Red Hat Developer Hub służy jako centralne miejsce dostępu do praktyk i narzędzi PractIQ dla zespołów inżynierskich — integrując się z istniejącym portalem deweloperskim organizacji i zapewniając spójne doświadczenie użytkownika niezależnie od projektu czy zespołu.

Red Hat Developer Hub Workflow umożliwia uruchamianie praktyk PractIQ w modelu bezserwerowym — w szczególności dla zadań orkiestracji agentów i przetwarzania zdarzeń, które wymagają elastycznego skalowania od zera do wielu instancji w zależności od obciążenia. Redukuje to koszty operacyjne w przypadku nieregularnego lub sezonowego wykorzystania platformy.

Human-in-the-Loop Layer jest dostępny przez OpenShift Routes z wbudowanym TLS, zapewniając bezpieczny dostęp dla użytkowników końcowych bez dodatkowej konfiguracji reverse proxy.

Korzyści z integracji z OpenShift. Automatyczne skalowanie komponentów PractIQ w zależności od obciążenia zapewnia stabilność działania nawet przy intensywnym użyciu przez wiele zespołów równocześnie. Izolacja środowisk development, test i produkcji zgodna z korporacyjnymi standardami governance eliminuje ryzyko niezamierzonych modyfikacji środowisk produkcyjnych. Wbudowane mechanizmy RBAC, network policies i skanowania

obrazów kontenerowych zapewniają bezpieczeństwo zgodne z wymaganiami enterprise bez dodatkowego nakładu pracy. Integracja z istniejącym CI/CD pipeline opartym na OpenShift Pipelines lub Tekton umożliwia automatyczne wdrażanie aktualizacji PractIQ w ramach standardowych procesów DevOps organizacji.

Modele wdrożenia. PractIQ na OpenShift może być wdrożony w trzech konfiguracjach. Wdrożenie on-premise w klastrze OpenShift zapewnia pełną kontrolę nad infrastrukturą i danymi – optymalne dla organizacji z najwyższymi wymaganiami bezpieczeństwa, takich jak banki i instytucje finansowe. Wdrożenie w modelu hybrid cloud łączy lokalny klaster OpenShift dla wrażliwych komponentów z zasobami chmurowymi dla komponentów wymagających elastycznego skalowania. Wdrożenie multi-cloud na zarządzanych usługach OpenShift (ROSA na AWS, ARO na Azure) umożliwia szybkie uruchomienie bez inwestycji w infrastrukturę własną.

CZĘŚĆ III: USE CASES I WARTOŚĆ BIZNESOWA

Scenariusze zastosowania

PractIQ z Red Hat OpenShift adresuje pełne spektrum wyzwań AI-driven delivery — od nowych projektów, przez istniejące systemy, po modernizację legacy. Trzy scenariusze odpowiadają trzem rzeczywistym sytuacjom, z którymi mierzą się działy IT dużych organizacji klasy enterprise.

Scenariusz 1: Forward Engineering — nowe projekty (Greenfield)

Organizacje budujące nowe systemy lub rozwijające istniejące produkty cyfrowe napotykają na fundamentalne wyzwanie: jak sprawić, żeby każdy nowy projekt korzystał z AI w sposób spójny, kontrolowany i powtarzalny — niezależnie od zespołu, technologii czy lokalizacji?

Bez ustrukturyzowanego modelu operacyjnego każdy projekt startuje od zera. Inżynierowie eksperymentują z agentami AI na własną rękę, wyniki są nieprzewidywalne, a jakość kodu zmienna. Dział IT pracuje szybciej, ale nie lepiej.

PractIQ dostarcza gotowy, AI-powered model operacyjny dla pełnego SDLC. Pięć zautomatyzowanych praktyk — Analysis, Design, Architecture, Development, QA — działa jako zintegrowany system, w którym każda praktyka korzysta z artefaktów wytworzonych przez poprzednią. Agenci AI wykonują zadania według zdefiniowanych standardów, a inżynierowie zachowują pełną kontrolę w modelu human-in-the-loop.

Efekty mierzone w projektach Inteca: time-to-market poprawiony o 40%, 60% wzrost efektywności zespołów, ponad 90% zgodności z governance, 95% kodu generowanego przez AI pod nadzorem inżynierów.

Scenariusz 2: Backward Engineering — istniejące systemy (Brownfield)

Wyobraźmy sobie dział IT zarządzający setkami aplikacji — z których większość działa, ale nikt nie potrafi powiedzieć dokładnie jak. Każda próba modernizacji zaczyna się od tego samego pytania: skąd wiedzieć, co można zmienić, nie psując tego, co działa? Gdy do tego obrazu dochodzą agenci AI — którzy

potrzebują precyzyjnego kontekstu, żeby generować użyteczne wyniki — pytanie to staje się pilne.

PractIQ rozwiązuje ten problem poprzez automatyczne odtwarzanie brakującej dokumentacji i architektury bezpośrednio z kodu źródłowego. Agenci AI analizują istniejące repozytoria, identyfikują strukturę domenową, mapują zależności między systemami i generują aktualną dokumentację w standardach C4 Model i ADR. Wynik trafia do Knowledge Layer PractIQ — tworząc AI-ready knowledge base, który staje się fundamentem dla dalszej pracy agentów.

Organizacja przechodzi od stanu, w którym nikt w pełni nie rozumie, co aplikacje robią, do stanu, w którym każdy agent — i każdy nowy inżynier — ma dostęp do aktualnej, ustrukturyzowanej wiedzy o systemach.

Scenariusz 3: Modernizacja i eliminacja vendor lock-in (Legacy & Modernization)

Gdy organizacja rozumie już swoje systemy — dzięki odtworzonej dokumentacji i architekturze z Scenariusza 2 — PractIQ umożliwia następny krok: aktywną modernizację. Przepisanie aplikacji w nowych technologiach, zastąpienie systemów kontrolowanych przez zewnętrznych dostawców, redukcja długu technologicznego — wszystko to bez ryzyka utraty wiedzy domenowej.

Tradycyjne podejście do modernizacji jest obarczone wysokim ryzykiem: zespoły przepisują systemy, których nie do końca rozumieją, co prowadzi do pominięcia kluczowej logiki biznesowej i kosztownych błędów. PractIQ eliminuje to ryzyko — agenci AI pracują na bazie wiedzy odtworzonej w poprzednim etapie, generując nowy kod zgodny z domeną biznesową i standardami organizacji.

Trzy kluczowe korzyści tego scenariusza to modernizacja przy minimalnym ryzyku (przepisanie aplikacji w nowych technologiach bez utraty wiedzy domenowej), redukcja długu technologicznego (zastąpienie przestarzałej architektury czystym, AI-driven fundamentem) oraz eliminacja vendor lock-in (odtworzenie i przejęcie kontroli nad systemami zarządzanymi przez zewnętrznych dostawców).

Trzy scenariusze tworzą naturalną ścieżkę dojrzewania: organizacje mogą wejść w dowolnym miejscu — zaczynając od nowych projektów, od porządkowania istniejących systemów, lub od pilnej modernizacji — i postępować według własnego tempa i priorytetów.

Wartość biznesowa

Wdrożenie PractIQ z Red Hat OpenShift przynosi mierzalną wartość biznesową na czterech poziomach — przy czym konkretne korzyści różnią się w zależności od scenariusza, w którym organizacja zaczyna swoją ścieżkę.

Wyniki operacyjne, osiągane przez Inteca we własnym procesie wytwórczym oraz w pierwszych projektach klientów, pokazują skalę możliwych usprawnień w scenariuszu Greenfield. Nakład pracy na wytwarzanie funkcjonalności spada o nawet 80%. Faza analizy i architektury skraca się o ponad 50%. Przepustowość i prędkość zespołów rośnie o 50%. Platforma zapewnia 90% pokrycia testami bez dodatkowego nakładu pracy inżynierów. Time-to-market poprawia się o 40%. 95% kodu jest wytwarzane przez agentów AI pod nadzorem inżynierów, a 80–90% specyfikacji technicznej powstaje automatycznie.

Wartość ze scenariuszy Brownfield i Modernization jest innego rodzaju, ale równie wymierna. Organizacje, które odtworzą dokumentację i architekturę istniejących systemów, zyskują przede wszystkim redukcję ryzyka operacyjnego — decyzje architektoniczne i modernizacyjne podejmowane są na podstawie aktualnej, ustrukturyzowanej wiedzy, a nie domysłów i szczątkowej pamięci instytucjonalnej. Skraca się czas onboardingu nowych inżynierów, którzy zamiast tygodni spędzanych na rekonstruowaniu wiedzy o systemach, otrzymują gotową bazę dokumentacji. Eliminacja vendor lock-in przynosi bezpośrednie korzyści finansowe — uniezależnienie od zewnętrznych dostawców, odzyskanie kontroli nad kosztami licencji i utrzymania oraz możliwość przeprowadzenia modernizacji na własnych warunkach. Według badań Gartner, dług technologiczny pochłania średnio 30–40% budżetu IT w dużych organizacjach — PractIQ dostarcza systematyczne narzędzie do jego redukcji.

Korzyści finansowe wspólne dla wszystkich scenariuszy wynikają z automatyzacji powtarzalnych praktyk IT i eliminacji kosztów pracy odtwórczej. Redukcja poprawek i rework — wynikająca z wyższej jakości i spójności wytwarzanych artefaktów — zmniejsza koszty późnych faz projektu, gdzie błędy są najdroższe. Architektura LLM oparta na dynamicznym doborze modeli i możliwości fine-tuningu z LoRA optymalizuje koszty tokenów. Dla organizacji, które chcą zachować pełną suwerenność danych, wdrożenie modeli lokalnych na OpenShift AI eliminuje zmienne koszty API przy intensywnym użyciu.

Wartość strategiczna PractIQ wynika przede wszystkim z tego, że wszystkie trzy scenariusze działają na jednej, zintegrowanej platformie. Organizacja nie musi wybierać między inwestycją w nowe projekty a porządkowaniem istniejących systemów — PractIQ obsługuje oba jednocześnie, a wiedza wytworzona w scenariuszu Brownfield staje się bezpośrednim wkładem do scenariuszy Greenfield i Modernization. Governance i compliance wbudowane w każdy workflow eliminują ryzyko operacyjne związane z niekontrolowanym użyciem AI. Niezależność od konkretnych dostawców modeli AI redukuje ryzyko vendor lock-in na poziomie platformy. Wdrożenie na Red Hat OpenShift zapewnia, że PractIQ naturalnie wpisuje się w istniejącą strategię platformową organizacji, minimalizując ryzyko wdrożeniowe i koszty integracji.

Model wdrożenia

PractIQ jest zaprojektowany do wdrożenia przyrostowego — organizacja może zacząć od jednego scenariusza i skalować w swoim tempie, unikając ryzyka dużego wdrożenia big-bang.

Etap 1: Prepare. Organizacja przygotowuje się do pracy z AI poprzez porządkowanie architektury, dokumentacji i danych. PractIQ AI Architecture i AI Documentation pozwalają szybko zbudować AI-ready knowledge base — niezależnie od tego, czy punktem startowym są nowe projekty, czy istniejące systemy wymagające odtworzenia dokumentacji.

Etap 2: Standardize. Organizacja wdraża pierwsze praktyki SDLC i standaryzuje sposób pracy z agentami AI. Inżynierowie uczą się pracować w modelu human-in-the-loop, a dział IT wypracowuje wewnętrzne standardy i reguły dla agentów.

Etap 3: Automate. Organizacja wdraża pełny zestaw praktyk SDLC i automatyzuje end-to-end proces wytwórczy. AI Software Factory staje się operacyjną rzeczywistością: agenci AI realizują cały cykl wytwórczy pod nadzorem inżynierów.

Etap 4: Scale. Organizacja rozszerza PractIQ na kolejne działy, projekty, technologie i obszary działalności — w tym na modernizację legacy i eliminację vendor lock-in. Model Forge umożliwia tworzenie własnych praktyk dopasowanych do specyfiki nowych obszarów.

Każdy etap ma jasno zdefiniowane wejście, wyjście i efekt biznesowy — co umożliwia mierzenie postępów i uzasadnianie kolejnych inwestycji przed zarządem organizacji.

Podsumowanie

Era AI zmienia wytwarzanie oprogramowania tak samo głęboko, jak kiedyś zmieniły je metodyki zwinne i DevOps. Jednak w odróżnieniu od poprzednich transformacji, zmiana ta dotyczy nie tylko narzędzi i procesów — dotyczy fundamentalnego pytania o to, jak ludzie i maszyny współpracują w tworzeniu oprogramowania.

Jednocześnie organizacje wchodzą w tę transformację z bagażem, który poprzednia era — era Agile — pozostawiła im w spadku: setki aplikacji z wiedzą architektoniczną rozproszoną i nieskodyfikowaną, bez aktualnej dokumentacji, z narastającym długiem technologicznym. Systemowa adopcja AI wymaga zmierzenia się z oboma wyzwaniami naraz — nie można budować przyszłości na fundamencie, którego nikt w pełni nie rozumie.

Działy IT dużych organizacji klasy enterprise, które próbują odpowiedzieć na te wyzwania wyłącznie narzędziami punktowymi, napotykają na ścianę: fragmentaryczne eksperymenty, nieprzewidywalne wyniki, brak skalowalności i rosnące ryzyko utraty kontroli nad procesem wytwórczym.

AI Delivery Operating System to odpowiedź na tę lukę — nie jako kolejne narzędzie, ale jako nowa warstwa operacyjna, która organizuje sposób pracy działu IT w erze AI. PractIQ realizuje tę koncepcję w praktyce, dostarczając kompletne środowisko dla trzech scenariuszy: budowy nowych systemów z AI od podstaw (Greenfield), odtworzenia wiedzy o istniejących aplikacjach i przygotowania ich do pracy z agentami (Brownfield) oraz modernizacji i eliminacji vendor lock-in systemów legacy (Modernization). Wszystkie trzy scenariusze działają na jednej, zintegrowanej platformie — wiedza wytworzona w każdym z nich zasila pozostałe.

Red Hat OpenShift zapewnia enterprise-grade fundament dla tej transformacji — bezpieczeństwo, skalowalność i gotowość produkcyjną wymaganą przez organizacje działające w regulowanych branżach i złożonych środowiskach IT.

Razem, PractIQ i Red Hat OpenShift tworzą kompletne rozwiązanie dla działów IT gotowych na następny krok. Bo AI nie skaluje się przez dodawanie kolejnych narzędzi. Skaluje się przez zbudowanie modelu operacyjnego, który zarządza tym, jak ludzie i agenci AI pracują razem — w nowych projektach, istniejących systemach i modernizacji legacy.

O firmach

Inteca to firma IT z siedzibą we Wrocławiu, założona w 2011 roku, zatrudniająca ponad 60 inżynierów i konsultantów. Inteca specjalizuje się w projektowaniu, wdrażaniu i utrzymaniu platform IT klasy enterprise, świadcząc usługi z obszaru Identity and Access Management, Business Automation, Cloud, integracji danych, DevOps oraz Enterprise Architecture. Firma obsługuje klientów z sektora finansowego, bankowego, ubezpieczeniowego, przemysłowego i energetycznego – w tym Santander, BNP Paribas, Credit Agricole, Allianz, Generali, Orlen i Tauron.

Inteca jest Red Hat Advanced Business Partner z wieloletnim doświadczeniem w konsultingu, wdrożeniach i zarządzanych usługach dla Red Hat OpenShift i Kubernetes. PractIQ stanowi rozszerzenie tej specjalizacji – jako platforma AI governance i automatyzacji SDLC komplementarna wobec ekosystemu Red Hat.

Więcej informacji: inteca.com | practiq.tech

Red Hat jest wiodącym światowym dostawcą rozwiązań open source dla przedsiębiorstw, oferującym sprawdzone technologie dla środowisk Linux, chmury, kontenerów i Kubernetes. Red Hat OpenShift to wiodąca platforma Kubernetes klasy enterprise, zapewniająca organizacjom bezpieczne, skalowalne środowisko do uruchamiania aplikacji kontenerowych w środowiskach on-premise, hybrid cloud i multi-cloud. Red Hat pomaga klientom integrować nowe i istniejące technologie IT, rozwijać aplikacje w chmurze oraz automatyzować i zarządzać złożonymi środowiskami.

Więcej informacji: redhat.com