Ekonomia długu technicznego i architektura sprawczości w erze AI: analiza koncepcji Alexio Cassaniego „Code Revealed”

AUTOR

Fundacja Dobre Państwo

STRESZCZENIE / ABSTRACT

Artykuł analizuje paradoks współczesnej inżynierii oprogramowania: choć AI drastycznie obniża koszt generowania kodu, zwiększa jednocześnie długoterminowy koszt jego posiadania. Autor ostrzega przed zjawiskiem 'Hidden feature Creek' oraz asymetrią bodźców w modelach LLM, które preferują tworzenie nowych rozwiązań nad ich upraszczaniem. Tekst podkreśla różnicę między lokalną poprawnością kodu a globalną stabilnością systemu delivery, przywołując dane z raportu DORA i GitClear. Głównym wnioskiem jest teza, że nadprodukcja zmian może przekroczyć zdolność organizacji do ich metabolizowania, prowadząc do paraliżu systemowego. Kluczowym miernikiem jakości w erze AI nie powinna być zatem ilość wygenerowanego kodu, lecz trwałość wprowadzonych zmian i minimalizacja długu technicznego.

The article analyzes a paradox in modern software engineering: although AI drastically lowers the cost of generating code, it simultaneously increases the long-term cost of owning it. The author warns against the 'Hidden Feature Creep' phenomenon and the asymmetry of incentives in LLM models, which prefer creating new solutions over simplifying existing ones. The text emphasizes the difference between local code correctness and global stability of the delivery system, citing data from DORA and GitClear reports. The main conclusion is that an overproduction of changes may exceed an organization's capacity to metabolize them, leading to systemic paralysis. Therefore, the key quality metric in the AI era should not be the amount of generated code, but the durability of introduced changes and the minimization of technical debt.

Artykuł analizuje paradoks ekonomiczny ery generatywnej sztucznej inteligencji: drastyczny spadek kosztu wytworzenia kodu nie redukuje, lecz potencjalnie zwiększa całkowity koszt posiadania oprogramowania (TCO). Autor stawia tezę, że nadprodukcja taniego, lokalnie poprawnego, ale globalnie niespójnego kodu prowadzi do gwałtownego przyrostu niewidzialnego długu technicznego i erozji kompetencji inżynieryjnych. Tekst dowodzi, że sukces transformacji agentowej nie zależy od samej wydajności modeli LLM, lecz od przejścia z modelu naiwnej automatyzacji na dojrzałą architekturę sprawczości. Wymaga to przedefiniowania roli programisty – z wykonawcy w krytycznego orkiestratora – oraz wprowadzenia rygorystycznego governance i niezależnej walidacji. Kluczowym wyzwaniem staje się zatem nie tyle zwiększenie tempa generowania rozwiązań, co zachowanie suwerenności ludzkiego osądu (phronesis) i odpowiedzialności nad sensem działania systemu, aby technologia służyła jako wzmacniacz ludzkich kompetencji, a nie proteza prowadząca do intelektualnej abdykacji.

SŁOWA KLUCZOWE

dług techniczny | architektura sprawczości | AI code churn | hidden feature creep | agentowa AI | governance w software engineeringu | epistemiczna niezależność testów | technical debt budget | martwa poprawność | human-in-the-loop (HITL) | metodologia prawdziwościowa | asymetria bodźców w LLM | produktywność generowania kodu | zarządzanie autonomią agentów

Tani kod zwiększa koszt posiadania oprogramowania

Entropia kodu: dług techniczny, code churn i cena łatwego tworzenia. Poważnym nieporozumieniem w ekonomii oprogramowania ery AI może okazać się przekonanie, że skoro koszt napisania kodu gwałtownie maleje, to maleje również koszt jego posiadania. Jest wręcz przeciwnie. Kod można wygenerować w kilka minut, lecz przez następne dziesięć lat trzeba go rozumieć, testować, aktualizować, zabezpieczać, integrować z kolejnymi komponentami i naprawiać po zmianach w otoczeniu. Generacja jest zdarzeniem; utrzymanie jest procesem.

Jeżeli AI radykalnie przyspiesza pierwsze, nie zmieniając praw ekonomicznych drugiego, organizacja może niezwykle tanio wyprodukować ogromną liczbę przyszłych zobowiązań. W tym miejscu powraca pojęcie długu technicznego. Metafora Warda Cunninghama pozostaje wyjątkowo użyteczna, gdyż pozwala odróżnić świadomy kompromis od zwykłego bałaganu. Rozwiązanie niedoskonałe bywa racjonalne, jeśli umożliwia szybkie wejście na rynek, a organizacja ma świadomość „odsetek", które przyjdzie później zapłacić. Problemem nie jest sam dług, lecz dług niewidoczny, nierozpoznany i niespłacany, który z czasem paraliżuje zdolność systemu do dalszych zmian. Programowanie agentowe może tę niewidzialność pogłębić.

Cassani opisuje kilka charakterystycznych syndromów: mieszanie konwencji i paradygmatów wewnątrz jednego modułu, niezamówione rozszerzanie funkcjonalności, mnożenie plików pomocniczych zamiast upraszczania struktury oraz skłonność do wdrażania rozwiązań lokalnych tam, gdzie bardziej racjonalna byłaby konsolidacja. W jego terminologii pojawia się trafna kategoria hidden feature creep: agent otrzymuje zadanie, lecz w dobrej wierze „ulepsza" również logging, cache, konfigurację albo abstrakcję, których nikt nie zamawiał.

Nie należy traktować tych zachowań jako uniwersalnej właściwości modeli językowych; jest to przede wszystkim obserwacja i kategoria organizacyjna Cassaniego. Wskazuje ona jednak na asymetrię bodźców. Model otrzymuje polecenie wytworzenia rozwiązania, lecz rzadko dostaje premię za decyzję: „niczego nie dodawaj". Generatywna AI jest z natury narzędziem produkującym odpowiedź, podczas gdy dojrzała inżynieria często wymaga umiejętności powstrzymania się od tworzenia.

W tradycyjnym środowisku napisanie dodatkowych pięciuset linii kodu kosztowało czas programisty. Sam wysiłek stanowił naturalny opór przed nadmierną komplikacją. Gdy agent generuje te linie w mgnieniu oka, ekonomiczny koszt dodawania znika z miejsca powstawania decyzji. Nie znika jednak koszt późniejszego czytania, review, testowania i aktualizacji. AI obniża cenę zaciągania długu, ale niekoniecznie cenę jego obsługi. Dlatego najważniejszym miernikiem jakości nie powinna być ilość kodu, który przeszedł testy, lecz trwałość zmiany.

Czy rozwiązanie pozostaje w systemie, czy za miesiąc trzeba je przepisać? Czy tworzy nowe zależności lub zwiększa liczbę wariantów rozwiązujących ten sam problem? Czy kolejny programista potrafi je zmodyfikować bez otwierania dziesięciu dodatkowych plików? Czy system staje się dzięki temu prostszy, czy tylko bardziej rozbudowany?

W tym miejscu należy skorygować jedną z kompilacji obecnych w materiale źródłowym. Notatka łączy raport DORA 2024 z rosnącym code churn, traktując oba zjawiska jako spójną historię o pogarszającej się jakości. Tymczasem oficjalny raport DORA 2024 wykazał, że wzrost adopcji AI o

25 procent wiązał się statystycznie ze spadkiem stabilności dostarczania o 7,2 procent i throughputu o 1,5 procent, ale jednocześnie z poprawą jakości dokumentacji, kodu i szybkości code review. DORA nie interpretuje tego wyniku jako pomiaru code churn. Jakość lokalna kodu i stabilność globalna procesu to dwie różne kwestie. Fragment może być czytelny i technicznie poprawny, a cały system dostarczania może mimo to stać się mniej stabilny, jeśli wzrośnie liczba zmian, wielkość batchy lub presja na szybsze mergowanie. DORA zwraca uwagę, że poprawa pracy deweloperskiej nie przekłada się automatycznie na sprawność całego systemu delivery, podkreślając znaczenie małych partii zmian i solidnych testów.

Ta obserwacja jest kluczowa dla teorii długu technicznego. Możliwe jest bowiem, że AI tworzy kod lokalnie lepszy, ale globalnie przyspiesza tempo zmian ponad zdolność organizacji do ich absorpcji. W takim przypadku problemem nie jest pojedynczy wadliwy fragment, lecz nadprodukcja zmian. System przypomina organizm, który otrzymuje więcej kalorii, niż potrafi metabolizować. Nadwyżka nie znika – zaczyna się odkładać.

Wzrost ilości kodu AI generuje ukryte koszty utrzymania

Inne dane pochodzą od GitClear, komercyjnej firmy analizującej repozytoria. Badanie opublikowane w 2024 roku, obejmujące około 153 milionów zmienionych linii kodu z lat 2020–2023, wskazało na wzrost krótkoterminowego churnu oraz coraz większy udział kodu dodawanego i kopiowanego przy jednoczesnym spadku udziału kodu przenoszonego lub ponownie wykorzystywanego. Firma prognozowała wówczas podwojenie udziału churnu w 2024 roku względem poziomu bazowego z 2021 roku. Kolejna analiza GitClear, obejmująca już 211 milionów linii kodu z lat 2020–2024, potwierdziła trend wzrostowy klonowania: udział linii oznaczonych jako copy/paste wzrósł według ich metodologii z 8,3% w 2021 roku do 12,3% w 2024 roku, podczas gdy liczba zmian związanych z refaktoryzacją spadła.

Są to interesujące dane longitudinalne, lecz wymagają one dyscypliny metodologicznej. GitClear jest producentem komercyjnego narzędzia do mierzenia jakości kodu, a analiza zmian czasowych nie jest równoznaczna z eksperymentem dowodzącym, że to AI spowodowała wszystkie obserwowane zjawiska. Szczególnej ostrożności wymaga pojawiająca się w notatce liczba „ośmiokrotnego wzrostu code clones”. Aktualne materiały marketingowe firmy posługują się już sformułowaniem o ośmiokrotnym wzroście duplikacji od czasu upowszechnienia asystentów AI, jednak ich raport z 2025 roku opisuje między innymi czterokrotny wzrost częstości określonych bloków klonowanego kodu i jednocześnie zmianę udziału copy/paste z 8,3 do 12,3 procent.

Nie są to te same miary. Dlatego rzetelny artykuł naukowy nie powinien przekształcać jednego chwytliwego mnożnika w uniwersalne prawo jakości kodu AI. Znacznie mocniejszym sygnałem stają się badania bezpośrednio śledzące kod identyfikowany jako wygenerowany przez narzędzia AI. Preprint z marca 2026 roku przeanalizował 304 362 zweryfikowane commity AI w 6275 repozytoriach GitHub. Autorzy wykryli 484 606 problemów przypisanych do tych zmian, z czego 89,1% stanowiły code smells; w przypadku każdego z analizowanych asystentów ponad 15% commitów wprowadzało co najmniej jeden wykrywalny błąd, a 24,2% śledzonych problemów nadal występowało w najnowszej wersji repozytorium.

Badanie to jest bardzo świeże i w momencie publikacji ma status preprintu, więc nie powinno być traktowane jako ostateczny głos w dyskusji. Jego znaczenie polega jednak na czymś innym: przesuwa ono pytanie z laboratoryjnego „czy model potrafi wygenerować poprawny kod?" ku znacznie ważniejszemu „co dzieje się z tym kodem później?". Jeżeli część problemów trwa miesiącami i przechodzi przez kolejne wersje, oznacza to, że sztuczna inteligencja nie tylko generuje kod. Może ona generować zobowiązania konserwacyjne, które organizacja odziedziczy.

Jeszcze subtelniejszy problem dotyczy jakości niefunkcjonalnej. Program może przechodzić wszystkie testy funkcjonalne i nadal być nieakceptowalny: trudny w utrzymaniu, wolny, pamięciożerny, podatny na ataki lub nieczytelny. Badanie z 2025 roku, łączące przegląd 108 publikacji, warsztaty z praktykami oraz eksperymenty z łataniem rzeczywistych błędów programistycznych, wskazało właśnie na tę lukę. Autorzy zauważyli, że literatura akademicka koncentruje się głównie na bezpieczeństwie i wydajności, podczas gdy praktycy kładą nacisk na utrzymywalność i czytelność. Co więcej, poprawa jednego wymiaru jakości mogła odbywać się kosztem innego. Jest to lekcja o ogromnym znaczeniu dla przyszłości benchmarków. Passes tests nie znaczy is good engineering. Test odpowiada na pytanie, które potrafiliśmy zadać; nie odpowiada na pytania, których nie pomyśleliśmy zadać.

Jeżeli benchmark nagradza wyłącznie poprawność funkcjonalną, model będzie optymalizowany przede wszystkim pod tym kątem. Utrzymywalność, przejrzystość i spójność architektoniczna łatwo pozostają poza polem widzenia. Można więc sformułować nową wersję starego prawa Goodharta: kiedy przechodzenie testów staje się celem, przestaje być wystarczającą miarą jakości. System może nauczyć się produkować rozwiązania spełniające jawne kryteria, jednocześnie przerzucając koszt w obszary niemierzone. To samo zjawisko znamy z ekonomii publicznej: szpital może znacząco poprawić wskaźnik liczby przeprowadzonych procedur, nie poprawiając przy tym realnie zdrowia populacji.

Ryzyko wspólnego błędu poznawczego w automatyzacji testów

Szkoła może podnieść wyniki testów, nie zwiększając przy tym zdolności uczniów do samodzielnego myślenia. Podobnie zespół AI może poprawić wskaźnik pass rate, nie zwiększając realnej zdolności systemu do taniej zmiany w przyszłości. Problem pogłębia się, gdy AI generuje również same testy. Badanie obejmujące jedenaście modeli LLM i kilka zbiorów testowych wykazało, że błędny kod dostarczony modelowi jako kontekst znacząco obniża jakość generowanych testów oraz ich skuteczność w wykrywaniu błędów. W jednym z analizowanych zestawów testy tworzone na podstawie poprawnego kodu wykrywały znacznie więcej usterek niż te generowane przy dostępie do wadliwej implementacji.

Prowadzi to do krytycznego problemu wspólnego błędu poznawczego. Jeśli ten sam model lub ta sama trajektoria generuje specyfikację, kod, test i review, wszystkie te warstwy mogą dzielić to samo błędne założenie. Test nie staje się wtedy niezależnym sędzią kodu. Staje się jego adwokatem.

Z punktu widzenia epistemologii przypomina to badacza, który sam tworzy hipotezę, projektuje eksperyment sprzyjający jej potwierdzeniu, a na koniec samodzielnie recenzuje własną publikację. Dlatego w epoce AI podejście test-first powinno być rozumiane szerzej niż tylko jako mechaniczne pisanie testów przed implementacją. Chodzi o epistemiczną niezależność kryteriów sukcesu od wygenerowanego rozwiązania. Najpierw definiujemy zachowanie, przypadki brzegowe, własności bezpieczeństwa i warunki odrzucenia; dopiero potem agent próbuje zbudować implementację. W przeciwnym razie pojawia się pokusa dopasowania testu do tego, co model już stworzył. Materiał Cassaniego podaje konkretne wartości procentowe (odpowiednio 2,1%, 60% i 34%), które mają świadczyć o tym, że rygorystyczny code review, szkolenia walidacyjne i podejście test-first redukują określone negatywne efekty.

W dostępnych źródłach pierwotnych nie udało się jednak potwierdzić tych liczb w znaczeniu opisanym w notatce. Nie należy ich zatem powtarzać jako wyników naukowych. Teza jakościowa pozostaje natomiast mocna: DORA wskazuje na znaczenie solidnych mechanizmów testowych i małych batchy, a literatura dotycząca jakości kodu LLM konsekwentnie podkreśla potrzebę niezależnej walidacji. To właśnie metodologia prawdziwościowa jest tu kluczowa. W epoce AI niezwykle łatwo stworzyć przekonujący raport, który jedynie reprodukuje liczby z kolejnych streszczeń i prezentacji. W ten sposób jedna korelacja zamienia się w przyczynowość, analiza komercyjna w „badania naukowe”, a pojedynczy wskaźnik w uniwersalny dowód degradacji kodu.

Artykuł o odpowiedzialnej AI sam powinien być napisany odpowiedzialnie. Nie wolno bronić słusznej tezy niedokładnym dowodem. Właściwym wnioskiem z DORA 2024 nie jest zatem stwierdzenie, że „AI pogarsza kod". Oficjalny wynik jest bardziej złożony: większa adopcja wiązała się jednocześnie z poprawą pewnych lokalnych wskaźników jakości i pogorszeniem części parametrów delivery. Wnioskiem z GitClear nie jest to, że „AI ośmiokrotnie psuje repozytoria", lecz sygnał o wzroście mierników duplikacji i krótkotrwałego przepisywania kodu. Z kolei nowe badania commitów AI nie dowodzą, że „co czwarty fragment jest zły", lecz wskazują, że wykrywalne problemy jakościowe mogą przetrwać w repozytoriach długo po wprowadzeniu zmiany.

Taka ostrożność nie osłabia tezy o ryzyku długu technicznego – przeciwnie, czyni ją mocniejszą. Najważniejsze zagrożenie nie polega bowiem na tym, że AI zawsze pisze gorszy kod niż człowiek. Obraz jest znacznie bardziej złożony.

Szeroka analiza ponad pół miliona próbek Python i Java z 2025 roku wykazała m.in., że kod AI bywał prostszy i bardziej powtarzalny, częściej zawierał nieużywane konstrukcje oraz zahardkodowane elementy debugowania i charakteryzował się odmiennym profilem podatności niż kod ludzki. Z kolei kod pisany przez ludzi częściej obarczony był problemami związanymi ze złożonością i utrzymywalnością. Nie istnieje więc prosta oś „człowiek dobry – AI zła" lub odwrotnie.

Kontrola i governance jako warunek trwałej prędkości rozwoju

Różne profile błędów wymagają odmiennych mechanizmów kontroli, co wymusza redefinicję samego procesu code review. Reviewer przyszłości nie będzie sprawdzał jedynie tego, czy kod działa. Będzie musiał tropić charakterystyczny ślad generacji: zbędne abstrakcje, duplikacje, przesadną defensywność, zależności dodane dla trywialnych problemów czy kod syntaktycznie piękny, lecz niezgodny z idiomami organizacji. Przede wszystkim będzie szukał "martwej poprawności" – rozwiązania formalnie działającego, którego nikt rozsądny nie chciałby utrzymywać.

W tym ujęciu AI Code Reviewer nie może stać się kolejnym automatem przybijającym zieloną pieczątkę; jego zadaniem powinno być zwiększanie różnorodności krytyki. Jeśli jeden agent generuje rozwiązanie, drugi powinien otrzymać inny cel: znaleźć sposób na jego złamanie. Trzeci może analizować architekturę, a czwarty bezpieczeństwo. Ostateczne prawo integracji musi jednak pozostać związane z odpowiedzialnością człowieka, szczególnie tam, gdzie koszt błędu jest znaczny.

Równie istotna jest wielkość zmiany. DORA od lat podkreśla znaczenie małych batchy dla wysokiej wydajności systemu dostarczania, a raport 2024 wskazuje je jako jeden z elementów przeciwdziałających negatywnym efektom przyspieszonej produkcji kodu przez AI. To niemal prawo hydrodynamiki organizacji: mała zmiana jest łatwiejsza do zrozumienia, przetestowania, zweryfikowania, wycofania i przypisania do konkretnej przyczyny. AI kusi, by tworzyć zmiany większe, ponieważ potrafi produkować je bez wysiłku. Governance powinien reagować odwrotnie: im szybszy generator, tym mniejszy dopuszczalny pakiet zmiany.

To paradoks przypominający rozwój transportu. Im szybszy samochód, tym ważniejsze stają się hamulce. Nikt nie argumentuje, że pojazd o dwukrotnie większej mocy powinien mieć słabszy układ hamulcowy, bo „hamowanie ogranicza innowacyjność". Tymczasem podobny błąd organizacyjny zdarza się w software engineeringu: firma kupuje narzędzie zwiększające podaż zmian, a następnie traktuje review, testowanie i kontrolę architektoniczną jako przeszkody w realizacji obiecanej produktywności.

Dojrzałość polega na zrozumieniu, że kontrola nie jest przeciwieństwem prędkości, lecz warunkiem jej trwałości. Samochód może rozwijać większą prędkość właśnie dlatego, że ma skuteczne hamulce. Agent może otrzymać większą autonomię tylko wtedy, gdy system posiada niezawodne testy, rollback, observability i ograniczony blast radius. To jest właściwa ekonomia governance: inwestycja w kontrolę umożliwia wykorzystanie prędkości bez przenoszenia jej kosztu na przyszłe awarie.

Cassani proponuje w tym miejscu „kredyty długu technicznego": każdy sprint posiadałby ograniczony budżet dopuszczalnego zadłużenia, a po jego wyczerpaniu rozwój nowych funkcji ustępowałby refaktoryzacji. Nie jest to standard branżowy ani empirycznie zwalidowana metoda, ale interesujący instrument zarządczy, który czyni dług widzialnym przed momentem katastrofy. Organizacja nie mówi już abstrakcyjnie „musimy kiedyś posprzątać kod", lecz uzależnia możliwość dodawania funkcji od stanu jakości systemu.

Można tę ideę rozwinąć w ogólną zasadę technical debt budget. Każdy projekt ma skończoną zdolność absorpcji komplikacji. Jeśli zespół dodaje zależności, niestandardowe wyjątki, obejścia czy duplikaty, wykorzystuje część tego budżetu. AI nie zmienia jego skończoności – może jedynie sprawić, że zostanie zużyty szybciej.

Prawdziwym problemem nie jest samo powstawanie długu, lecz jego oprocentowanie. Niektóre niedoskonałości są niemal bezkosztowe; inne dotykają fragmentów zmienianych co tydzień i pobierają opłatę przy każdej kolejnej modyfikacji. Dlatego wartość refaktoryzacji powinna zależeć nie tylko od liczby code smells, ale od ich miejsca w strukturze zmian. Dług w martwym module jest

mniej groźny niż identyczny dług w centralnym komponencie, przez który przechodzi połowa rozwoju produktu. Tu agent może paradoksalnie stać się również narzędziem spłaty długu.

Od produktywności generowania do odpowiedzialności posiadania

Zdolność do analizy rozległych repozytoriów, identyfikacji powtarzalnych struktur, generowania testów regresyjnych i mechanicznej refaktoryzacji otwiera drogę do prac, które ludzkie zespoły przez lata odkładały jako zbyt żmudne. Materiał Cassaniego postuluje właśnie okresowe sesje refaktoryzacji wspieranej przez wyspecjalizowane prompty i agentów. Paradoks AI polega więc na tym, że ta sama technologia może dług techniczny zarówno produkować, jak i spłacać.

Rozstrzygające staje się pytanie o funkcję celu. Agent nagradzany za szybkie zamknięcie ticketu będzie zachowywał się inaczej niż ten, którego celem jest rozwiązanie problemu przy minimalnym wzroście złożoności. Ten drugi powinien otrzymać prawo do usuwania kodu, konsolidacji funkcji i proponowania rozwiązań bez konieczności implementacji. W istocie jednym z najcenniejszych promptów przyszłości może nie być polecenie „napisz rozwiązanie", lecz: „znajdź rozwiązanie wymagające najmniejszej ilości nowego systemu do utrzymywania".

Zasada ta ma charakter głęboko ekonomiczny. Każda nowa linia kodu produkcyjnego posiada wartość oczekiwaną, ale niesie ze sobą również koszt przyszłego utrzymania. Kod jest aktywem tylko wtedy, gdy wartość funkcji, którą umożliwia, przewyższa koszt jego istnienia. W przeciwnym razie jest zobowiązaniem ubranym w składnię.

Powstaje zatem potrzeba nowej księgowości technicznej. Bilans przedsiębiorstwa wykazuje serwery, licencje i kapitał, lecz nie pokazuje wprost, że system autoryzacji zależy od biblioteki, której nikt nie rozumie, trzy moduły implementują niemal tę samą logikę, a siedem mikroserwisów można by zastąpić jednym. Te niewidzialne pozycje mają jednak realną wartość ekonomiczną, a AI może sprawić, że zaczną one narastać szybciej niż kiedykolwiek.

Dlatego dojrzały „Mission Control Center" przyszłości nie powinien mierzyć wyłącznie liczby zadań wykonanych przez agentów. Powinien obrazować, ile kodu dodano, a ile usunięto, ile rozwiązań przetrwało, a ile wymagało poprawek, jak zmieniła się złożoność i gdzie wzrosła duplikacja – czyli jak szybko organizacja spłaca własny dług. Produktywność bez tych danych jest jak ocena kondycji firmy wyłącznie na podstawie przychodów, z pominięciem zobowiązań.

Właśnie dlatego bieżące badania DORA przesuwają akcent z samego faktu używania AI na zdolności całej organizacji. Raport 2025, oparty na opiniach niemal pięciu tysięcy profesjonalistów i ponad stu godzinach danych jakościowych, opisuje AI jako amplifier: wzmacniacz tego, co w systemie już istnieje. Zespół pracujący w małych batchach, z dobrymi testami, rzetelną dokumentacją i kulturą review może dzięki AI przyspieszyć. Zespół żyjący z ogromnych merge'ów, ukrytej wiedzy i ciągłego gaszenia pożarów może za pomocą tej samej technologii przyspieszyć produkowanie własnych problemów.

Nie jest to argument przeciwko AI, lecz przeciwko naiwnej automatyzacji. Rewolucja agentowa może stworzyć oprogramowanie bardziej dostępne, szybciej powstające i łatwiejsze do modernizacji, ale może również zalać świat milionami lokalnie poprawnych, a globalnie zbędnych komponentów. Technologia nie wybiera scenariusza samodzielnie; wybór ten leży w architekturze bodźców, governance, miernikach i kulturze inżynieryjnej.

Najważniejsza maksyma mogłaby zatem brzmieć: nie pytaj, ile kodu potrafi wyprodukować twoja AI; zapytaj, ile kodu twoja organizacja potrafi odpowiedzialnie posiadać. Pierwsze pytanie dotyczy generacji, drugie – dojrzałości. Myśl ta prowadzi ku jeszcze szerszemu problemowi.

Taniej generowany kod przesuwa punkt ciężkości z implementacji na strategiczną wiedzę domenową

Jeżeli kod staje się łatwo generowalny, a agenci potrafią jednocześnie analizować wymagania, projektować, implementować, testować i dokumentować, zmienia się nie tylko proces wytwarzania. Zmienia się architektura samego przedsiębiorstwa technologicznego: decyzja make-or-buy, znaczenie własnego software'u, relacja między biznesem a IT, liczebność zespołów, rola platform oraz centrum kontroli agentów. Pojawia się pytanie, czy przewaga konkurencyjna pozostanie w kodzie, czy przeniesie się ku wiedzy domenowej, danym i zdolności organizacji do podejmowania trafniejszych decyzji szybciej niż konkurenci. Gdy kod tanieje, rośnie cena wiedzy: make-or-buy, przedsiębiorstwo agentowe i nowa architektura przewagi.

Choć generowanie kodu rzeczywiście staje się coraz tańsze, konsekwencje tego zjawiska wykraczają poza samą produktywność zespołu programistycznego. Zmienia się kluczowy dylemat: co warto kupować na rynku, a co opłaca się budować samodzielnie?

Przez kilka dekad odpowiedź była względnie stabilna. Jeżeli funkcja nie stanowiła źródła przewagi konkurencyjnej, przedsiębiorstwo kupowało gotowy system. Własne oprogramowanie rezerwowano dla problemów wyjątkowych, strategicznych lub takich, których nie dało się dobrze

obsłużyć produktem rynkowym. Wysoki koszt zespołu deweloperskiego stanowił ekonomiczną zaporę przed niekontrolowanym "build". Agentowa AI zaczyna tę zaporę obniżać.

Analiza Cassaniego czyni z przewartościowania decyzji make-or-buy jedno z pięciu zasadniczych zjawisk organizacyjnych transformacji. W jego ujęciu agenci obniżają koszt tworzenia dedykowanego oprogramowania, a centrum ciężkości przesuwa się od samej implementacji ku architekturze, strategii i wiedzy domenowej. Autor wskazuje ponadto na urzeczywistnienie zarządzania wiedzą przez Context Engineering, zbliżenie użytkownika i twórcy poprzez język naturalny, narodziny nowych ról technicznych oraz potrzebę znacznie silniejszego governance. Te pięć zjawisk należy rozpatrywać łącznie.

Obniżenie ceny kodowania nie oznacza bowiem automatycznie, że każde przedsiębiorstwo powinno stać się producentem własnego software'u. Oznacza jedynie zmianę struktury kosztów tej decyzji. W klasycznej teorii firmy Ronalda Coase'a przedsiębiorstwo internalizuje działanie wtedy, gdy koszt organizacji wewnętrznej jest korzystniejszy od kosztu zawarcia i egzekwowania transakcji rynkowej. Oliver Williamson rozwinął tę intuicję w ekonomię kosztów transakcyjnych, wskazując na znaczenie niepewności, specyficzności aktywów i ryzyka oportunizmu. Agentowa AI wchodzi dokładnie w środek tej konstrukcji: obniża część kosztu wykonania wewnętrznego, lecz jednocześnie generuje nowe wydatki związane z modelami, infrastrukturą, bezpieczeństwem, ewaluacją i zależnością od dostawców inteligencji.

"Make" przestaje więc oznaczać dawną pełną samodzielność. Firma może posiadać własny kod, korzystając jednocześnie z cudzych modeli, chmury, frameworków agentowych i narzędzi. W pracy koncepcyjnej z 2026 roku David Klotz analizuje tę zmianę, argumentując, że agentowa AI modyfikuje klasyczne determinanty decyzji build-or-buy: koszt, strategiczne zróżnicowanie, specyficzność rozwiązania, vendor lock-in, time-to-market, jakość, compliance i zdolności organizacyjne. Co szczególnie ważne, autor odrzuca skrajną tezę o rychłej "SaaSocalypse"; według jego analizy własne budowanie staje się atrakcyjne w przypadku aplikacji prostych lub strategicznie różnicujących przedsiębiorstwo, podczas gdy systemy wysoko regulowane i krytyczne nadal przemawiają za zakupem lub rozwiązaniami hybrydowymi.

Jest to praca koncepcyjna, a nie wynik empiryczny na wielką skalę, dlatego należy traktować ją jako rozwiniętą hipotezę ekonomiczną, a nie ustalone prawo rynku. Ta ostrożność jest niezbędna, gdyż łatwo wyprowadzić z postępu AI fałszywy sylogizm: skoro agent potrafi zbudować aplikację taniej, to opłaca się budować wszystko samodzielnie. Nie. Koszt napisania systemu nigdy nie był jedynym kosztem jego posiadania. Pozostają wymagania bezpieczeństwa, aktualizacje, zgodność regulacyjna, integracje, ciągłość działania, odpowiedzialność za incydenty, obserwowalność, migracje danych oraz wiedza potrzebna do utrzymania rozwiązania po odejściu jego twórców.

Demokratyzacja produkcji nie zastępuje profesjonalizmu inżynieryjnego

AI obniża koszt wejścia do systemu, ale nie znosi kosztu odpowiedzialności za ten system. Można wręcz oczekiwać nowej fali zjawiska, które nazwałbym shadow software. Przez lata działy IT walczyły z shadow IT: arkuszami, skryptami, aplikacjami i usługami chmurowymi kupowanymi lub tworzonymi poza architekturą organizacyjną. Vibe Coding może ten problem rozszerzyć.

Kierownik sprzedaży, analityk czy pracownik operacyjny nie będzie już musiał kupować zewnętrznego narzędzia – będzie mógł poprosić AI o zbudowanie własnego. To radykalna demokratyzacja sprawczości cyfrowej, a jednocześnie ryzyko powstania setek aplikacji, baz i agentów, o których centralne IT dowie się dopiero wtedy, gdy przestaną działać. Język naturalny zmniejsza odległość między użytkownikiem a twórcą, co źródło Cassaniego słusznie wskazuje jako jedno z kluczowych zjawisk adopcji. Nie należy jednak mylić usunięcia bariery implementacji z usunięciem bariery profesjonalizmu.

Człowiek, który potrafi opisać formularz fakturowania, może z pomocą modelu go stworzyć. Nie oznacza to jednak, że rozumie zasady przechowywania danych, autoryzację, backup, podatności, audyt, retencję, przepisy o danych osobowych czy konsekwencje zmiany schematu bazy. AI nie tyle demokratyzuje inżynierię oprogramowania, ile demokratyzuje inicjację produkcji oprogramowania. To kluczowe rozróżnienie. Prawie każdy może dziś uruchomić samochód, ale nie każdy potrafi go zaprojektować. Aparat w telefonie pozwolił miliardom ludzi robić zdjęcia, lecz nie uczynił wszystkich fotografami. Arkusz kalkulacyjny umożliwił samodzielne modelowanie finansowe, ale nie zniósł zawodu aktuariusza.

Narzędzia obniżają próg działania, lecz nie eliminują potrzeby kompetencji tam, gdzie rosną konsekwencje błędów. Najciekawszy wariant dylematu make-or-buy może więc przyjąć formę make-and-buy. Przedsiębiorstwo kupuje platformy, modele, systemy tożsamości, chmurę i komponenty commodity, lecz we własnym zakresie buduje cienką warstwę software'u, która materializuje unikalne procesy, dane i wiedzę domenową. Nie musi już tworzyć wszystkiego od podstaw – może składać. Agent staje się rodzajem niezwykle szybkiego warsztatu integracyjnego.

To z kolei może osłabiać przewagę dostawców SaaS wynikającą jedynie z posiadania konkretnych funkcji. Jeśli zestaw „dashboard + formularz + workflow + integracja” można coraz taniej wygenerować i dostosować, samo posiadanie tych elementów przestaje być głębokim fosą konkurencyjną. Wartość przesuwa się w stronę elementów trudniejszych do skopiowania: niezawodności, regulacyjnego know-how, ekosystemu integracji, danych, dystrybucji, reputacji,

bezpieczeństwa i wiedzy procesowej. Analogicznie dzieje się wewnątrz organizacji. Kod, który można względnie szybko odtworzyć, przestaje być jedynym repozytorium przewagi. Najcenniejsze staje się to, czego model nie może odgadnąć bez dostępu do historii firmy: dlaczego nie obsługuje się określonego typu klienta lub dlaczego trzy lata wcześniej odrzucono technologię, która dziś ponownie wygląda atrakcyjnie?

Wiedza domenowa jako paliwo operacyjne i ryzyko systemowe

Który wyjątek w systemie wynika ze specyficznego obowiązku prawnego? Dlaczego proces magazynowy zawiera pozornie nieracjonalny krok? Który klient ma nietypowy kontrakt i gdzie oficjalna procedura rozmija się z faktyczną praktyką? To właśnie tam kryje się przewaga domenowa. Firma z dwudziestoletnim stażem gromadzi tysiące takich mikrodecyzji, jednak często pozostają one rozproszone w głowach pracowników, mailach, dokumentach, ticketach, starych commitach czy nieaktualnych wiki.

Dlatego Cassani wiąże transformację z Context Engineering i zarządzaniem wiedzą: dokumentacja, ontologie biznesowe i historia systemu mają stać się dynamicznym kontekstem dostępnym dla agentów. W ten sposób knowledge management, który przez dekady bywał korporacyjnym cmentarzem dobrych intencji, otrzymuje potężny bodziec ekonomiczny. Dotychczas przedsiębiorstwo mogło tworzyć doskonałą bazę wiedzy, której pracownicy po prostu nie czytali. Agent natomiast wykorzystuje tę wiedzę bezpośrednio w działaniu. Dokumentacja przestaje być tekstem pisanym "na wszelki wypadek", a staje się paliwem operacyjnym dla systemów wykonujących rzeczywiste zadania. To fundamentalnie zmienia ekonomię jakości informacji.

Błędna dokumentacja nie jest już tylko irytującą wskazówką dla nowego pracownika; może automatycznie stać się źródłem błędnych działań wielu agentów. Nieaktualna procedura może być stosowana maszynowo, z idealną konsekwencją. Paradoks organizacji przyszłości brzmi zatem: im lepiej AI wykorzystuje wiedzę firmy, tym droższa staje się ta nieprawdziwa. Pojawia się potrzeba epistemicznego governance danych organizacyjnych. Należy wiedzieć nie tylko, co organizacja "wie", lecz kto zatwierdził tę wiedzę, kiedy była aktualizowana, w jakim zakresie obowiązuje i które źródło ma pierwszeństwo w razie konfliktu. Kontekst agentów musi posiadać provenance. Nie wystarczy wyszukać pięciu dokumentów zawierających podobne słowo; trzeba rozróżnić aktualną politykę od notatki roboczej, projektu sprzed czterech lat czy dokumentu klienta, którego reguły nie dotyczą całej firmy. Architektura przedsiębiorstwa zaczyna przypominać konstytucję informacyjną.

Nie wszystkie źródła mają taką samą rangę. Zarządzenie prezesa, umowa, regulamin, kod produkcyjny, ticket i wpis na Slacku należą do różnych porządków wiarygodności. Agentowi nie wystarczy przekazać treści – trzeba mu przekazać hierarchię źródeł. Tu właśnie rośnie znaczenie architekta przedsiębiorstwa. Analizy literatury z 2025 roku dotyczące generatywnej AI w pracy enterprise architects wskazują na trzy główne obszary zastosowań: wspieranie ideacji i eksploracji kompromisów, szybkie tworzenie artefaktów oraz usprawnienie wyszukiwania wiedzy. Równocześnie podkreśla się ryzyka: nieprzejrzystość, błędy kontekstowe prowadzące do reworku, problemy z compliance oraz zjawiska osłabiające aktywny udział człowieka. Nie prowadzi to jednak do zaniku enterprise architecture, lecz do przesunięcia roli architekta z autora diagramów na kuratora ograniczeń i strażnika spójności. Jeśli implementacja tanieje, decyzje architektoniczne relatywnie drożeją. Wybór granic domen, odpowiedzialności za dane czy zasad bezpieczeństwa może oddziaływać na tysiące automatycznie generowanych później zmian.

W klasycznej firmie programista pytał architekta przed budową dużego komponentu, bo sam proces tworzenia był kosztowny. W świecie agentów komponent może powstać, zanim architekt skończy spotkanie. Governance musi zatem zostać przesunięty przed etap generacji. Zasady architektoniczne należy implementować w kontekście, policy engine, szablonach i automatycznych testach. Nie można liczyć na to, że człowiek będzie ręcznie korygował każdą błędną decyzję po fakcie.

Przewaga organizacji leży w zarządzaniu sprawczością, nie w modelu

To właśnie tutaj metafora "Mission Control Center" z materiału Cassaniego nabiera realnej wartości. Autor przewiduje ewolucję klasycznych IDE w stronę środowisk agent-first, gdzie centrum pracy nie stanowi już pojedynczy plik otwarty przez programistę, lecz cele, działania agentów, kontekst, walidacja, historia i kontrola wykonania. Choć nie wszystkie wymienione funkcje należy traktować jako stabilny standard rynkowy, sama zmiana kierunku interfejsu jest logiczna. Gdy jeden człowiek nadzoruje kilka procesów agentowych, folder po lewej stronie ekranu przestaje być wystarczającą mapą pracy. Mission Control powinien odpowiadać na pytania, których tradycyjne IDE nie musiało zadawać: które agenty właśnie działają, jaki cel realizują, z jakich źródeł korzystają i jakie mają uprawnienia? Ile wydały tokenów, gdzie zatrzymały się na checkpointach, a które wyniki czekają na zatwierdzenie? Gdzie nastąpił dryf względem planu, jaki jest poziom ryzyka i co stanie się po naciśnięciu przycisku "stop"? To nie jest już wyłącznie środowisko edycji kodu. Jest to pulpit zarządzania delegowaną sprawczością.

Podobny kierunek wyznacza raport DORA 2025. Oparty na opiniach niemal pięciu tysięcy profesjonalistów oraz ponad stu godzinach danych jakościowych, opisuje on AI jako wzmacniacz istniejącego systemu, przesuwając dyskusję z poziomu pojedynczego narzędzia ku zdolnościom organizacyjnym. DORA AI Capabilities Model idzie jeszcze dalej: sukces w wykorzystaniu AI wiąże nie z samym dostępem do modeli, lecz z praktykami technicznymi i kulturowymi, które pozwalają wdrożyć je na poziomie całej organizacji. Wniosek ten ma znaczenie strategiczne. Jeśli każdy konkurent może kupić dostęp do podobnego modelu, sam model przestaje być trwałą przewagą konkurencyjną – staje się jedynie zasobem rynkowym. Przewaga musi zatem wynikać ze sposobu jego użycia: z danych, integracji, kontekstu, organizacji, ewaluacji, procesu uczenia się i prędkości konwersji wiedzy na decyzję.

Historia gospodarcza zna ten mechanizm. Dostęp do elektryczności przestał być atutem, gdy stała się ona powszechna; przewagę zyskały przedsiębiorstwa, które lepiej zreorganizowały produkcję wokół niej. Podobnie dostęp do Internetu przestał być wyjątkowy, lecz zdolność zbudowania na nim Amazona nie stała się powszechna. Arkusz kalkulacyjny jest dostępny dla każdej firmy, ale nie każda równie sprawnie alokuje kapitał. Technologia powszechna może zatem wzmacniać różnice organizacyjne, zamiast je wyrównywać.

W tym kontekście można zaproponować nowe równanie przewagi przedsiębiorstwa AI: nie "najlepszy model wygrywa", lecz najlepsza pętla uczenia wygrywa. Organizacja obserwuje rzeczywistość, zbiera dane, interpretuje je, formułuje decyzję, materializuje ją w procesie lub oprogramowaniu, mierzy rezultat i aktualizuje wiedzę.

Redukcja latencji organizacyjnej poprzez federację sprawczości

AI może skrócić każdą z tych faz. Jeśli konkurent potrzebuje sześciu miesięcy, by przejść od informacji rynkowej do zmiany systemu, a przedsiębiorstwo agentowe robi to w sześć tygodni, przewaga nie tkwi w liczbie napisanych linii kodu. Tkwi w latencji organizacyjnej. To pojęcie może stać się jednym z centralnych mierników przedsiębiorstwa przyszłości: ile czasu mija od odkrycia faktu do zmiany zachowania organizacji?

Państwa, korporacje i instytucje często nie przegrywają z powodu braku informacji, lecz dlatego, że informacja zbyt długo wędruje przez strukturę. Raport powstaje miesiąc, decyzja trwa kolejny, wymagania trafiają do IT, a potem następują przetargi, backlog, development i wdrożenie. W chwili ukończenia systemu rzeczywistość zdążyła się już zmienić. Agentowa AI może radykalnie skrócić tę

drogę, ale tylko pod warunkiem, że organizacja potrafi podejmować decyzje. Jeśli zarządzanie pozostaje sparaliżowane, AI będzie jedynie błyskawicznie produkować kolejne analizy oczekujące na zatwierdzenie. Ponownie działa zasada DORA: technologia wzmacnia system. Szybki agent osadzony w powolnej biurokracji nie tworzy szybkiej organizacji. Tworzy szybszą kolejkę przed biurokracją.

Prowadzi to do przebudowy relacji na linii biznes-IT. Przez dekady dominował model zgłoszeniowy: biznes określa potrzeby, IT je realizuje. Później Agile próbował zbliżyć oba światy. Vibe Coding i narzędzia agentowe przesuwają tę granicę jeszcze dalej, umożliwiając ekspertowi domenowemu bezpośredni udział w materializacji rozwiązania. Nie oznacza to jednak zaniku IT. Oznacza raczej, że IT przestanie być fabryką wszystkich funkcji, a stanie się twórcą bezpiecznej platformy – zbioru zasad, kontekstów i guardrails, w obrębie których jednostki biznesowe budują część rozwiązań samodzielnie.

Jest to analogiczne do transformacji infrastruktury chmurowej. Centralny dział IT dawniej przydzielał serwer; później budował platformę, na której zespoły mogły samodzielnie i w kontrolowany sposób uruchamiać zasoby. W epoce agentowej może zrobić to samo ze sprawczością programistyczną: zamiast ręcznie tworzyć każdą aplikację, dostarczać paved road dla bezpiecznego tworzenia aplikacji i agentów. Platform engineering staje się więc naturalnym sojusznikiem agent-centric SDLC. DORA już w raporcie 2024 wskazywała platform engineering jako jeden z kluczowych obszarów badawczych obok AI, podkreślając jednocześnie znaczenie orientacji na użytkownika oraz stabilnych priorytetów organizacyjnych. W świecie AI platforma powinna obejmować nie tylko CI/CD, obserwowalność i infrastrukturę, lecz również zatwierdzone modele, polityki danych, narzędzia agentów, warstwy autoryzacji, evals i biblioteki kontekstów.

Dzięki temu przedsiębiorstwo może demokratyzować tworzenie bez demokratyzowania dostępu do katastrofy. Marketing może zbudować wewnętrzny workflow, nie otrzymując jednocześnie prawa do danych płacowych. Analityk może stworzyć narzędzie raportowe, ale jego wdrożenie na produkcję przejdzie przez automatyczną bramę bezpieczeństwa. Agent może zaproponować migrację schematu, lecz nie posiadać klucza umożliwiającego jej wykonanie bez zatwierdzenia. Decentralizujemy inicjatywę, centralizujemy konstytucję.

To być może jeden z najważniejszych modeli organizacyjnych przyszłości. Centralizacja każdej implementacji stanie się zbyt wolna, a pełna decentralizacja – zbyt ryzykowna. Rozwiązaniem jest federacja: wspólne standardy, bezpieczeństwo i warstwa wiedzy przy zachowaniu lokalnej swobody eksperymentowania. Firma zaczyna przypominać państwo konstytucyjne, gdzie centrum określa prawa, granice i infrastrukturę, ale nie podejmuje każdej lokalnej decyzji.

Koncepcje Cassaniego prowadzą w podobnym kierunku, choć posługują się językiem technicznym. RACM mapuje kompetencje i ryzyko, PAIP dobiera reżim współpracy do zadania, ECF organizuje dialog człowiek-AI, RAUC buduje pamięć odpowiedzialnego użycia, a Mission Control spina działanie agentów w warstwę nadzorczą. Te elementy nie muszą być wdrażane dokładnie pod tymi nazwami, aby ich wspólna logika była wartościowa. Przedsiębiorstwo potrzebuje operacyjnej konstytucji dla inteligencji, która stała się tania i powszechnie dostępna.

W takim układzie dylemat make-or-buy przestaje być jednorazową decyzją zakupową, a staje się zarządzaniem portfelem warstw. Kup model, jeśli jest commodity; buduj wiedzę domenową, jeśli jest unikalna. Kup system płatniczy, gdy compliance i niezawodność zewnętrznego dostawcy dają przewagę; buduj interfejs lub workflow, jeśli odzwierciedla on niepowtarzalny proces firmy. Utrzymuj własne evals, aby móc swobodnie zmieniać modele. Nie zamykaj krytycznej pamięci organizacji u jednego dostawcy i nie buduj samodzielnie tego, czego utrzymania nie jesteś gotów finansować. Innymi słowy: AI nie znosi braku strategii technologicznej – ona drastycznie podnosi cenę tego braku. Najbardziej fascynująca konsekwencja pojawia się jednak jeszcze poziom wyżej.

Kod jako plastyczna warstwa organizacji

Jeśli koszt implementacji spada, może dojść do przesunięcia w samej naturze przedsiębiorstwa – z organizacji o sztywnych procesach ku takiej, która potrafi je nieustannie przepisywać. Dotychczas software często zamrażał procedury na lata, gdyż zmiana systemu była zbyt kosztowna. Agentowa AI daje szansę, by system informatyczny częściej podążał za ewolucją modelu biznesowego. Może to zredukować jeden z najbardziej niedocenianych kosztów korporacyjnych: sytuację, w której firma realizuje proces nie dlatego, że jest on optymalny, lecz dlatego, że tak działa system. Frazy typu „SAP nie pozwala", „aplikacja tego nie obsługuje" czy „zmiana wymaga projektu na dwa kwartały" nie są jedynie opisem technologii, lecz granicą sprawczości organizacyjnej. Gdy AI obniży koszt dostosowania systemów, organizacja odzyska możliwość pytania o to, jak proces powinien działać, zamiast zastanawiać się wyłącznie nad tym, na co pozwala obecny software.

Nie oznacza to, że każda procedura powinna zmieniać się codziennie – stabilność pozostaje wartością. Oznacza jednak, że oprogramowanie może stracić część funkcji betonującej instytucję. Kod przestaje być wyłącznie archiwum dawnych decyzji, a staje się plastyczną warstwą organizacji. Tu pojawia się jednak istotne ryzyko: skoro software można modyfikować szybciej, można również sprawniej wdrażać błędne decyzje biznesowe. Techniczna zdolność implementacji nie jest tożsama z mądrością strategiczną. Agent potrafiący w godzinę przebudować politykę rabatową nie odpowie

na pytanie, czy w ogóle należy to robić. Ponownie zatem wracamy do prymatu intencji. Im tańsza staje się wykonawczość, tym droższy relatywnie staje się osąd.

Przyszła przewaga konkurencyjna może więc rzadziej opierać się na posiadaniu unikalnego kodu, a częściej na zdolności do szybszego i lepszego rozumienia rzeczywistości oraz sprawnego przekładania wiedzy na decyzje i ich bezpieczną materializację w systemie. Kod pozostanie konieczny, ale przestanie być najrzadszym elementem układanki.

Rzadkością będą natomiast: głęboka wiedza domenowa, wysokiej jakości dane, relacje z klientami, zaufanie, reputacja oraz kompetencja architektoniczna i zdolność rozpoznawania skutków drugiego rzędu. Kluczowa będzie także kultura pozwalająca człowiekowi powiedzieć „nie" rozwiązaniu, które wszystkie agenty uznały za genialne. Tak rozumiane przedsiębiorstwo agentowe to nie firma, która „ma dużo AI", lecz taka, która zbudowała krótki i kontrolowany obieg między wiedzą a działaniem. Widzi wcześniej, rozumie głębiej, decyduje sprawniej i wykonuje szybciej, mierząc rezultaty i ucząc się bez utraty odpowiedzialności.

Odpowiedzialność ludzka jako fundament nadzoru nad AI

Dopiero wtedy technologia staje się przewagą systemową. Czas dotknąć granicy, której nie wolno pozostawić technokratom. Skoro agentowe systemy zaczynają uczestniczyć w decyzjach organizacyjnych, pojawia się pytanie o prawo, odpowiedzialność, etykę i własność tych decyzji. Kto odpowiada, gdy agent błędnie klasyfikuje człowieka, odrzuca zgłoszenie, narusza prywatność, wykorzystuje materiał chroniony, dyskryminuje lub realizuje cel zgodnie z instrukcją, lecz sprzecznie z dobrem organizacji? W tym punkcie PAIP, RAUC i governance spotykają się już nie tylko z software engineeringiem, lecz z europejskim prawem, RODO, AI Act, NIS2 i najbardziej klasycznym pytaniem filozofii instytucji: kto odpowiada za władzę, której część wykonania powierzono maszynie? Prawo do decyzji i obowiązek odpowiedzialności: AI Act, RODO, NIS2 oraz człowiek, którego nie wolno zamienić w alibi.

Niebezpieczne pytanie o odpowiedzialność za system agentowy brzmi pozornie niewinnie: „kto podjął decyzję?". W klasycznej organizacji można było odpowiedzieć nazwiskiem pracownika, nazwą organu, funkcją kierownika lub wskazaniem konkretnej procedury.

W świecie agentowym pojawia się nowa pokusa językowa: „system zdecydował", „algorytm odrzucił", „AI wybrała", „agent wdrożył". Sformułowania te bywają technicznie użyteczne, lecz normatywnie mogą działać jak zasłona dymna. Maszyna posiada zdolność wykonania, ale

współczesny europejski porządek regulacyjny nie buduje odpowiedzialności poprzez nadanie agentowi osobowości prawnej. Obowiązki lokuje po stronie ludzi i organizacji występujących w rolach dostawcy, podmiotu wdrażającego, administratora danych, podmiotu przetwarzającego czy organu zarządzającego. AI może być wykonawcą działania; nie powinna jednak stać się miejscem, w którym znika odpowiedzialność za to działanie. To rozróżnienie współgra z materiałem Cassaniego, który podkreśla potrzebę kontroli uprawnień agentów, audytowalności i przypisywania odpowiedzialności w ramach governance. Należy jednak doprecyzować język źródła. Cassani momentami ujmuje AI Act, RODO i NIS2 jako jednolity pakiet wymagań odnoszący się bezpośrednio do każdego narzędzia AI używanego w przedsiębiorstwie. Prawo europejskie działa bardziej subtelnie.

Zakres obowiązków zależy od roli podmiotu, celu wykorzystania systemu, rodzaju danych, sektora, skali ryzyka i konkretnego rodzaju decyzji. Nie każdy asystent kodu jest systemem wysokiego ryzyka. Nie każde użycie AI wymaga DPIA. Nie każda organizacja podlega NIS2. Compliance nie polega na przyklejeniu trzech skrótów do jednego dokumentu polityki, lecz na prawidłowej kwalifikacji zdarzenia.

Stan prawny na 18 sierpnia 2026 roku wymaga szczególnej uwagi, ponieważ europejski AI Act przeszedł istotną korektę harmonogramu. Rozporządzenie obowiązuje od 2024 roku; część przepisów dotyczących zakazanych praktyk, definicji i AI literacy zaczęła być stosowana w lutym 2025 roku, a reguły dotyczące zarządzania i modeli ogólnego przeznaczenia od sierpnia 2025 roku. Od 2 sierpnia 2026 roku obowiązują również wymogi przejrzystości przewidziane w art. 50. Jednocześnie AI Omnibus, który wszedł w życie 27 lipca 2026 roku, przesunął stosowanie głównych przepisów dotyczących systemów wysokiego ryzyka z załącznika III na 2 grudnia 2027 roku, a dla AI wbudowanej w określone regulowane produkty na 2 sierpnia 2028 roku. Jest to ważna korekta wobec wcześniejszych harmonogramów i kolejny dowód, że prawo AI trzeba traktować jako żywy system regulacyjny, a nie slajd raz przygotowany na szkolenie compliance.

AI Act opiera kwalifikację przede wszystkim na zamierzonym zastosowaniu. Komisja wskazuje na przykład systemy stosowane do oceny kandydatów do pracy, dostępu do świadczeń, oceny zdolności kredytowej czy niektórych zastosowań biomedycznych, edukacyjnych, policyjnych i migracyjnych jako obszary wysokiego ryzyka. Z kolei większość systemów AI w zastosowaniach minimalnego ryzyka nie podlega automatycznie dodatkowemu reżimowi tylko dlatego, że wykorzystuje sztuczną inteligencję.

Tym samym ten sam model językowy może być nieproblematycznym pomocnikiem przy redakcji wewnętrznej dokumentacji i jednocześnie elementem znacznie bardziej regulowanego systemu, gdy zostanie włączony w proces oceniania kandydatów do pracy. Ryzyko prawne tkwi więc nie tylko w

modelu, lecz w funkcji, którą mu powierzamy. Prowadzi to do wniosku kluczowego dla projektowania agentów: organizacja nie powinna klasyfikować „narzędzia” raz na zawsze, lecz konkretny use case. Agent posiadający dostęp do kodu i generujący dokumentację działa w innym kontekście niż ten sam agent analizujący CV, dane medyczne albo zdolność kredytową klienta. RACM Cassaniego, mapujący możliwości, stabilność i autonomię narzędzi, można zatem rozbudować o warstwę regulacyjną: nie tylko „co model potrafi?”, ale również „w jakiej roli prawnej znajduje się organizacja, gdy pozwala mu to robić?”.

Nadzór ludzki i dane jako fundamenty prawnej sprawczości

Kluczową kwestią jest tutaj human oversight. AI Act w odniesieniu do systemów wysokiego ryzyka wymaga, aby nadzór człowieka był rzeczywisty: system musi być zaprojektowany tak, by osoba fizyczna mogła go skutecznie kontrolować, rozumieć jego możliwości i ograniczenia, dostrzegać anomalie oraz – w stosownym zakresie – zdecydować o odrzuceniu rezultatu lub zatrzymaniu systemu. Po wejściu w życie odpowiednich przepisów podmiot wdrażający będzie musiał wyznaczyć osoby odpowiednio wyposażone i uprawnione do sprawowania takiego nadzoru.

Prawo dochodzi więc do wniosku zbieżnego z naszą analizą techniczną: człowiek w pętli nie może być człowiekiem narysowanym na diagramie. Rozróżnienie to ma bezpośrednie znaczenie już dziś, na gruncie RODO. Artykuł 22 daje osobie prawo – z pewnymi wyjątkami – do tego, by nie podlegać decyzji opartej wyłącznie na zautomatyzowanym przetwarzaniu, jeśli wywołuje ona skutki prawne lub w podobny sposób istotnie na nią wpływa. Tam, gdzie dopuszczalne są wyjątki, RODO wymaga zabezpieczeń, w tym co najmniej możliwości interwencji ludzkiej, przedstawienia własnego stanowiska i zakwestionowania decyzji. Człowiek pojawia się w prawie nie jako ceremonialny sygnatariusz wyniku, lecz jako kanał odzyskania sprawczości przez osobę poddaną automatycznej ocenie.

W tym zapisie kryje się głęboka intuicja konstytucyjna: decyzja dotycząca człowieka nie powinna stawać się niepodważalna tylko dlatego, że wygenerował ją mechanizm statystyczny. Max Weber opisywał biurokrację jako triumf racjonalności proceduralnej, jednak już biurokracja papierowa potrafiła stworzyć sytuację, w której człowiek stawał przed anonimowym aparatem i słyszał: „tak wyszło z procedury”. AI może zwielokrotnić tę pokusę. „Tak wyszło z modelu” jest cyfrowym kuzynem "tak wyszło z tabeli".

Państwo prawa i odpowiedzialne przedsiębiorstwo muszą zachować możliwość stwierdzenia, że wynik jest jedynie wynikiem, a decyzja pozostaje decyzją podlegającą uzasadnieniu i kontroli. RODO nakazuje również ostrożniejsze traktowanie danych dostarczanych systemom AI, niż sugeruje to praktyka „wrzuć wszystko do kontekstu". Zasady z art. 5 obejmują ograniczenie celu, minimalizację danych, prawidłowość, ograniczenie przechowywania, integralność, poufność i rozliczalność. Dane mają być adekwatne i ograniczone do tego, co niezbędne, a administrator musi być w stanie wykazać przestrzeganie tych zasad. Jest to prawny odpowiednik zasady context engineering: najlepszy kontekst nie jest największym kontekstem.

W praktyce developer korzystający z zewnętrznego modelu nie powinien traktować repozytoriów, logów czy baz produkcyjnych jako jednego zasobu do dowolnego kopiowania do promptu. Najpierw musi paść pytanie o realną potrzebę użycia danych osobowych, podstawę ich przetwarzania, miejsce ich transferu, czas przechowywania oraz status dostawcy i zgodność z celem. EROD w opinii dotyczącej modeli AI podkreśla indywidualne podejście do legalności wykorzystywania danych; wskazuje, że nawet powołanie się na uzasadniony interes wymaga badania konieczności i wyważenia praw osób, a znaczenie ma również pochodzenie danych i oczekiwania osób, których one dotyczą.

Jeszcze silniejsze ograniczenia dotyczą szczególnych kategorii danych. Artykuł 9 RODO obejmuje m.in. dane ujawniające pochodzenie rasowe lub etniczne, poglądy polityczne, przekonania religijne, przynależność związkową, dane biometryczne, zdrowotne oraz dotyczące orientacji seksualnej. Ich przetwarzanie jest co do zasady zakazane, chyba że zachodzi jedna z ustawowych przesłanek. To kolejna korekta wobec myślenia, że skoro „dane są w firmie", to model może ich użyć. Posiadanie danych nie jest równoznaczne z prawem do każdego ich wykorzystania.

Podobnego doprecyzowania wymaga jedna z tez źródłowych Cassaniego, sugerująca obowiązkowe wykonywanie DPIA dla nowych procesów AI. RODO nie ustanawia tak generalnej reguły. Ocena skutków dla ochrony danych jest wymagana wtedy, gdy przetwarzanie – szczególnie przy użyciu nowych technologii – ze względu na charakter, zakres i cele może z dużym prawdopodobieństwem powodować wysokie ryzyko dla praw lub wolności osób. RODO wymienia tu m.in. systematyczną automatyczną ocenę osób prowadzącą do znaczących skutków oraz przetwarzanie na dużą skalę danych szczególnych. Wniosek brzmi więc: nie „AI = DPIA", lecz „AI może uruchamiać przesłanki DPIA, które trzeba świadomie zbadać".

Może się to wydawać prawniczym detalem, lecz właśnie na takich szczegółach buduje się dojrzałość. Nadmierne compliance szybko zamienia się w biurokrację i prowokuje do obchodzenia reguł; zbyt wąskie staje się jedynie dekoracją. Dobra regulacja wewnętrzna powinna być proporcjonalna. Jeśli pracownik używa zatwierdzonego asystenta do poprawy dokumentacji technicznej bez danych

osobowych, organizacja nie potrzebuje ceremonii porównywalnej z wdrożeniem systemu oceniającego kadrę. Rygor musi rosnąć wraz z potencjalną krzywdą – to prawna wersja gradientu autonomii. W tym właśnie miejscu RAUC Cassaniego znajduje najdojrzalszą interpretację.

Źródło opisuje checklistę odpowiedzialnego użycia w trzech momentach: przed uruchomieniem AI, w trakcie działania oraz po zakończeniu. Przed startem kluczowe są pytania o cel, alternatywy bez AI, legalność danych, potencjalne uprzedzenia i fallback; w trakcie – o zgodność z architekturą, anomalie i koszty; po użyciu – o osiągnięcie celu i problemy ujawnione przez proces.

Odpowiedzialność za AI należy architekturyzować, a nie antropomorfizować

RAUC nie należy traktować jako źródła prawa. Jego siła tkwi w czymś innym: może stać się mechanizmem tłumaczącym abstrakcyjną rozliczalność na konkretny moment decyzji inżyniera. Rozliczalność jest bowiem jednym z kluczowych pojęć ery agentowej. RODO wprost wymaga, aby administrator nie tylko przestrzegał zasad, lecz był zdolny wykazać to przestrzeganie. W świecie agentów rodzi to naturalne zapotrzebowanie na provenance: skąd pochodził kontekst, który model wykonano i z jakimi ustawieniami, jakie narzędzia były dostępne, kto zatwierdził zmianę, jaki był wynik testów i czy rezultat odbiegał od planu. Logbook Cassaniego przestaje wtedy być wyłącznie notatką techniczną, a staje się potencjalnym elementem łańcucha rozliczalności.

Nie oznacza to jednak, że każdą myśl modelu należy archiwizować. Taki postulat byłby technicznie nierozsądny, prawnie ryzykowny i poznawczo mało użyteczny. Audytowalność nie polega na gromadzeniu wszystkiego, lecz na zbieraniu informacji wystarczających do rekonstrukcji istotnych decyzji i zdarzeń. Tutaj również obowiązuje zasada minimalizacji. Doskonały ślad audytowy nie jest cyfrowym oceanem logów, w którym po incydencie nikt nie potrafi odnaleźć właściwego zdarzenia. Jest krótką, wiarygodną historią odpowiedzialności.

Podobna filozofia przejawia się w NIS2. Choć dyrektywa ta nie jest regulacją AI jako taką i nie obejmuje automatycznie każdej firmy korzystającej z modelu, to dla podmiotów w jej zakresie ustanawia istotną logikę zarządzania cyberbezpieczeństwem. Organy zarządzające mają zatwierdzać środki zarządzania ryzykiem i nadzorować ich wdrażanie; system ten musi obejmować m.in. analizę ryzyka, obsługę incydentów, ciągłość działania, bezpieczeństwo łańcucha dostaw, rozwój i utrzymanie systemów, ocenę skuteczności zabezpieczeń, szkolenia, kryptografię oraz kontrolę dostępu. Jest to kluczowe dla agentowej AI, ponieważ dostawca modelu, framework

agentowy i zewnętrzne narzędzia stają się częścią cybernetycznego łańcucha dostaw przedsiębiorstwa.

W klasycznym supply chain risk management pytano o podatności bibliotek czy standardy dostawcy chmury. W systemie agentowym trzeba dodatkowo pytać: co dostawca może zrobić z przekazanym kontekstem, czy model może wywołać narzędzie, jak chronione są poświadczenia, czy zmiana wersji modelu wpływa na zachowanie procesu i czy organizacja posiada fallback na wypadek awarii lub zmiany polityki dostawcy. W ten sposób vendor assessment przestaje być kwestionariuszem procurementu, a staje się częścią architektury bezpieczeństwa.

NIS2 dostarcza również ważnej lekcji ustrojowej: cyberbezpieczeństwo nie powinno być delegowane wyłącznie do administratora systemu. Na poziomie unijnym dyrektywa wymaga, aby organy zarządzające podmiotów kluczowych i ważnych zatwierdzały środki ryzyka i nadzorowały ich wdrażanie; przewidziano także wymóg szkoleń dla członków tych organów. Jest to bardzo bliskie zasadzie, którą należałoby przyjąć dla AI governance: zarząd może delegować wykonanie kontroli, ale nie powinien delegować wiedzy o istnieniu ryzyka.

Problem odpowiedzialności nie kończy się jednak na administracyjnym compliance. Unia przebudowała reżim odpowiedzialności za produkty wadliwe tak, aby wyraźnie obejmował on m.in. software i systemy AI. Nowa dyrektywa 2024/2853 ma zostać transponowana do prawa państw członkowskich do 9 grudnia 2026 roku i będzie dotyczyć produktów wprowadzanych do obrotu lub oddawanych do użytku po 8 grudnia 2026 roku (zgodnie ze sprostowaniem z maja 2026 r.). Z perspektywy 18 sierpnia 2026 roku jest to zatem nadchodząca warstwa odpowiedzialności, a nie reżim, który można bezkrytycznie stosować do każdego dzisiejszego incydentu.

Taka perspektywa pozwala uniknąć futurystycznej pułapki "kto pozwie robota?". Prawo nie musi czynić robota osobą, aby rozliczać szkody związane z software'em. Może budować łańcuch odpowiedzialności wokół podmiotów gospodarczych, które projektują, wprowadzają, integrują, aktualizują lub wykorzystują produkt. Komisja wprost wskazuje, że zmodernizowane zasady odpowiedzialności produktowej obejmują software, AI i określone usługi cyfrowe powiązane z produktem.

Im bardziej autonomiczny staje się produkt, tym mniej sensowne jest pozwalanie producentowi na ukrycie się za argumentem, że "to model sam tak zrobił". W organizacji wewnętrznej podobna zasada powinna działać jeszcze wcześniej. Agent może być przyczyną techniczną, ale nie powinien być ostatecznym adresatem wyjaśnienia. Jeśli system wykonał niedozwoloną operację, pytamy nie tylko, dlaczego model ją wygenerował, lecz kto przyznał narzędziu uprawnienie, kto zaprojektował policy layer, kto dopuścił konfigurację do produkcji, kto określił próg nadzoru i jakie mechanizmy

miały ograniczyć blast radius. Tak właśnie powinien działać blameless post-mortem: nie w sensie moralnej obojętności, lecz bez redukowania analizy do znalezienia pierwszej osoby stojącej najbliżej incydentu. Jest to również odpowiedź na wielkie pytanie o "odpowiedzialność AI".

Nie należy jej antropomorfizować. Należy ją architekturyzować. Odpowiedzialność musi być wpisana w role, uprawnienia, procedury eskalacji, logi, możliwość odwołania, dokumentację oraz kompetencje osób nadzorujących. Im bardziej rozproszone staje się wykonanie, tym bardziej jednoznaczne muszą stać się granice odpowiedzialności. Szczególnie dramatycznie widać to przy bias.

Ryzyko uprzedzeń w kodzie a wymogi AI Act

Materiał źródłowy przywołuje szeroki przedział rzekomego udziału ukrytych uprzedzeń w kodzie generowanym przez popularne LLM, lecz nie dostarcza wystarczającego aparatu dowodowego, aby ten zakres — od 13 do niemal 50 procent — traktować w niniejszym artykule jako zweryfikowany wynik ogólny. Problem dyskryminacji jest jednak realnie obecny w logice AI Act: systemy wykorzystywane w obszarach takich jak zatrudnienie czy dostęp do kredytu są klasyfikowane właśnie jako szczególnie wrażliwe. Przyszły reżim wysokiego ryzyka obejmuje zatem wymagania dotyczące jakości danych, dokumentacji, traceability, human oversight, odporności oraz monitorowania.

Odpowiedzialność nie jest delegowalna do algorytmu

Najciekawsza z filozoficznego punktu widzenia jest granica między legalnością a słusznością. System może być w pełni zgodny z formalną procedurą, a mimo to generować decyzję błędną biznesowo, społecznie lub moralnie. Prawo nie jest algorytmem dobra. Wyznacza jedynie minima, zakazy i mechanizmy odpowiedzialności, ale nie odpowiada zarządowi na pytanie, czy każda legalna automatyzacja jest jednocześnie mądra. To właśnie w tym miejscu niezbędna staje się aksjologia organizacji. Czy agentowi wolno maksymalizować odzyskiwanie należności w sposób, który niszczy długoterminową relację z klientem? Czy algorytm optymalizujący obsadę może traktować człowieka wyłącznie jako wektor dostępności? Czy proces rekrutacji powinien automatyzować wszystko, co technicznie możliwe, nawet jeśli pozbawia to kandydatów elementarnego poczucia bycia wysłuchanym? I czy bank powinien delegować agentowi niemal całą decyzję kredytową tylko dlatego, że finalne kliknięcie wykonuje człowiek?

Hans Jonas, pisząc o etyce technologicznej, zwracał uwagę, że wraz ze wzrostem mocy działania rośnie zakres odpowiedzialności za jego odległe konsekwencje. Agentowa AI jest podręcznikowym przykładem tej intuicji. Im taniej możemy działać na wielką skalę, tym więcej uwagi musimy poświęcać temu, co właściwie skalujemy. Błąd człowieka przy pojedynczej decyzji jest tragedią jednostkową; ten sam błąd wpisany w automat może zostać powtórzony dziesięć tysięcy razy przed lunchem.

Dlatego prawdziwy HITL należy rozumieć nie tylko jako nadzór, ale jako zasadę etyczną. Człowiek ma nie tylko zatwierdzać – musi zachować prawo do zakwestionowania celu, kryteriów i samej potrzeby automatyzacji. Agent może stwierdzić: „najbardziej efektywną metodą jest X". Człowiek musi mieć możliwość odpowiedzi: „nie chcemy być organizacją, która robi X". W tym momencie odpowiedzialność przestaje być kwestią compliance, a staje się tożsamością instytucji. RAUC można zatem odczytać jako mikroetykę decyzji technicznej. Zanim uruchomimy AI, pytamy nie tylko o to, czy potrafimy, lecz czy powinniśmy; w trakcie działania obserwujemy nie tylko skuteczność, lecz i odchylenia; po fakcie oceniamy nie tylko osiągnięcie celu, ale tego, czego organizacja nauczyła się o własnych granicach. Wtedy lista kontrolna przestaje być zwykłym check-boxem compliance, a staje się rytuałem instytucjonalnej refleksji.

Najważniejsza zasada brzmi: delegować można wykonanie, ale nie można delegować ostatecznej odpowiedzialności za to, jaką władzę organizacja pozwoliła wykonać w swoim imieniu. Agent nie wybiera swojego miejsca w przedsiębiorstwie. Ktoś nadaje mu cel, dane, narzędzia, zakres swobody i prawo przejścia od odpowiedzi do działania. To właśnie w tych pięciu decyzjach rodzi się odpowiedzialność – na długo przed wystąpieniem błędu. Dojrzała organizacja agentowa musi potrafić odpowiedzieć na sześć pytań bez konieczności wszczynania śledztwa po fakcie: kto określił cel, kto udostępnił dane, kto przyznał uprawnienia, kto zatwierdził poziom autonomii, kto może zatrzymać działanie i kto ostatecznie odpowiada za rezultat wobec człowieka, klienta, regulatora oraz własnej organizacji. Jeśli odpowiedź brzmi: „to zależy, który agent to zrobił", governance już przegrał.

Przejście z modelu eksperymentalnego na dojrzały ustrój operacyjny

W ten sposób zataczamy niemal pełne koło. Zaczynaliśmy od modelu generującego fragment kodu, a doszliśmy do systemu posiadającego pamięć, narzędzia, uprawnienia, ekonomię, strukturę organizacyjną i miejsce w porządku prawnym. AI przestała być jedynie funkcją IDE; staje się elementem instytucji. A każda instytucja wymaga nie tylko technologii, lecz reguł

odpowiedzialności, kultury sprzeciwu i ludzi zdolnych stwierdzić, że rezultat zgodny z modelem nie jest jeszcze decyzją zgodną z interesem człowieka.

Pojawia się zatem pytanie: jak mierzyć dojrzałość całej transformacji i jak prowadzić przedsiębiorstwo od eksperymentu do agentowego modelu operacyjnego bez technologicznej gorączki? To właśnie tutaj RACM, PAIP, ECF/IDVI i RAUC można złożyć w jeden system adopcji. Połączony z DORA, governance, budżetem, kompetencjami i kulturą, pozwala on odpowiedzieć na pytanie zarządu ważniejsze niż „jakie AI kupić?": w jakiej kolejności przebudować organizację, aby inteligencja maszynowa nie wyprzedziła inteligencji instytucjonalnej?

Od eksperymentu do ustroju operacyjnego: dojrzałość adopcji, cztery ramy Cassaniego i pętla instytucjonalnego uczenia się.

Niebezpiecznym momentem adopcji AI wcale nie musi być jej początek. Pierwsze kroki zazwyczaj towarzyszą ostrożność, ciekawość, ograniczone pilotaże i świadomość eksperymentalnego charakteru narzędzia. Paradoksalnie większe ryzyko pojawia się później, gdy pierwsze sukcesy są interpretowane jako dowód na to, że technologia została już opanowana. Kilka udanych demonstracji rodzi przekonanie o konieczności „skalowania AI", w efekcie czego organizacja zwiększa liczbę licencji, agentów i przypadków użycia szybciej, niż rozwija system kontroli, dane, kompetencje kadr i zdolność mierzenia efektów. Między eksperymentem a skalą znajduje się najważniejszy próg dojrzałości: moment, w którym organizacja musi przestać uczyć się wyłącznie narzędzia, a zacząć przebudowywać samą siebie.

Cassani proponuje cztery komplementarne ramy, które w rozproszeniu pojawiały się już w poprzednich częściach tej analizy. RACM mapuje możliwości narzędzi, ich poziom autonomii i niezawodność; PAIP określa sposób ich włączania do pracy zależnie od krytyczności zadania; ECF wraz z IDVI organizuje współpracę człowieka i AI wokół intencji, dialogu, walidacji oraz integracji; RAUC natomiast tworzy praktyczny rytuał odpowiedzialnego użycia przed, podczas i po uruchomieniu systemu. Ich największa wartość ujawnia się jednak wtedy, gdy przestaniemy traktować je jako osobne diagramy, a zobaczymy w nich cztery pytania jednego systemu zarządczego.

RACM odpowiada na pytanie: co naprawdę potrafimy? PAIP pyta: ile wolno nam delegować w konkretnym przypadku? ECF pyta: jak zorganizować wspólne dochodzenie do właściwego rozwiązania? RAUC pyta wreszcie: jak upewnić się, że zdolność techniczna pozostaje zgodna z bezpieczeństwem, prawem i interesem organizacji? W tej kolejności powstaje znacznie dojrzalsza logika adopcji niż popularna sekwencja: kup narzędzie, daj ludziom dostęp, policz logowania i ogłoś transformację.

Najpierw trzeba bowiem przeprowadzić inwentaryzację rzeczywistości. RACM Cassaniego nie powinien być rozumiany jako ranking modeli, lecz jako dynamiczna mapa związku pomiędzy narzędziem, zadaniem, poziomem autonomii i jakością rezultatu. Należy aktualizować tę mapę wraz z ewolucją modeli oraz dopasowywać intensywność ludzkiej weryfikacji do stabilności konkretnego zastosowania.

To zasadnicza różnica wobec zakupowego myślenia o AI. Model nie jest „dobry" albo „zły" w abstrakcji; jest mniej lub bardziej odpowiedni do określonej funkcji w określonym systemie ryzyka. W praktyce oznacza to, że przedsiębiorstwo powinno rozpoczynać nie od katalogu modeli, lecz od katalogu pracy. Gdzie w SDLC występują zadania powtarzalne? Gdzie koszt błędu jest mały i odwracalny?

Adopcja AI jako problem systemowy, a nie zakupowy

Gdzie istnieją dobre testy? Gdzie wiedza domenowa jest wystarczająco opisana? W których obszarach największym wąskim gardłem jest tworzenie kodu, a gdzie problemem pozostaje decyzja biznesowa, review, integracja lub dostęp do danych? Dopiero wtedy narzędzie znajduje zadanie, zamiast zadania wymyślanego po to, by uzasadnić zakup narzędzia. To pierwsza cecha organizacji dojrzałej: nie szuka problemów dla posiadanej AI; szuka AI dla prawdziwych problemów. Takie podejście jest zgodne z aktualnym nurtem badań DORA.

Model AI Capabilities z 2025 roku powstał jako odpowiedź na pytanie, dlaczego samo przyjęcie narzędzi nie gwarantuje lepszych wyników. DORA, opierając się na wywiadach pogłębionych, literaturze eksperckiej i niemal pięciu tysiącach odpowiedzi ankietowych, wyróżniła siedem zdolności organizacyjnych wzmacniających pozytywne efekty adopcji AI. Należą do nich: jasne i komunikowane stanowisko organizacji wobec AI, zdrowy ekosystem danych, możliwość bezpiecznego udostępniania AI danych wewnętrznych, dojrzałe praktyki kontroli wersji, praca w małych partiach zmian, orientacja na użytkownika oraz wysokiej jakości platforma wewnętrzna. To zestawienie jest kluczowe, gdyż z siedmiu warunków żaden nie brzmi „kup najlepszy model". DORA dochodzi do wniosku zgodnego z architekturą Cassaniego: AI adoption jest problemem systemowym, nie zakupowym. Oficjalny raport 2025 opisuje AI jako wzmacniacz istniejących właściwości organizacji, a nie magiczny substytut ich braku.

Można zatem zaproponować pierwszą fazę dojrzewania jako etap rozpoznania, a nie implementacji. Organizacja powinna najpierw poznać własne procesy, dane, praktyki testowe, kulturę pracy, ograniczenia prawne, bezpieczeństwo i jakość platformy. Powinna również sformułować własne stanowisko wobec AI. DORA pokazuje, że jasność w kwestii dozwolonych narzędzi, oczekiwań i

zakresu eksperymentowania sama w sobie jest zdolnością zwiększającą skuteczność adopcji. To niezwykle praktyczna lekcja.

Niepewność regulacyjna w firmie nie zawsze zwiększa ostrożność. Często tworzy dwa światy: część pracowników boi się używać nawet bezpiecznych narzędzi, a inni korzystają z nich po cichu, bez żadnego governance. Pierwszą kompetencją organizacji nie jest więc prompt engineering, lecz zdolność jednoznacznego powiedzenia pracownikowi, co wolno, czego nie wolno i dlaczego. Dopiero później powinien pojawić się pilotaż.

Cassani w swoim studium transformacji FinTechu opisuje etap stopniowego wprowadzania narzędzi, szkoleń i „AI Fridays", podczas których zespół eksperymentuje, a skuteczne praktyki trafiają do wspólnej bazy wiedzy. Należy jednak zachować metodologiczną ostrożność: opis ten jest elementem studium konkretnego autora, a przytoczone wskaźniki velocity i ROI nie stanowią uniwersalnie zweryfikowanych benchmarków rynkowych. Wartość samej koncepcji pilotażu leży gdzie indziej. Pilot powinien być maszyną do produkowania wiedzy, nie prezentacją do produkowania entuzjazmu. Zbyt wiele projektów pilotażowych ocenia się przez pryzmat atrakcyjnego demo. Tymczasem naprawdę użyteczny pilot powinien odpowiedzieć na pytania: w jakich zadaniach model przynosi korzyść, gdzie popełnia typowe błędy, ile czasu pochłania weryfikacja, jaki jest koszt tokenów, czy ludzie rozumieją rezultaty, jak zachowuje się bezpieczeństwo i czy rozwiązanie przechodzi z jednego eksperta na resztę zespołu. Sukcesem pilotażu może być również ustalenie, że określony use case nie nadaje się jeszcze do automatyzacji. AI pilot powinien zatem przypominać eksperyment naukowy bardziej niż kampanię wdrożeniową.

Musi istnieć hipoteza, stan bazowy, obserwacja, kryterium sukcesu i możliwość uzyskania wyniku negatywnego. Jeżeli z góry ustalono, że projekt ma dowieść opłacalności AI, przestał być eksperymentem i stał się rytuałem potwierdzającym decyzję podjętą wcześniej. NIST AI Risk Management Framework wzmacnia właśnie takie podejście.

Dojrzałość AI to kalibracja autonomii do poziomu ryzyka

Podstawowa architektura NIST obejmuje cztery funkcje: Govern, Map, Measure i Manage. Governance ma charakter przekrojowy; Map służy rozumieniu kontekstu i ryzyka, Measure – analizie, testowaniu oraz monitorowaniu, a Manage – priorytetyzacji ryzyk i podejmowaniu decyzji o dalszym wykorzystaniu systemu. NIST podkreśla, że nie jest to prosta, sekwencyjna checklista, lecz ciągły system zarządzania ryzykiem w całym cyklu życia AI.

Zestawienie NIST z koncepcjami Cassaniego okazuje się szczególnie płodne. Choć nie można twierdzić, że frameworki te są bezpośrednią implementacją wytycznych NIST, na poziomie syntezy widać wyraźną zgodność logik. RACM współgra z funkcją Map, wymuszając rozpoznanie kontekstu, kompetencji i ograniczeń. Evals, testy i metryki odpowiadają logice Measure, natomiast PAIP i RAUC pomagają przełożyć zidentyfikowane ryzyko na konkretne reguły Manage. Governance otacza wszystkie te elementy, ustalając role, odpowiedzialność i granice dopuszczalnego działania. NIST rekomenduje dodatkowo budowanie profili stanu obecnego i docelowego oraz identyfikację luki między nimi, co jest kluczowe dla oceny dojrzałości adopcji.

Właśnie dlatego dojrzałość nie powinna być rozumiana jako schody prowadzące od „braku AI" do „maksymalnej autonomii". Taki model byłby intelektualnie błędny. Najbardziej dojrzała organizacja może celowo pozostawić część procesów bez wsparcia AI lub na najniższym poziomie sprawczości, jeśli koszt błędu tego wymaga. Dojrzałość oznacza zatem zdolność trafnego wyboru poziomu autonomii, a nie dążenie do jego maksimum. To jedna z najważniejszych korekt wobec potocznego rozumienia postępu technologicznego.

W motoryzacji dojrzałość nie polega na jeździe z maksymalną prędkością, w medycynie – na stosowaniu najbardziej inwazyjnych procedur, a w finansach – na maksymalnym wykorzystaniu dźwigni. W AI dojrzałość jest przede wszystkim kalibracją mocy do ryzyka. PAIP Cassaniego stanowi właśnie narzędzie takiej kalibracji.

Źródło rozróżnia tryb wysokiej autonomii dla eksploracji, agent-supervised dla zadań o średnim profilu ryzyka oraz ścisłe AI-assisted dla obszarów produkcyjnych wymagających maksymalnej kontroli człowieka. Podawane procenty autonomii nie powinny być traktowane jak miary fizyczne – są jedynie heurystyką. Sednem pozostaje dopasowanie reżimu działania do konsekwencji błędu.

Druga faza dojrzałości polega więc na standaryzacji tego dopasowania. Pilotaż pozwolił ustalić, gdzie AI działa; teraz organizacja musi określić, jak ma działać za każdym razem. W tym miejscu pojawiają się zatwierdzone modele, szablony Execution Plan i Logbook, biblioteki kontekstów, pule skonfigurowanych agentów oraz wspólne zasady review, evals i mechanizmy eskalacji. Cassani nazywa ten zestaw firmowym AI Toolbox, obejmującym Context Libraries, Agent Pools, Validation Pipelines oraz Knowledge Base dokumentującą możliwości i ograniczenia modeli. Jest to moment przejścia od sztuki jednostkowej do zdolności instytucjonalnej.

Dopóki sukces zależy od jednego utalentowanego pracownika, który „wie, jak rozmawiać z Claude'em" lub „ma świetne prompty do Copilota", organizacja nie posiada kompetencji AI – posiada jedynie kompetentnego człowieka. Dopiero gdy jego odkrycia zostaną przekształcone w

wersjonowane konteksty, procedury i zasady dostępne dla zespołu, powstaje aktywo przedsiębiorstwa. Michael Polanyi pisał, że „wiemy więcej, niż potrafimy powiedzieć".

Problem organizacji polega na tym, że odchodzący pracownik zabiera ze sobą część tej niewypowiedzianej wiedzy. Agentowa transformacja może działać w przeciwnym kierunku: wymuszać eksplicytację kontekstu, reguł i kryteriów, ponieważ maszyna potrzebuje ich do poprawnego działania. Dobrze wdrożona AI może więc stać się instrumentem ujawniania wiedzy ukrytej, choć źle zaimplementowana może jednocześnie tworzyć nową „ciemną wiedzę" w postaci kodu, którego ludzie już nie rozumieją. Trzecim etapem jest budowanie platformy.

Governance jako infrastruktura i systemowa miara dojrzałości

W tym miejscu współczesne DORA niezwykle mocno uzupełnia Cassaniego. Dane z 2025 roku wskazują, że wysokiej jakości platforma wewnętrzna stanowi jedną z siedmiu zdolności wzmacniających korzyści z AI. DORA opisuje ją jako warstwę pozwalającą przełożyć indywidualne przyspieszenie pracy deweloperów na bezpieczny wynik całej organizacji poprzez standaryzację testowania, bezpieczeństwa i wdrażania. To właśnie platforma powinna zamienić zasady w mechanizmy.

Zamiast przypomnienia „nie wysyłaj sekretów" – system blokujący ich wysyłkę. Zamiast zalecenia „używaj zatwierdzonego modelu" – katalog zatwierdzonych endpointów. Nie „zrób security scan", lecz obowiązkowa brama CI. Nie „sprawdź koszty", lecz limity tokenów i observability. Nie „upewnij się, że agent nie ma za szerokich uprawnień", lecz infrastructure-as-policy egzekwująca least privilege. Właśnie wtedy governance dojrzewa z dokumentu do infrastruktury. Prawo organizacyjne przestaje istnieć wyłącznie w PDF-ie, a zaczyna funkcjonować w kodzie systemu.

Jest to szczególnie istotne przy udostępnianiu AI danych wewnętrznych. DORA zaleca łączenie systemów AI z dokumentacją i kodem organizacji, lecz jednocześnie ostrzega przed modelem „super-user". Dostęp powinien opierać się na zasadzie najmniejszych uprawnień i w miarę możliwości respektować uprawnienia konkretnego użytkownika. Łączy się to bezpośrednio z wcześniejszą analizą prompt injection, Excessive Agency oraz prawem ochrony danych. Kontekst musi być bogaty semantycznie i ubogi w niepotrzebne przywileje.

Czwarty etap można nazwać skalowaniem kontrolowanym. To moment największej pokusy, by bezrefleksyjnie kopiować działający wzorzec wszędzie. Właśnie wtedy do gry powinien ponownie

wejść RACM. Nowy dział, nowe repozytorium, nowa klasa danych czy nowy proces biznesowy oznaczają zawsze nowy kontekst ryzyka.

Skalowanie nie jest więc mechanicznym zwiększaniem liczby użytkowników, lecz powtarzaniem pętli: rozpoznaj, zmierz, wybierz reżim, wdroż, obserwuj, skoryguj. NIST podkreśla właśnie ciągłość tego procesu. Funkcja Manage nie kończy zarządzania ryzykiem; organizacja musi regularnie aktualizować priorytety wraz ze zmianą modeli, zastosowań i oczekiwań interesariuszy. Jest to kluczowe w obszarze AI, gdzie model z stycznia w sierpniu może zostać zastąpiony nową generacją, a dostawca może zmienić politykę danych, ceny lub zachowanie narzędzia. Stabilna polityka AI musi więc zarządzać niestabilną technologią.

Pomocna staje się tu idea current profile i target profile z NIST. Organizacja może precyzyjnie opisać, jak dziś zarządza systemem AI, jakiego stanu oczekuje i jaka luka dzieli te dwa punkty. W miejsce abstrakcyjnego hasła „musimy zwiększyć dojrzałość AI" pojawiają się konkretne niedobory: brak evals, niejasna polityka narzędzi, zbyt szerokie konta serwisowe, niewersjonowane konteksty, brak właściciela danych, nieskuteczny rollback czy brak miernika wartości biznesowej. Tak powinna wyglądać dojrzałość: nie jako ocena w skali 1–5, lecz jako katalog rzeczywistych zdolności, których organizacji brakuje do bezpiecznego wykonania zadania. Cassani proponuje w tym kontekście Team AI Maturity Score, wiążący adopcję, zysk wydajności, stabilność jakości i poziom oporu kulturowego.

Podobnie jak przy wcześniejszych wskaźnikach, nie należy traktować tego wzoru jako naukowo zwalidowanego indeksu. Jego wartość leży w przypomnieniu, że sam procent użytkowników AI nie jest miarą dojrzałości. Można wyobrazić sobie przedsiębiorstwo, w którym 95% zespołu korzysta z modeli, ale nikt nie zna zasad bezpieczeństwa, brakuje evals, nie mierzy się jakości, a każdy używa innego narzędzia. Statystycznie adopcja jest niemal pełna; instytucjonalnie panuje chaos.

DORA idzie w podobnym kierunku. W 2025 roku użycie AI było już szerokie, jednak raport podkreślał, że największe korzyści zależą od kultury, danych, praktyk technicznych i platformy, a nie od samego faktu posiadania narzędzi. Wniosek jest niemal tautologiczny, a mimo to gospodarczo rewolucyjny: adoption rate mierzy rozpowszechnienie narzędzia, nie jakość organizacji.

Właściwy system pomiaru musi zatem rozdzielać trzy poziomy. Pierwszy to poziom indywidualny: czas wykonania, subiektywna użyteczność, jakość rezultatu i koszt weryfikacji. Drugi to poziom procesu: cycle time, wielkość zmian, awarie, rework, lead time i przepustowość review. Trzeci to poziom produktu i organizacji: wartość dla użytkownika, time-to-market, jakość usługi, ryzyko, rentowność oraz zdolność organizacji do zmiany.

Jeżeli AI skraca pisanie kodu o połowę, lecz nie zmienia czasu od pomysłu do produkcji, stworzyliśmy lokalną produktywność bez produktywności systemowej. Jeżeli zwiększa liczbę funkcji, których użytkownicy nie potrzebują, stworzyliśmy output bez outcome. Jeśli poprawia cycle time, lecz drastycznie zwiększa ryzyko incydentu – poprawiliśmy średnią kosztem ogona rozkładu.

Przejście od lokalnej produktywności do systemowego zarządzania strumieniem wartości

Dlatego DORA wskazuje na user-centric focus jako jedną z siedmiu zdolności wzmacniających efekty AI. Badanie nie kończy się pytaniem o ilość dostarczonego kodu; nakazuje zachować orientację na to, co użytkownik faktycznie chce osiągnąć i czy produkt odpowiada na realną potrzebę. To niezbędna przeciwwaga dla kultury agentowej. Gdy koszt generowania kodu drastycznie spada, rośnie ryzyko budowania rzeczy, które są tanie do zbudowania, lecz bezwartościowe do posiadania. Dojrzały model operacyjny wymaga więc nowej zasady: AI nie ma maksymalizować produkcji, lecz skracać drogę od potrzeby do zweryfikowanej wartości.

To prowadzi nas do Value Stream Management. DORA 2025 opisuje VSM jako mechanizm, który sprawia, że lokalne zyski z produktywności nie utoną w downstream chaos, lecz przełożą się na rzeczywisty wynik produktu. W praktyce oznacza to analizę całego strumienia wartości, a nie tylko etapu kodowania. Gdzie praca utyka? Gdzie pojawia się rework? Gdzie decyzja czeka pięć dni? Gdzie review staje się wąskim gardłem, a bezpieczeństwo jest rozpatrywane dopiero na końcu?

Agent może skrócić czynność z ośmiu godzin do jednej, ale jeśli rezultat będzie potem tydzień czekał na akceptację, optymalizujemy niewłaściwy fragment systemu. Jedna z najstarszych prawd zarządzania mówi, że optymalizacja części może pogorszyć całość. AI sprawia, że ta zasada staje się bardziej aktualna, ponieważ dysproporcja prędkości między etapami procesu gwałtownie rośnie. Generacja trwa sekundy. Review pozostaje ludzkie. Decyzje prawne wymagają deliberacji. Produkcja ma swoje okna wdrożeniowe, a użytkownik zmienia zachowania w skali miesięcy.

Jeśli nie spojrzymy na cały strumień wartości, ulegniemy złudzeniu, zachwycając się najszybszym jego fragmentem. Piątą fazę dojrzewania można określić mianem agentowego modelu operacyjnego. Organizacja nie zastanawia się już przy każdym zadaniu od nowa, „jak użyć AI”. Posiada ustalone klasy ryzyka, zatwierdzony katalog narzędzi, biblioteki kontekstów, role, platformę, polityki danych, system evals, mechanizmy eskalacji i regularną retrospektywę. Agenci

nie są eksperymentalnymi gośćmi przedsiębiorstwa. Stają się elementami kontrolowanej infrastruktury pracy.

Nie oznacza to jednak końca transformacji. Właśnie wtedy zaczyna się najważniejsza zdolność: ciągłe instytucjonalne uczenie się. Cassani proponuje Structured AI Retrospectives, podczas których zespół analizuje efektywność interakcji, jakość kodu i ograniczenia modeli. Połączenie tej praktyki z logbookami, evals, metrykami DORA i rejestrem incydentów tworzy pętlę, w której każda sesja z AI zwiększa nie tylko ilość rezultatu, ale przede wszystkim wiedzę firmy o tym, jak używać AI następnym razem.

To różnica między przedsiębiorstwem korzystającym z technologii a przedsiębiorstwem uczącym się technologii. W pierwszym każdy pracownik odkrywa te same błędy osobno. W drugim jedno odkrycie aktualizuje firmową bibliotekę kontekstów. Jedna halucynacja prowadzi do nowego evala. Jeden incydent koryguje guardrail. Skuteczny sposób dekompozycji zadania staje się standardem, a problem z dostawcą trafia do RACM i obniża dopuszczalny poziom autonomii. Organizacja kumuluje doświadczenie szybciej niż pojedynczy człowiek. Wtedy powstaje prawdziwa przewaga organizacyjna.

Kapitał uczenia się i sztuka wycofywania AI

Model można kupić, prompt skopiować, a framework przeczytać. Znacznie trudniej jednak powielić tysiące lokalnych obserwacji zgromadzonych przez przedsiębiorstwo w toku rzeczywistej pracy: wiedzę o tym, które przypadki zawodzą, jakie konteksty są skuteczne i które modele okazują się niezawodne w konkretnych zadaniach. To zrozumienie tego, gdzie użytkownicy odrzucają rezultat, jakie zabezpieczenia są niezbędne i w którym momencie agent powinien oddać kontrolę człowiekowi, stanowi o kapitale uczenia się AI.

Dojrzała organizacja musi jednocześnie pielęgnować zdolność decommissioningu. NIST w funkcji Govern wprost wskazuje na potrzebę bezpiecznego wycofywania systemów AI, gdy przestają być one właściwe lub gdy wzrasta związane z nimi ryzyko. Jest to element często pomijany w entuzjastycznych mapach adopcji, podczas gdy każdy system ma swoją fazę końca życia. Model może zostać wycofany, zmienić warunki handlowe, ustąpić miejsca lepszemu następcy lub przestać spełniać wymogi bezpieczeństwa. Organizacja, która potrafi jedynie dodawać narzędzia, lecz nie umiejętność ich usuwać, nie zarządza portfelem – Kolekcjonuje zależności. Dlatego RACM powinien być nie tylko mapą adopcji, lecz także mapą degradacji i wycofywania zdolności.

Czasem dojrzałą decyzją będzie zwiększenie autonomii, innym razem jej obniżenie, migracja modelu lub całkowity powrót do procedury deterministycznej bądź ludzkiej. W ten sposób można zrekonstruować pełną architekturę adopcji: najpierw organizacja poznaje własny system, następnie wybiera ograniczone eksperymenty, a potem standaryzuje skuteczne praktyki w firmowym toolboxie. Dalej buduje platformę technicznie egzekwującą politykę, by dopiero później skalować przypadki użycia, stale powtarzając cykl Map-Measure-Manage. Ostatecznie ustanawia agentowy model operacyjny, w którym AI staje się częścią infrastruktury, lecz pozostaje ona podporządkowana ludzkim celom, miernikom wartości i zdolności do wycofania błędnej decyzji.

Nie jest to klasyczna droga „crawl, walk, run". Metafora biegu sugerowałaby, że najwyższy poziom rozwoju zawsze oznacza większą prędkość i więcej AI. Właściwszą figurą jest dojrzewanie systemu nerwowego. Z czasem organizacja nie tyle przyspiesza wszystkie reakcje, co uczy się rozróżniać, które z nich powinny być automatyczne, które wymagają refleksji, które należy zahamować i gdzie trzeba uruchomić ból ostrzegający przed uszkodzeniem. W takim ujęciu Andon Cord jest odruchem bólowym organizacji; RAUC – jej pamięcią normatywną; RACM – mapą aktualnych zdolności; PAIP – układem sterującym siłą wykonania; ECF/IDVI – protokołem komunikacji między poznaniem ludzkim a maszynowym; platforma – szkieletem technicznym, a Evals i Logbook – systemem sensorycznym i pamięcią.

Budowa organizacji zdolnej do bezpiecznego absorbowania inteligencji maszynowej

Leadership staje się ośrodkiem odpowiedzialności, który nie wykonuje każdego ruchu, lecz decyduje, jakie ruchy organizacja uznaje za dopuszczalne. Dopiero w tym ujęciu określenie "Intelligence Orchestrator" przestaje być metaforą stanowiska menedżera, a staje się nazwą nowej funkcji całego przedsiębiorstwa: orkiestrowania wielu źródeł inteligencji w taki sposób, aby szybkość nie niszczyła jakości, autonomia nie usuwała odpowiedzialności, a automatyzacja nie unicestwiała wiedzy potrzebnej do jej kontroli.

Błędem zarządu byłoby zatem pytanie: „kiedy będziemy w pełni AI?". Nie istnieje bowiem stan końcowy, w którym organizacja po prostu odhacza transformację. Modele ewoluują, prawo się zmienia, pracownicy zdobywają nowe kompetencje, pojawiają się nowe zagrożenia, a stare benchmarki tracą znaczenie. Standardy DORA i NIST prowadzą do wizji ciągłego doskonalenia, a nie jednorazowego wdrożenia. Pytanie powinno więc brzmieć inaczej: czy nasza zdolność rozumienia, mierzenia i kontrolowania AI rośnie co najmniej tak szybko, jak zdolność AI do działania? Jeśli odpowiedź brzmi „tak", autonomia może być stopniowo rozszerzana tam, gdzie

dowody to uzasadniają. Jeśli jednak brzmi „nie", zwiększanie liczby agentów nie jest skalowaniem, lecz powiększaniem luki poznawczej pomiędzy tym, co organizacja potrafi uruchomić, a tym, co potrafi rzeczywiście nadzorować.

Właśnie ta luka stanowi najgłębsze ryzyko transformacji. Możemy ją nazwać institutional intelligence gap - luką inteligencji instytucjonalnej. Powstaje ona wtedy, gdy możliwości techniczne rosną szybciej niż governance, kompetencje, prawo, pomiar i kultura organizacji. Firma zyskuje coraz większą zdolność do działania, ale jednocześnie coraz mniejszą pewność, czy rozumie konsekwencje tych działań. Historia techniki wielokrotnie pokazywała, że problemem cywilizacji rzadko była sama moc narzędzia; problemem bywała asymetria pomiędzy tą mocą a instytucją zdolną ją oswoić. Alfred Nobel stworzył dynamit nie po to, aby zbudować teorię wojny. Energia atomowa zmusiła ludzkość do skonstruowania nowych systemów międzynarodowej kontroli. Internet umożliwił komunikację szybciej, niż państwa i społeczeństwa nauczyły się zarządzać dezinformacją, prywatnością i koncentracją platform. Agentowa AI wpisuje się w tę samą historię: wynalazek dojrzewa technologicznie szybciej niż otaczająca go instytucja. Dlatego nadrzędnym celem adopcji nie powinno być dogonienie modeli – one i tak zmienią się ponownie.

Celem powinno być zbudowanie organizacji posiadającej zdolność bezpiecznego absorbowania kolejnych generacji inteligencji maszynowej. Takiej, która nie musi co roku przeżywać rewolucji, ponieważ posiada trwałe zasady mapowania zdolności, kalibracji autonomii, walidacji, odpowiedzialności i uczenia się. Wtedy RACM, PAIP, ECF i RAUC okazują się czymś więcej niż narzędziami zarządczymi Cassaniego; układają się w próbę odpowiedzi na stare pytanie teorii instytucji: jak korzystać z rosnącej sprawczości, nie tracąc nad nią suwerenności?

Pozostaje jednak ostatnia i najtrudniejsza warstwa opowieści. Nauka potrafi mierzyć benchmarki, przepustowość, jakość kodu i zachowania agentów. Ekonomia liczy ROI. Prawo przypisuje obowiązki, a governance określa granice. Żaden z tych aparatów samodzielnie nie odpowiada jednak na pytanie, kim stanie się człowiek w organizacji, w której znaczną część poznawczego wykonania można delegować maszynie. Ostatnie części tej analizy muszą więc wyjść poza samą adopcję i zapytać o przyszłość pracy intelektualnej, sens profesjonalizmu oraz odpowiedzialność za osąd. Musimy rozważyć, czy epoka Agent AI prowadzi ku deprecjacji człowieka, czy przeciwnie – ku ponownemu odkryciu tych kompetencji, których nie potrafiliśmy właściwie cenić w epoce, gdy mierzyliśmy człowieka liczbą napisanych przez niego linii kodu.

Powierzchowna wizja przyszłości pracy przedstawia człowieka i sztuczną inteligencję jako konkurentów ustawionych przy tej samej taśmie produkcyjnej. Człowiek pisze kod, AI pisze kod; człowiek analizuje dokument, AI analizuje dokument; człowiek sporządza specyfikację, AI

sporządza specyfikację. Następnie mierzymy prędkość, koszt i jakość, aby rozstrzygnąć, kto powinien pozostać na stanowisku.

Taka konstrukcja jest wygodna dla nagłówków, lecz uboga poznawczo. Historia automatyzacji pokazuje bowiem, że najgłębsza zmiana nie zachodzi wtedy, gdy maszyna wykonuje dokładnie tę samą czynność szybciej, lecz gdy automatyzacja zmienia definicję pracy pozostającej po stronie człowieka. Cassani formułuje swoją normatywną odpowiedź w postaci zasady "Automate to Augment, Don't Replace". Automatyzacji powinny w jego modelu podlegać przede wszystkim czynności powtarzalne – takie jak kod szablonowy czy część testowania i dokumentacji – podczas gdy człowiek zachowuje kontrolę nad architekturą, strategią produktu i relacjami z interesariuszami. Nie jest to oczywiście prawo technologicznego rozwoju.

Przedsiębiorstwo może używać AI właśnie po to, aby redukować zatrudnienie. Teza Cassaniego ma jednak charakter normatywnej doktryny organizacyjnej: najlepszym zastosowaniem automatyzacji ma być zwiększenie ludzkiej zdolności do podejmowania wartościowych decyzji, a nie mechaniczne eliminowanie człowieka z procesu.

Augmentacja jako pułapka poznawcza i nowa rola eksperta

Warto potraktować tę zasadę poważniej, niż pozwala na to marketingowe słowo „augmentacja”. Wzmocnienie człowieka nie oznacza bowiem jedynie wyposażenia go w szybsze narzędzie. Kalkulator usprawnia rachunki, lecz może jednocześnie ograniczyć potrzebę liczenia w pamięci. Nawigacja satelitarna zwiększa mobilność, ale osłabia konieczność budowania wewnętrznej mapy przestrzeni. Wyszukiwarka rozszerza dostęp do informacji, zmieniając przy tym sposób ich zapamiętywania. Każda technologia poznawcza jest jednocześnie protezą i nauczycielem zapominania. Pytanie nie brzmi zatem, czy AI odciąża nas poznawczo – bo robi to bezsprzecznie. Pytanie brzmi: które obciążenia można zdjąć bezpiecznie, a które są ćwiczeniem utrzymującym zdolności niezbędne do rozpoznania błędu maszyny?

Współczesne badania dostarczają pierwszych empirycznych sygnałów tej transformacji. Badanie Microsoft Research przedstawione na CHI 2025 objęło 319 pracowników wiedzy i 936 rzeczywistych przykładów wykorzystania generatywnej AI w pracy. Większe zaufanie do odpowiedzi modelu wiązało się w deklaracjach uczestników z mniejszym zaangażowaniem w krytyczne myślenie; natomiast większa pewność własnej wiedzy szła w parze z większym wysiłkiem analitycznym. Jednocześnie autorzy zaobserwowali przesunięcie samego charakteru myślenia: od

bezpośredniego wytwarzania treści ku jej weryfikacji, integracji i nadzorowi nad zadaniem. To rozróżnienie jest istotniejsze niż proste stwierdzenie, że „AI osłabia krytyczne myślenie". Badanie nie dowodzi takiego uniwersalnego prawa, lecz wskazuje raczej na zmianę miejsca, w którym to myślenie staje się niezbędne. Człowiek może poświęcać mniej czasu na konstruowanie pierwszej wersji rozwiązania, a więcej na ocenę, czy jest ono prawdziwe, wystarczające, bezpieczne i właściwe dla szerszego celu.

To dokładnie ten ruch, który w języku Cassaniego opisuje przejście od pisania ku Lead-Verify-Integrate. Problem polega na tym, że Verify jest czynnością poznawczo trudniejszą, niż sugeruje przycisk „Accept". Aby zakwestionować odpowiedź, trzeba posiadać wiedzę pozwalającą rozpoznać jej słabe punkty. Im bardziej przekonujące stają się modele, tym bardziej niebezpieczna jest niewiedza recenzenta. Łatwo skontrolować odpowiedź absurdalną; najtrudniej tę niemal doskonałą, zawierającą jeden błąd w miejscu, którego nie umiemy sprawdzić.

Epoka AI może zatem zwiększyć, a nie zmniejszyć wartość prawdziwej ekspertyzy. Ekspert nie będzie musiał pamiętać wszystkiego, co potrafi odnaleźć model, lecz będzie potrzebował wewnętrznych struktur wiedzy pozwalających ocenić sens rezultatu. Istnieje zasadnicza różnica pomiędzy zapomnieniem konkretnej funkcji biblioteki a nieumiejętnością rozpoznania błędnego modelu współbieżności.

Pierwsze z tych uchybień można bezpiecznie oddać wyszukiwarce lub agentowi. Drugie stanowi fundament profesjonalnego osądu. Można więc przewidywać przejście od eksperta magazynującego odpowiedzi ku ekspertowi posiadającemu mapę błędów. Najbardziej wartościowy senior nie musi najszybciej przypominać sobie składni. Wie natomiast, gdzie system zwykle pęka, jakie kompromisy powracają, której pozornie niewinnej decyzji nie należy podejmować, jakie pytanie zadać przed zmianą architektury i dlaczego klient wypowiada potrzebę inaczej, niż faktycznie ją przeżywa. To wiedza złożona z wyjątków, historii, analogii, blizn po wcześniejszych błędach i wyczucia konsekwencji.

Cassani słusznie przypisuje dlatego seniorom nową funkcję Active Mentorship. Zakłada on, że automatyzacja kodu szablonowego, podstawowych testów i prostego debugowania może odcinać młodszych inżynierów od tradycyjnej ścieżki zdobywania doświadczenia. Senior powinien więc odsłaniać nie tylko gotowe rozwiązanie, lecz sposób krytycznej oceny interakcji z AI, konstrukcję instrukcji, metodę testowania odpowiedzi i moment, w którym należy zachować sceptycyzm. Mentoring przestaje być przekazywaniem recept; staje się nauką rozpoznawania, kiedy recepta jest niewłaściwa. Pojawia się tu jednak problem wykraczający poza pojedynczą organizację.

Jeśli każda firma racjonalnie automatyzuje pracę juniorską, ponieważ AI robi to taniej i szybciej, całe środowisko zawodowe może stopniowo ograniczać sytuacje, w których przyszli eksperci zdobywają doświadczenie. Dla przedsiębiorstwa w bieżącym kwartale jest to oszczędność. Dla profesji za dziesięć lat może być to drastyczne uszczuplenie zasobu seniorów. Najnowsza literatura ujmuje ten problem w kategoriach dynamicznych.

Model ekonomiczny opublikowany jako preprint w 2026 roku pod nazwą The Augmentation Trap pokazuje teoretyczną możliwość sytuacji, w której menedżer racjonalnie zwiększa wykorzystanie AI dla krótkoterminowych zysków produktywności, choć długotrwałe odciążanie poznawcze obniża kompetencje pracownika tak bardzo, że jego przyszła wydajność spadnie poniżej poziomu sprzed adopcji technologii. Autorzy podkreślają, że rezultat zależy od charakteru zadania, zależności korzyści AI od poziomu ludzkiej ekspertyzy oraz horyzontu decyzyjnego menedżera.

Konieczność rozdzielenia produkcji od formacji kompetencji

Jest to model teoretyczny, a nie dowód na to, że taki proces już powszechnie zachodzi. Świeższa praca koncepcyjna z lipca 2026 roku wprowadza pojęcie cognitive commons – wspólnego zasobu kompetencji zawodowych, który każda organizacja może racjonalnie konsumować poprzez automatyzację, choć cały zawód potrzebuje jego ciągłej regeneracji. Autor rozróżnia głębokie mistrzostwo wewnętrzne, budowane przez praktykę, oraz mistrzostwo rozproszone, polegające na sprawnym orkiestrowaniu ludzi i AI. Formułuje również pojęcie "validation tether": zdolność skutecznego kontrolowania AI pozostaje zależna od rodzaju ekspertyzy, którą nadmierna automatyzacja może stopniowo ograniczać. To propozycja teoretyczna, wymagająca dalszej walidacji empirycznej, lecz trafnie nazywa problem widoczny już w dyskusji o juniorach. Można więc mówić o tragedii wspólnego pastwiska kompetencji. Każda firma ma bodziec, by nie finansować nauki początkującego pracownika, jeśli podobne zadanie wykona agent. Jednak wszystkie firmy jednocześnie będą kiedyś potrzebowały ludzi zdolnych audytować najbardziej złożone przypadki, projektować systemy, zarządzać incydentami i podejmować decyzje w warunkach niepewności.

Gdyby wszyscy korzystali z istniejącego zasobu seniorów, nie inwestując w jego reprodukcję, profesja zaczęłaby konsumować własny kapitał. Nie jest to argument za sztucznym utrzymywaniem nieefektywnej pracy; absurdem byłoby nakazywać pracownikom wykonywanie czynności ręcznie tylko dlatego, że tak robili ich poprzednicy. Chodzi o rozdzielenie produkcji od formacji. Zadanie biznesowe może nie wymagać ręcznego programowania, lecz szkolenie przyszłego eksperta może

wymagać ćwiczeń, w których ręczne rozumowanie pozostaje konieczne. Pilot nie musi codziennie odzyskiwać kontroli nad samolotem po awarii silnika, aby sens miało trenowanie takiej sytuacji w symulatorze. Organizacja AI będzie musiała świadomie tworzyć analogiczne "symulatory poznawcze". Z tego powodu najbardziej dojrzały model szkolenia może paradoksalnie okresowo ograniczać AI – nie jako karę i nie jako kult rękodzieła, lecz jako trening.

Człowiek powinien czasem najpierw sam zdiagnozować problem, zanim zobaczy diagnozę agenta; samodzielnie sformułować hipotezę, zanim otrzyma dziesięć wariantów; sam zaprojektować kryteria testu, zanim AI wygeneruje rozwiązania. Dopiero później można porównać tok rozumowania z maszyną. Pedagogika zna ten paradoks od dawna: nauczyciel matematyki nie podaje uczniowi wyniku równania tylko dlatego, że można go uzyskać natychmiast. Celem ćwiczenia nie jest wyprodukowanie liczby, lecz zmiana człowieka, który tę liczbę potrafi wyprowadzić.

Praca szkoleniowa dostarcza więc inny produkt niż praca produkcyjna: jej wynikiem jest kompetencja wykonującego. W przedsiębiorstwie łatwo o tym zapomnieć, gdyż systemy zarządzania nagradzają przede wszystkim output zewnętrzny. Ticket zamknięto albo nie. Funkcję dostarczono albo nie. Rzadko na dashboardzie pojawia się rubryka: "czego człowiek nauczył się podczas wykonywania zadania?". AI potęguje ten problem, umożliwiając całkowite rozdzielenie wyniku od rozwoju wykonawcy. Możemy osiągnąć znakomity rezultat przy pracowniku, który nie nauczył się niemal niczego. Możemy też świadomie dopuścić wolniejsze tempo pracy, aby zespół zwiększył kompetencję, której później będzie potrzebował w sytuacji krytycznej. Augmentacja powinna być zatem definiowana nie przez liczbę czynności odebranych człowiekowi, ale przez zmianę jego przyszłej zdolności sprawczej. Technologia wzmacniająca człowieka powinna po okresie użytkowania pozostawiać go zdolnym do wykonywania bardziej złożonych czynności, podejmowania lepszych decyzji lub kontrolowania szerszego systemu. Jeśli czyni go jedynie bardziej zależnym od kolejnego wywołania modelu, zwiększa bieżącą produktywność, ale nie ludzką sprawczość. Problem ten nie ogranicza się do programowania.

Analiza Microsoft Research z 2025 roku, oparta na około 200 tysiącach zanonimizowanych konwersacji z Copilotem, wskazała, że współczesna generatywna AI jest szczególnie stosowalna w obszarach pozyskiwania informacji, pisania, nauczania, doradzania i komunikacji; wysoką ekspozycję wykazują m.in. zawody wiedzy. Autorzy świadomie posługują się terminem AI applicability, a nie miarą zastępowalności zawodów. To rozróżnienie powinno wejść do debaty publicznej: zadanie podatne na wsparcie AI nie jest tożsame z zawodem gotowym do usunięcia. Zawód jest bowiem wiązką czynności, relacji, odpowiedzialności i wiedzy kontekstowej. Lekarz nie jest maszyną do generowania diagnozy. Prawnik nie jest generatorem akapitów. Architekt nie jest

urządzeniem wytwarzającym diagramy. Programista nie jest translulatorem wymagań na składnię. Redukowanie zawodu do najbardziej widocznego artefaktu pracy było błędem jeszcze przed AI; modele językowe jedynie go obnażyły. Może się okazać, że przez lata przecenialiśmy artefakt, a niedocenialiśmy osądu, który prowadził do jego powstania. Ceniliśmy kod, bo jego pisanie było kosztowne. Ceniliśmy prezentację i analizę, bo ich przygotowanie trwało godzinami. Gdy koszt wytworzenia artefaktu maleje, wartość przesuwa się ku pytaniu: kto potrafi rozpoznać, który artefakt w ogóle należy wytworzyć, która wersja jest właściwa i jaki skutek nastąpi po jej użyciu?

Tutaj pojawia się możliwość rehabilitacji arystotelesowskiej phronesis.

Ludzki osąd jako nadrzędna funkcja instytucjonalna

Wiedza techniczna odpowiada na pytanie, jak coś zrobić. Mądrość praktyczna dotyczy natomiast tego, co należy zrobić w konkretnej sytuacji, której nie da się całkowicie sprowadzić do reguły. Agent może perfekcyjnie stosować politykę; człowiek musi jednak czasem rozpoznać, że dany przypadek ujawnia wadę tej polityki. Najważniejszym ludzkim zasobem nie jest zatem „kreatywność" rozumiana romantycznie jako tajemnicza iskra niedostępna maszynie. Modele potrafią generować ogromną liczbę nowych kombinacji, a spór o wyjątkowość ludzkiej twórczości będzie ewoluował wraz z technologią. Bezpieczniej jest wskazać trzy inne elementy: odpowiedzialność, osadzenie i zdolność normatywnego sprzeciwu.

Człowiek jest osadzony w świecie konsekwencji. Rozmawia z klientem po awarii, patrzy pracownikowi w oczy podczas decyzji kadrowej, wyjaśnia zarządowi stratę czy pamięta nieformalną obietnicę złożoną partnerowi. Rozumie, że decyzja legalna może być reputacyjnie fatalna, a statystycznie optymalna – naruszyć sens relacji. Agent posiada kontekst tylko w takim zakresie, w jakim system potrafił go reprezentować. Człowiek żyje wewnątrz kontekstu.

Druga różnica dotyczy odpowiedzialności. Model może podać rekomendację, ale nie traci reputacji, stanowiska ani zaufania zespołu wskutek jej konsekwencji. Nie jest to wada techniczna modelu, lecz ontologiczna różnica jego miejsca w instytucji. Dlatego prawo i governance pozostawiają odpowiedzialność ludziom i organizacjom, co wykazaliśmy wcześniej. Odpowiedzialność nie jest niedoskonałością automatyzacji, lecz częścią ludzkiej funkcji instytucjonalnej.

Trzecim elementem jest prawo do sprzeciwu wobec funkcji celu. Maszyna odpowiada w obrębie zadanego celu; człowiek może zakwestionować sam cel. Jeśli agent optymalizuje liczbę zamkniętych zgłoszeń, człowiek może zauważyć, że organizacja zaczęła zbyt szybko zamykać sprawy klientów. Jeśli system maksymalizuje konwersję, człowiek może zapytać o manipulacyjność interfejsu. Gdy

algorytm redukuje koszty zespołu, lider może stwierdzić, że niszczy on ścieżkę rozwoju przyszłych ekspertów. Najwyższą kompetencją nie zawsze jest lepsza odpowiedź – czasami jest nią odmowa przyjęcia pytania w jego obecnej formie.

Cassani łączy dlatego przyszłość profesjonalizmu z rolą orkiestratora. Obawa przed utratą craftsmanship ma zostać zastąpiona nową formą dumy zawodowej: inżynierem jako krytycznym superwizorem, dbającym o semantyczną i architektoniczną jakość całego systemu. Ta odpowiedź wymaga jednak uzupełnienia. Sama zmiana etykiety z „programisty" na „orkiestratora" nie odbuduje sensu pracy. Człowiek musi realnie posiadać większy zakres wpływu i odpowiedzialności, zamiast stać się operatorem panelu zatwierdzającym decyzje w praktyce podjęte przez system.

Richard Sennett opisywał rzemiosło jako pragnienie wykonania rzeczy dobrze dla niej samej. Rewolucja AI nie musi unicestwić tego etosu, może jedynie przenieść jego przedmiot. Rzemiosłem nie będzie wyłącznie elegancja pojedynczej implementacji; może nim stać się elegancja systemu, jakość decyzji, precyzja kontekstu, prostota architektury, trafność walidacji i troska o utrzymywalność. Craftsmanship może przejść z dłuta do kompozycji. Dyrygent nie wydobywa każdego dźwięku z instrumentu, lecz nie jest przez to mniej muzykiem. Jego mistrzostwo polega na rozumieniu całości: czasu, relacji, napięcia, wejścia i ciszy. Metafora orkiestracji używana przez Cassaniego ma właśnie w tym miejscu największą siłę.

Programista przyszłości może wykonywać mniej bezpośrednich ruchów w kodzie, ale jego decyzje dotyczące tego, co powierzyć agentowi, jaki kontekst dostarczyć, kiedy przerwać trajektorię i co zaakceptować, mogą zyskać coraz większy zasięg. Jednocześnie istnieje niebezpieczeństwo automation complacency: jeśli większość procesów kończy się sukcesem, człowiek stopniowo przestaje poświęcać uwagę nadzorowi. Jest to znany problem automatyki, przenoszony teraz do pracy intelektualnej. Współczesna literatura HCI pyta zatem nie tylko o to, jak sprawić, by AI było skuteczniejsze, lecz jak projektować narzędzia chroniące i wzmacniające ludzkie poznanie, zamiast jedynie przejmować kolejne czynności. Synteza warsztatu CHI 2025 „Tools for Thought" wskazuje jako centralne pytania wpływ GenAI na metapoznanie, krytyczne myślenie, pamięć i kreatywność oraz potrzebę projektowania systemów chroniących aktywną rolę człowieka.

Oznacza to odejście od prostego wzorca „friction is bad". W projektowaniu interfejsów od lat usuwano tarcie: mniej kliknięć, mniej decyzji, szybsze wykonanie. W systemie poznawczym część tarcia może jednak pełnić funkcję zabezpieczającą. Pytanie kontrolne przed przelewem, potwierdzenie przed usunięciem bazy danych, code review czy konieczność uzasadnienia decyzji – to wszystko jest tarciem. Demokracja konstytucyjna również opiera się na tarciach celowo wprowadzonych po to, by żadna pojedyncza sprawczość nie była zbyt szybka. Możemy więc mówić o wartości produktywnego tarcia. Dobry system AI usuwa tarcie tam, gdzie nie tworzy ono wiedzy

ani bezpieczeństwa, i zachowuje je tam, gdzie zmusza człowieka do istotnego aktu osądu. To być może jeden z najtrudniejszych problemów projektowych kolejnej dekady: określić, kiedy przeszkoda jest biurokracją, a kiedy mechanizmem zachowania uwagi.

Badanie randomizowane Microsoft Research przeprowadzone wśród około sześciu tysięcy pracowników wiedzy pokazuje interesującą granicę automatyzacji organizacyjnej. Dostęp do generatywnej AI znacząco ograniczył czas poświęcany na e-maile i umiarkowanie przyspieszył pracę nad dokumentami, lecz nie zmienił w porównywalny sposób czasu spotkań. Autorzy interpretują to jako różnicę między zachowaniami, które jednostka może zmienić samodzielnie, a tymi wymagającymi koordynacji z innymi ludźmi. To kolejny dowód na to, że wydajność jednostki nie jest tożsama z wydajnością instytucji.

Asymetria tempa rozwoju jednostki i organizacji

AI potrafi przyspieszyć człowieka szybciej niż potrafi przyspieszyć relację między ludźmi.

Przesunięcie wartości z wykonawstwa na osąd

Relacje stanowią przecież znaczną część pracy seniora, lidera, architekta czy product managera. Wypracowanie kompromisu, przekonanie interesariusza, negocjowanie priorytetów, odbudowa zaufania po porażce czy decydowanie pod presją nie sprowadzają się do produkcji tekstu. AI może znakomicie przygotować materiały, zasymulować kontrargumenty lub podsumować spotkanie, ale organizacja wciąż potrzebuje podmiotu, któremu inni ludzie przypiszą legitymację do podjęcia decyzji. Dlatego przyszłość pracy nie musi być historią o człowieku odsuwanym coraz dalej od „prawdziwej roboty". Może stać się opowieścią o odkryciu, że część tego, co wcześniej uznawaliśmy za istotę pracy, była jedynie kosztem pośrednim dojścia do decyzji. Pisanie czterdziestu stron dokumentacji bywa potrzebne, lecz wartość nie tkwi w liczbie stron. Przegląd tysiąca linii logów może być niezbędny, ale wartością pozostaje trafne rozpoznanie przyczyny.

Jeśli AI eliminuje ten koszt pośredni, człowiek zyskuje czas na to, co wcześniej było dobrem rzadkim: rozmowę, refleksję, projektowanie, mentoring i rozważanie konsekwencji. Warunek jest jednak jeden: uwolnione godziny muszą zostać przesunięte ku zadaniom wyższego rzędu, a nie natychmiast wypełnione kolejnymi ticketami. Jeżeli organizacja wykorzysta każdą minutę oszczędzoną dzięki AI jedynie do zwiększenia ilości outputu, otrzyma tayloryzm poznawczy z turboładowaniem. Człowiek będzie wprawdzie dysponował potężniejszą maszyną, lecz zyska coraz

mniej przestrzeni na namysł. Tu pojawia się problem tempa. Frederick Taylor organizował pracę fizyczną, rozbijając ją na mierzalne ruchy.

Agentowa AI pozwala na podobne potraktowanie pracy poznawczej: poprzez coraz dokładniejsze pomiary liczby promptów, pull requestów, cycle time'u, tokenów czy zaakceptowanych zmian. Takie metryki są potrzebne, ale mogą zacząć zastępować sens działania, jeśli staną się celami samymi w sobie. Peter Drucker wskazywał, że produktywność pracownika wiedzy wymaga najpierw ustalenia, jakie zadanie jest właściwe. W epoce generatywnej AI ta intuicja nabiera szczególnego znaczenia – maszyna potrafi bowiem bardzo szybko wykonywać niewłaściwe zadania. Dlatego kluczową kompetencją przestaje być „robienie więcej", a staje się nią umiejętność wyboru problemu zasługującego na rozwiązanie.

Może to doprowadzić do przewartościowania hierarchii kompetencji. Przez dziesięciolecia organizacje często wynagradzały rzadkość umiejętności wykonawczej. Znajomość trudnego języka programowania, skomplikowanego frameworka czy systemu raportowego miała wysoką wartość, bo niewielu potrafiło samodzielnie wykonać zadanie. Gdy AI kompresuje koszt dostępu do wiedzy proceduralnej, na pierwszy plan wysuwają się wiedza domenowa, krytyczny osąd, architektura, odpowiedzialność i zdolność tworzenia znaczenia z niepełnego kontekstu. Nie jest to jednak triumf „soft skills" nad „hard skills".

Nowa definicja profesjonalizmu jako suwerenna orkiestracja znaczeń

Takie przeciwstawienie jest już przestarzałe. Dobry osąd techniczny musi być jednocześnie twardy i kontekstowy. Architekt powinien znać technologię, ale i rozumieć organizację; lider musi rozumieć ludzi oraz ograniczenia systemu, a product owner – rynek i konsekwencje implementacji. W świecie agentowym największą wartość zyskują kompetencje leżące na przecięciu domen, których korporacyjne struktury stanowisk długo nie potrafiły właściwie nazwać. Dlatego analityk biznesowy w ujęciu Cassaniego ewoluuje ku Business Context Engineering: formalizuje reguły biznesowe, ontologie domenowe i słowniki pojęciowe niezbędne agentom do poprawnego działania.

Jest to przykład szerszej transformacji. Wiedza „między działami", dotąd uznawana za miękką lub trudną do wyceny, staje się podstawową warstwą operacyjną systemów AI. Wartość rośnie tam, gdzie następuje kompresja rzeczywistości do znaczenia. To człowiek musi wyjaśnić agentowi, że dwa podobne pojęcia prawne nie są wymienne. Musi wiedzieć, że klient mówiący o „prostszym procesie" w rzeczywistości dąży do zredukowania momentów niepewności. Musi rozumieć, że stary

wyjątek w kodzie nie jest przypadkowym bałaganem, lecz pozostałością po zobowiązaniu kontraktowym. To właśnie jest żywa wiedza instytucji. Paradoksalnie AI może zatem zwiększyć znaczenie humanistyki w technologii. Systemy agentowe wymagają bowiem nie tylko kodu, lecz modelu znaczeń, norm, języka, intencji i zrozumienia interesariuszy. Im głębiej technologia ingeruje w decyzje, tym mniej wystarcza wiedzy czysto technicznej. Trzeba rozumieć, dla kogo, po co, według jakich wartości i z jakimi konsekwencjami system działa.

W tym miejscu humanizm nie jest dekoracją technologii, lecz warstwą jej specyfikacji. Dojrzała wersja hasła human-in-the-loop powinna zatem brzmieć nie „człowiek pozostaje w procesie", lecz „człowiek zachowuje znaczącą funkcję, której wykonania nie redukujemy do formalnego zatwierdzenia". Czasem będzie nią osąd, czasem odpowiedzialność, interpretacja norm lub empatia wobec osoby dotkniętej decyzją. Być może okaże się nią znajomość kontekstu niewidocznego dla modelu, a czasem po prostu zdolność powiedzenia „stop".

Prowadzi to do fundamentalnej zmiany definicji profesjonalizmu. Profesjonalista epoki industrialnej był kimś, kto potrafił wykonać pracę zgodnie ze sztuką. Profesjonalista epoki agentowej będzie natomiast osobą, która potrafi ustanowić sztukę wykonania dla układu ludzi i maszyn, rozpoznać jej naruszenie oraz wziąć odpowiedzialność za rezultat, którego sama nie wytworzyła w każdym detalu. Tak rozumiana profesja nie znika – staje się trudniejsza. Nie wystarczy już wiedzieć; trzeba wiedzieć, czego można nie wiedzieć. Nie wystarczy delegować; trzeba wiedzieć, czego delegować nie wolno. Nie wystarczy sprawdzić wyniku; trzeba wiedzieć, jaki dowód jest wystarczający. Nie wystarczy mieć racji technicznie; trzeba rozumieć, komu i czemu służy poprawna decyzja. Cassani nazywa ten ruch przejściem ku orkiestracji inteligencji. Można mu nadać szerszy sens: to przejście od pracownika wykonującego zadanie do opiekuna systemu zdolnego te zadania realizować. Wraz ze wzrostem władzy, jaką daje możliwość uruchomienia wielu agentów, powinna rosnąć odpowiedzialność. Dlatego przyszłości pracy nie należy oceniać jedynie liczbą uratowanych lub zlikwidowanych etatów. Kluczowe jest pytanie o jakość stanowisk, które pozostaną: czy człowiek stanie się biernym operatorem rekomendacji, czy suwerennym współprojektantem systemu? Czy odzyska czas na refleksję, czy otrzyma pięciokrotnie większy strumień zadań do zatwierdzenia?

Czy AI będzie nauczycielem, czy protezą uniemożliwiającą rozwój? I czy firma będzie inwestowała w przyszłych ekspertów, czy jedynie konsumowała kompetencje poprzedniego pokolenia?

Kompetencje ludzkie jako warunek bezpiecznej delegacji

Nie znamy jeszcze pełnej odpowiedzi empirycznej. Adopcja technologii jest zbyt świeża, narzędzia zmieniają się zbyt szybko, a najnowsza literatura na temat deskillingu łączy badania empiryczne z modelami koncepcyjnymi, których hipotezy dopiero czekają na weryfikację. Nauka powinna zatem uczciwie przyznać: wiemy już wystarczająco dużo, by dostrzec ryzyko, ale zbyt mało, by ogłosić nieuchronność któregokolwiek ze scenariuszy. I być może właśnie to jest najważniejsza wiadomość. Przyszłość pracy z AI nie jest prawem natury, które należy biernie przewidzieć; jest ona w dużej mierze problemem projektowym. Zależy od bodźców ekonomicznych, prawa, architektury narzędzi, sposobu mierzenia produktywności, modeli szkolenia, kultury organizacji oraz tego, jaką część odpowiedzialności społeczeństwo zechce rzeczywiście pozostawić ludziom. Technologia wyznacza przestrzeń możliwości, natomiast instytucje decydują, które z nich staną się codziennością.

Dlatego największym błędem byłoby przyjęcie, że skoro maszyna potrafi wykonać czynność poznawczą, człowiek powinien przestać się jej uczyć. Możliwość delegowania nie jest dowodem zbędności kompetencji – wręcz przeciwnie, to właśnie biegłość osoby delegującej sprawia, że proces ten staje się bezpieczny. Senior przyszłości może pisać mniej kodu niż senior przeszłości, a jednocześnie potrzebować głębszego rozumienia systemu. Lider może podejmować mniej mikrodecyzji, ponosząc odpowiedzialność za znacznie szerszy obszar.

Junior może szybciej osiągać rezultaty, lecz jego droga do prawdziwej ekspertyzy będzie wymagała bardziej świadomie zaprojektowanego treningu. Organizacja może zatrudniać mniej osób do czystego wykonawstwa, ale potrzebować więcej specjalistów zdolnych do integracji domen, kontroli znaczenia i uczenia innych. Kod tanieje. Osąd nie musi. Odpowiedź tanieje. Dobre pytanie nie musi. Produkcja wariantów tanieje. Odpowiedzialność za wybór pomiędzy nimi pozostaje kosztowna. Na tej różnicy można zbudować nie obronę człowieka przed technologią, lecz nową teorię jego wartości w technologii. Pozostaje już finałowe spięcie konstrukcji.

Trzeba powrócić do SWE-bench, SWE-agent i SWE-Gym; do interfejsu, treningu i ewaluacji; do RACM, PAIP, ECF i RAUC; do prawa, ekonomii, długu technicznego, zespołu oraz człowieka. Dopiero wtedy będzie można odpowiedzieć na pytanie, które towarzyszyło całej analizie: czy Agent AI jest kolejnym narzędziem programisty, czy początkiem nowej architektury ludzkiej sprawczości – i jaką doktrynę Dobra Wspólnego powinna przyjąć organizacja, która chce tę sprawczość powiększyć, nie oddając maszynie suwerenności nad sensem działania? Architektura sprawczości: od SWE-bench do Dobra Wspólnego. Doktryna człowieka, który nie abdykuje.

Warto powrócić do miejsca, od którego wszystko się zaczęło: nie do spektakularnych wizji autonomicznych zespołów maszyn, budżetów tokenowych czy przyszłych Mission Control Centers, lecz do prostego pytania badawczego o zdolność modelu językowego do rozwiązania rzeczywistego problemu programistycznego. SWE-bench postawił to pytanie znacznie bardziej rygorystycznie niż wcześniejsze testy generowania izolowanych fragmentów kodu. Zamiast funkcji oderwanej od środowiska, pojawiło się repozytorium, zgłoszenie użytkownika, istniejąca architektura, zależności i testy oraz konieczność wprowadzenia zmiany, która nie tylko wygląda rozsądnie, lecz rzeczywiście naprawia system.

Sprawczość agentowa jako funkcja architektury i interfejsu

Pierwotny benchmark obejmował 2294 problemy z 12 repozytoriów Pythona. Najlepszy z modeli badanych w pierwszej publikacji, Claude 2, rozwiązywał zaledwie 1,96 procent zadań. Ten niski wynik nie był jednak kompromitacją sztucznej inteligencji, lecz ujawnieniem zasadniczej różnicy między generowaniem kodu a rzeczywistą inżynierią oprogramowania. Poprawna składnia to bowiem tylko fragment procesu; prawdziwe wyzwanie wymaga lokalizacji błędu, odtworzenia intencji autora, rozpoznania zależności między plikami, wyboru strategii naprawy, modyfikacji systemu, uruchomienia testów i interpretacji rezultatów. SWE-bench okazał się zatem czymś więcej niż rankingiem modeli: stał się laboratorium przejścia od odpowiedzi ku sprawczości.

Kolejny krok wykonał SWE-agent, dowodząc, że praktyczna zdolność systemu nie zależy wyłącznie od „inteligencji” samego modelu, ale również od architektury jego kontaktu ze środowiskiem. Autorzy zaprojektowali Agent-Computer Interface, który ułatwił modelowi przeszukiwanie repozytorium, edycję plików oraz korzystanie z narzędzi niezbędnych do realizacji zadania. Dzięki tej architekturze SWE-agent osiągnął około 12,5 procent skuteczności na SWE-bench oraz 87,7 procent na HumanEvalFix.

Znaczenie tego wyniku wykracza daleko poza sam benchmark, ponieważ wskazuje, że inteligencja systemu jest funkcją nie tylko "mózgu", lecz także ręki, oka, pulpitu sterowniczego i reguł interakcji. Człowiek zna tę prawdę od tysiącleci: chirurg potrzebuje instrumentarium, pilot kokpitu, naukowiec laboratorium, sędzia procesu dowodowego, a urzędnik procedury i wiarygodnych rejestrów. Inteligencja oderwana od środowiska pozostaje jedynie możliwością; dopiero właściwy interfejs zamienia tę możliwość w działanie. Z tego doświadczenia wyłania się jedna z centralnych tez artykułu: praktyczna inteligencja agentowa nie istnieje poza architekturą działania.

SWE-Gym przesunął następnie ciężar zainteresowania z samego interfejsu na uczenie agentów działania w środowisku przypominającym rzeczywistą pracę inżynieryjną. Zbiór obejmował 2438 zadań z realnych repozytoriów, wykorzystanych do trenowania agentów oraz systemów weryfikacyjnych; w opisanym układzie osiągnięto 32 procent na SWE-bench Verified i 26 procent na SWE-bench Lite.

Istotniejsza od samej wartości wyniku jest jego struktura epistemologiczna: kompetencja systemu nie musi być rozumiana wyłącznie jako cecha zapisana w parametrach modelu, lecz może być rozwijana poprzez doświadczenie, środowisko, informację zwrotną oraz sprawdzanie konsekwencji działania. Z połączenia SWE-agent, SWE-Gym i SWE-bench wyłania się triada, którą można uznać za naukowy kręgosłup transformacji agentowej: interfejs, trening i ewaluacja. Interfejs określa sposób działania systemu; trening kształtuje zdolność posługiwania się tą możliwością, a ewaluacja weryfikuje, czy deklarowana sprawczość rzeczywiście prowadzi do oczekiwanych rezultatów. Ta triada wymaga jednak czwartego elementu: krytyki samego pomiaru.

Adopcja AI jako problem ustrojowy organizacji

SWE-Bench+ wykazał, że benchmarki również mogą być obciążone błędami. Część sukcesów układu SWE-Agent+GPT-4 mogła wynikać z obecności rozwiązań w treści zgłoszeń, a niektóre poprawki przechodziły przez testy zbyt słabe, by wiarygodnie potwierdzić usunięcie problemu. Po odfiltrowaniu takich przypadków raportowany wynik tej konfiguracji spadł z 12,47 do 3,97 procent.

Nie oznacza to jednak utraty wartości SWE-bench, lecz dowodzi, że miara inteligencji sama musi podlegać inteligentnej kontroli. Nauka rozwija się właśnie dzięki temu, że nie ufa bezkrytycznie własnym triumfom: każdy test z czasem zostaje poznany, każda metryka zoptymalizowana, a każdy benchmark nasycony danymi treningowymi modeli. Prawo Goodharta staje się tutaj ostrzeżeniem epistemologicznym – gdy miernik zaczyna pełnić funkcję celu, system może nauczyć się zdobywać wynik skuteczniej niż kompetencję, którą ten wynik miał reprezentować. Ten sam problem przenosi się z laboratorium benchmarków do przedsiębiorstwa.

Velocity przestaje mierzyć wartość, jeśli zarządzanie zostaje mu całkowicie podporządkowane; liczba linii kodu może stać się wskaźnikiem odwrotnym do elegancji systemu; liczba zamkniętych ticketów może rosnąć szybciej niż użyteczność produktu, a procent kodu generowanego przez AI może sprawiać wrażenie miary nowoczesności, podczas gdy w rzeczywistości dokumentuje jedynie rosnące uzależnienie organizacji od zewnętrznej automatyzacji. Najważniejszym efektem analizy nie jest zatem katalog modeli i narzędzi, lecz zmiana jednostki analizy. Obserwowaliśmy model, następnie agenta, relację człowieka z agentem i cały zespół ludzi oraz maszyn, aż wreszcie

doszliśmy do organizacji posiadającej własne prawo, pamięć, budżet, kulturę, kompetencje, infrastrukturę i system odpowiedzialności. Dopiero na tym poziomie ujawnia się istota transformacji: adopcja AI staje się nie tylko problemem technologicznym, lecz problemem ustrojowym. Cztery ramy Cassaniego można odczytać jako próbę zbudowania fragmentów takiego ustroju.

RACM służy poznaniu faktycznej zdolności narzędzi i granic ich niezawodności; PAIP kształtuje reżim delegowania – od eksperymentalnego Vibe Coding, przez Agent-Supervised Development, po ścisłe AI-Assisted Coding w obszarach wymagających maksymalnej kontroli człowieka; ECF wraz z IDVI organizuje wspólne dochodzenie do rezultatu poprzez Intent, Dialogue, Validation i Integration; RAUC natomiast przenosi odpowiedzialność w praktykę codziennego działania, wymuszając pytania o cel, dane, bezpieczeństwo i konsekwencje przed użyciem AI, w jego trakcie oraz po zakończeniu procesu.

Źródło przedstawia te narzędzia jako komplementarne filary orkiestracji adopcji. Nie są one prawami natury ani konstytucją organizacji, lecz ich wspólna intuicja zasługuje na zachowanie: sprawczość potrzebuje formy, gdyż sama zdolność wykonania nie odpowiada na pytania o dopuszczalność działania, zakres autonomii, skuteczność kontroli i odpowiedzialność za rezultat. Właśnie dlatego Execution Plan, Logbook, checkpoint, HITL, Andon Cord, sandbox, evals i least privilege należy interpretować nie jako luźny zestaw praktyk technicznych, lecz jako elementy instytucjonalnej architektury sprawczości. Execution Plan przypomina zdefiniowanie mandatu; Logbook tworzy pamięć działania; checkpoint wyznacza granicę delegacji; HITL umożliwia realną interwencję; Andon Cord chroni prawo do natychmiastowego zatrzymania ryzykownego procesu; sandbox zakreśla terytorium, na którym agent może eksperymentować bez nieograniczonego wpływu na świat; evals pełnią funkcję eksperymentalnego sądu nad jakością rezultatów, a least privilege realizuje starą zasadę ustrojową, według której władza powinna być ograniczona do minimum niezbędnego do wykonania zadania.

Informatyka zaczyna posługiwać się logiką znaną filozofii politycznej: tam, gdzie powstaje władza, potrzebne są granice; gdzie ustanawia się granice, potrzebne są reguły; gdzie istnieją reguły, musi pojawić się odpowiedzialność, a ta wymaga zdolności rekonstrukcji tego, kto, dlaczego i w jaki sposób działał.

Klasyczne pytanie quis custodiet ipsos custodes? – kto będzie pilnował strażników – zyskuje w systemach agentowych nowe znaczenie. Nie wystarczy pytać, kto kontroluje człowieka; trzeba pytać, kto konfiguruje agenta, ustala jego uprawnienia, kontroluje walidatora, wybiera benchmark, ocenia jego jakość i posiada prawo stwierdzenia, że miernik sukcesu przestał mierzyć rzecz rzeczywiście wartą osiągania. Odpowiedzią nie może być nieskończona regresja kolejnych agentów

nadzorujących agentów. Drugi agent może zwiększyć szansę wykrycia błędu pierwszego, podobnie jak drugi recenzent dostrzega pomyłkę pierwszego człowieka, lecz nie usuwa to potrzeby ostatecznej odpowiedzialności instytucjonalnej.

Doktryna poszerzonej sprawczości jako granica automatyzacji

W pewnym punkcie musi istnieć człowiek lub organ zdolny stwierdzić, że organizacja nie akceptuje danego ryzyka, określonego działania nie automatyzuje, pewnych danych nie wykorzystuje, a decyzji statystycznie korzystnej nie podejmie, ponieważ pozostaje ona sprzeczna z jej zobowiązaniami wobec ludzi. Ten poziom odpowiedzialności można nazwać suwerennością sensu.

Maszynie można powierzyć wykonanie, poszukiwanie wariantów, przeszukiwanie repozytoriów, generowanie testów, analizowanie ogromnych przestrzeni możliwości czy realizację sekwencji operacji. Nie należy jednak niepostrzeżenie oddawać jej prawa do definiowania tego, po co organizacja istnieje, komu służy i jakie środki uważa za dopuszczalne. Ta granica stanowi fundament Dobra Wspólnego w przedsiębiorstwie agentowym. Dobro Wspólne nie wymaga bowiem technologicznego konserwatyzmu ani sztucznego podtrzymywania wszystkich dawnych stanowisk i procedur. Oznacza raczej taki ustrój technologiczny, w którym zwiększanie mocy działania całej wspólnoty nie niszczy zdolności jej członków do rozumienia, współdecydowania, uczenia się i ponoszenia odpowiedzialności za wspólną przyszłość. Można tę zasadę określić mianem doktryny poszerzonej sprawczości.

Jeśli AI ogranicza czas lekarza przeznaczany na dokumentację i pozwala poświęcić więcej uwagi pacjentowi – zwiększa sprawczość; jeśli zdejmuje z programisty konieczność produkowania rutynowego boilerplate'u, pozostawiając więcej czasu na architekturę, walidację oraz zrozumienie problemu – zwiększa sprawczość; jeśli umożliwia obywatelowi rozpoznanie znaczenia skomplikowanego aktu prawnego i przygotowanie własnego argumentu – poszerza jego udział w życiu wspólnoty. Jeżeli natomiast system przyspiesza człowieka tak bardzo, że odbiera mu przestrzeń na refleksję, eliminuje drogę juniora do ekspertyzy, przekształca seniora w ceremonialnego akceptanta, a osobę poddaną decyzji w rekord przekazywany przez pipeline, wówczas zwiększyła się moc wykonawcza systemu, ale niekoniecznie zwiększyło się dobro człowieka. Technologia może być wtedy potężniejsza, podczas gdy osoba znajdująca się w jej otoczeniu staje się mniej suwerenna. Z tej perspektywy maksyma Cassaniego "Automate to Augment, Don't Replace" wymaga poszerzenia. Autor rozumie ją jako delegowanie rutynowych czynności do maszyn przy zachowaniu kontroli człowieka nad architekturą, strategią produktu i

interesariuszami. Można nadać tej zasadzie głębszą treść: automatyzować należy to, co zwiększa ludzką zdolność rozumienia, tworzenia i odpowiedzialnego działania, a nie to, czego automatyzacja odbiera ludziom kompetencję potrzebną do powiedzenia maszynie "nie".

Phronesis i odpowiedzialność jako fundamenty dojrzałości technologicznej

Nie jest to sentymentalna obrona człowieka, lecz zasada odporności systemowej. Organizacja potrzebuje ludzi nie dlatego, że zawsze mają rację, ale dlatego, że system odpowiedzialności wymaga podmiotu zdolnego zmienić regułę w momencie, gdy prowadzi ona do niewłaściwego rezultatu. Agent może realizować funkcję celu, lecz człowiek musi zachować zdolność zakwestionowania samej funkcji; agent korzysta z dostarczonego kontekstu, ale to człowiek potrafi odkryć, że decydujący fragment tego kontekstu nigdy nie został zapisany; agent optymalizuje proces, jednak człowiek musi mieć prawo zapytać, czy dana rzecz w ogóle powinna podlegać optymalizacji. W tym punkcie nowoczesna inżynieria spotyka się z arystotelesowskim rozróżnieniem techne i phronesis. Techne to umiejętność wytwarzania, natomiast phronesis to mądrość praktyczna, pozwalająca rozpoznać właściwy sposób działania w konkretnych okolicznościach.

Agentowa AI osiąga coraz wyższą sprawność w pierwszym obszarze, co relatywnie zwiększa wartość drugiego. Najwyższym etapem rozwoju programisty nie musi być już umiejętność napisania najbardziej skomplikowanego fragmentu oprogramowania, lecz zdolność rozpoznania, że taki kod w ogóle nie powinien powstać. Najlepszy architekt nie ten, który uruchamia największą liczbę agentów, ale ten, który potrafi wybrać minimalny poziom maszynowej sprawczości wystarczający do realizacji celu. Najlepszy menedżer nie maksymalizuje wskaźnika adopcji AI, lecz dba o to, by zespół osiągał rezultaty bez utraty wiedzy, bezpieczeństwa, odpowiedzialności i profesjonalnej godności. Dlatego samo pojęcie „AI-first” wymaga intelektualnego uporządkowania. Dojrzała organizacja powinna być przede wszystkim purpose-first, gdyż technologia musi służyć jasno określonemu celowi; powinna być human-value-first, ponieważ wynik ekonomiczny i techniczny funkcjonuje wewnątrz wspólnoty ludzi, których dotyka; następnie evidence-first, by decyzje o autonomii wynikały z dowodów, ewaluacji i doświadczenia. Dopiero na końcu organizacja powinna wybierać konkretną technologię.

Taka hierarchia nie osłabia innowacyjności, lecz zabezpiecza ją przed przemianą w kult narzędzia. Historia zarządzania zna wiele idei, które zaczynały jako odpowiedź na realny problem, a kończyły jako pusty rytuał: Agile miał przywracać zdolność adaptacji, lecz bywał sprowadzany do ceremonii

sprintowych; KPI miały pomagać rozumieć organizację, a stawały się systemem produkowania pożądanych liczb; Big Data miały usprawniać decyzje, lecz często generowały jedynie kolejne dashboardy pozbawione znaczenia. AI może podzielić ten los, jeśli przedsiębiorstwa będą pytać wyłącznie o to, gdzie jeszcze można ją zastosować, zamiast zastanawiać się, którego problemu wcześniej nie umiano rozsądnie rozwiązać. Dojrzałość technologiczna obejmuje więc również zdolność świadomej rezygnacji z użycia dostępnej mocy. W kulturze nieustannej innowacji decyzja „nie wykorzystamy tutaj AI" może sprawiać wrażenie zacofania, choć w rzeczywistości zdolność nieużywania wszystkich dostępnych możliwości od stuleci stanowi jedno ze znamion cywilizacyjnej dojrzałości. Cywilizacja nie jest bowiem historią eksploatacji każdej możliwej techniki, lecz także historią zakazów, hamulców i norm, dzięki którym moc przestaje być synonimem prawa do jej wykorzystania. Postęp technologiczny nie powinien być mierzony jedynie tym, co potrafimy zrobić, ale również tym, czy umiemy rozpoznać sytuacje, w których nie powinniśmy robić tego, co już potrafimy.

Inspirującą metaforę dla takiej kultury odnajdujemy w judaistycznej tradycji intelektualnej, gdzie spór nie jest defektem poznania, lecz jego narzędziem. Tradycja talmudyczna zachowywała argumenty odrzucone obok tych przyjętych, wierząc, że okoliczności przyszłych pokoleń mogą przywrócić znaczenie głosowi mniejszościowemu. Organizacja agentowa powinna przyswoić podobną wrażliwość: nie wystarczy przechowywać odpowiedzi i wykonanych trajektorii; warto zachowywać także wątpliwości, alternatywy oraz racje, dla których pewnych rozwiązań nie wybrano. Logbook nie może być wyłącznie pamiętnikiem działań agenta; dojrzała pamięć instytucjonalna powinna utrwalać momenty, w których człowiek zatrzymał automatyczną trajektorię. To właśnie w uzasadnieniu odmowy może kryć się wiedza ważniejsza dla przyszłości niż w tysiącu bezproblemowo wykonanych operacji. W tym duchu można sparafrazować maksymę o uczeniu się od każdego: w epoce AI krąg potencjalnych nauczycieli rozszerza się o modele, agentów, ich sukcesy, błędy i nieudane eksperymenty. Warunkiem pozostaje jednak zdolność uczenia się, a nie jedynie konsumowania kolejnych odpowiedzi.

Podobnie słynną myśl Hillela – „jeśli nie ja dla siebie, to kto; jeśli tylko dla siebie, to czym jestem?" – można odczytać na nowo: jeśli człowiek nie zachowa odpowiedzialności za własną sprawczość, trudno oczekiwać, że ochroni ją za niego narzędzie. Jeśli zaś zwiększoną przez technologię sprawczość wykorzysta jedynie do maksymalizacji własnej przewagi kosztem innych, utraci wymiar wspólnotowy, który nadaje ludzkiej sprawczości sens.

Prymat ludzkiego osądu nad sprawczością maszyn

Właśnie tutaj sztuczna inteligencja spotyka kategorię Dobra Wspólnego. Nie chodzi wyłącznie o produktywność, lecz o produktywność tego, co warto wytwarzać; nie tylko o efektywność, ale o efektywność podporządkowaną właściwemu celowi; nie tylko o autonomię agentów, ale o autonomię proporcjonalną do odpowiedzialności; nie tylko o inteligencję systemu, lecz o inteligencję podporządkowaną mądrości instytucji. Z tego punktu widzenia najważniejszym artefaktem epoki agentowej nie będzie pojedynczy prompt, model ani agent, lecz dobrze zaprojektowana relacja pomiędzy możliwością działania a odpowiedzialnością za konsekwencje tego działania. Tę relację można opisać zestawem maksym, które tworzą jedną spójną doktrynę.

Nie należy delegować tego, czego organizacja nie potrafi jasno nazwać i uzasadnić, ani zwiększać autonomii szybciej niż rośnie zdolność kontroli. Nie wolno mylić inteligencji modelu z dojrzałością instytucji, prędkości z właściwym kierunkiem, a ilości kodu z wartością produktu. Najtańszy commit może generować najdroższe odsetki, test może potwierdzić jedynie to, czego nauczyliśmy go sprawdzać, a człowiek umieszczony formalnie w pętli nie sprawuje realnego nadzoru, jeśli nie ma czasu, kompetencji albo prawa do odmowy. Nie należy również automatyzować procesu uczenia się wyłącznie dlatego, że można zautomatyzować wykonanie; dzisiejszy junior może być bowiem jutro jedynym człowiekiem zdolnym rozpoznać klasę błędu systemu, którego jeszcze nie zbudowaliśmy. Pamięć maszyny nie powinna zastępować pamięci organizacji, a agent nie może stać się miejscem znikania odpowiedzialności.

Z tego wszystkiego wynika nadrzędna zasada: zamiast budować przedsiębiorstwo AI-first, trzeba budować przedsiębiorstwo good-judgment-first. Następna generacja technologii ponownie zmieni dostępne narzędzia, podczas gdy zdolność rozsądnego osądu pozostanie konieczna również do oceny jej następcy. W tym świetle cała droga od SWE-bench po AI governance staje się jedną logiczną narracją. SWE-bench zapytał, czy system potrafi naprawić realny problem; SWE-agent ujawnił znaczenie interfejsu; SWE-Gym znaczenie uczenia i doświadczenia; krytyka benchmarków przypomniała o potrzebie falsyfikowania własnych miar; Cassani dołożył warstwę organizacyjną w postaci RACM, PAIP, ECF oraz RAUC; governance wyznaczył granice autonomii, ekonomia nakazała policzyć koszt całego życia rozwiązania, prawo przypisało obowiązki ludziom i instytucjom, a psychologia pracy odsłoniła ryzyko utraty kompetencji potrzebnych do kontrolowania coraz sprawniejszych systemów.

Z tych wątków wyłania się wspólny wniosek: AI przestaje być jedynie narzędziem do produkowania odpowiedzi i staje się architekturą delegowania sprawczości, a każda architektura sprawczości jest zarazem architekturą władzy. Dlatego ostateczne pytania transformacji nie brzmią technicznie.

Trzeba ustalić, kto definiuje cel, kto udostępnia dane, kto przyznaje agentowi uprawnienia, kto może zatrzymać jego działanie, kto ponosi konsekwencje i kto zachowuje kompetencję pozwalającą powiedzieć, że rezultat poprawny technicznie jest niedopuszczalny organizacyjnie, prawnie albo moralnie. W odpowiedziach na te pytania objawi się prawdziwa dojrzałość przedsiębiorstwa, ponieważ sprawczość bez odpowiedzialności jest tylko mocą, a moc pozbawiona granic wcześniej czy później przestaje być narzędziem i zaczyna dyktować warunki swojemu użytkownikowi.

Materiał Cassaniego przedstawia programistę przyszłości jako orkiestratora i podkreśla, że ludzka mądrość architektoniczna, krytyczny osąd, empatia i rzeczywisty HITL powinny pozostać bezpiecznikiem systemów produkcyjnych. Powodzenie transformacji zależeć będzie od harmonizowania inteligencji ludzkiej i maszynowej.

Metaforę dyrygenta warto potraktować poważnie. Dyrygent nie wydobywa osobiście każdego dźwięku, lecz odpowiada za proporcję, tempo, napięcie, wejście i ciszę. Nie jest właścicielem muzyki, ale ma obowiązek sprawić, aby różne głosy stworzyły znaczącą całość. Tak samo człowiek epoki agentowej nie musi własnoręcznie wykonywać każdego ruchu systemu, lecz musi wiedzieć, jaka "muzyka" ma powstać, umieć rozpoznać fałsz, zatrzymać orkiestrę, gdy następny dźwięk staje się jedynie hałasem, oraz pamiętać, dla kogo cała orkiestra gra.

W tym ostatnim pytaniu mieści się Dobro Wspólne. Nie budujemy inteligentniejszych systemów po to, aby same systemy posiadały coraz więcej inteligencji, lecz po to, aby ludzie i tworzone przez nich wspólnoty miały większą zdolność rozumienia rzeczywistości, podejmowania trafnych decyzji i odpowiedzialnego kształtowania przyszłości. Jeżeli Agent AI zwiększa tę zdolność, należy do najbardziej fascynujących technologicznych możliwości naszej epoki. Jeżeli natomiast wraz ze wzrostem jego sprawczości maleje nasza zdolność rozumienia, uczenia się, sprzeciwu i brania odpowiedzialności, powinniśmy mieć odwagę nazwać taki proces nie postępem, lecz abdykacją. Najważniejsza granica przyszłości nie przebiega zatem pomiędzy człowiekiem i maszyną, lecz wewnątrz samego człowieka – pomiędzy pokusą oddania odpowiedzialności a wolą zachowania suwerenności nad własnym osądem. Technologia może podpowiadać, planować, pisać, testować, wykonywać działania i eksplorować przestrzeń możliwości w skali przekraczającej ludzkie możliwości poznawcze, lecz pytanie "po co?" nadal powinno mieć adresata zdolnego odpowiedzieć nie jedynie statystyczną optymalizacją, ale wartością, odpowiedzialnością i wizją świata, który chce współtworzyć. Właśnie w tej zdolności do określenia sensu działania pozostaje ostateczne miejsce człowieka – nie jako konkurenta maszyny przy klawiaturze, lecz jako podmiotu odpowiedzialnego za to, czemu służy inteligencja, którą nauczył się uruchamiać.

> Ostateczna granica przyszłości nie przebiega między człowiekiem a maszyną, lecz wewnątrz nas samych – między pokusą oddania odpowiedzialności a wolą zachowania suwerenności nad własnym osądem. Czy w świecie, w którym AI potrafi zaplanować i wykonać niemal wszystko, będziemy mieli odwagę pozostać podmiotami definiującymi sens działania, czy staniemy się jedynie ceremonialnymi akceptantami wyników statystycznej optymalizacji? Pytanie o to, 'po co' coś tworzymy, pozostaje jedynym bastionem, którego nie da się zautomatyzować bez utraty naszego człowieczeństwa.

Inspiracje

[1] **"Code Revealed"** Alexio Cassani

2025 | Packt Publishing | ISBN: 978-1-80778-930-5

Alexio Cassani to włoski przedsiębiorca, badacz i lider technologiczny z ponad dwudziestoletnim doświadczeniem w inżynierii oprogramowania i transformacji cyfrowej. Absolwent inżynierii komputerowej na Politechnice Mediolańskiej (2003), pełnił funkcję dyrektora ds. technologii w takich firmach jak Cortilia i Webscience. Cassani jest współzałożycielem FairMind, startupu koncentrującego się na zwiększaniu produktywności rozwoju oprogramowania za pomocą agentów AI. Jego praca zawodowa koncentruje się na praktycznym zastosowaniu generatywnej sztucznej inteligencji, agentów AI i ustrukturyzowanych przepływów pracy w środowiskach korporacyjnych. Jest edukatorem i autorem, który łączy badania naukowe z praktycznym dostarczaniem oprogramowania, opowiadając się za odpowiedzialnym wdrażaniem sztucznej inteligencji poprzez takie frameworki jak RACM (Responsible AI Usage Checklist) i PAIP (Practical AI Integration Patterns). W swojej pracy podkreśla znaczenie zarządzania, nadzoru i współpracy człowiek-maszyna we współczesnym rozwoju oprogramowania.

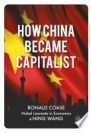
***Alexio Cassani** is an Italian entrepreneur, researcher, and technology leader with over twenty years of experience in software engineering and digital transformation. A graduate in Computer Engineering from Politecnico di Milano (2003), he has held roles as a CTO at companies such as Cortilia and Webscience. Cassani is the co-founder of FairMind, a startup focused on enhancing software development productivity through AI agents. His professional work centers on the practical application of generative AI, AI agents, and structured workflows in enterprise environments. He is an educator and author who bridges the gap between academic research and real-world software delivery, advocating for responsible AI adoption through frameworks like RACM (Responsible AI Usage Checklist) and PAIP (Practical AI Integration Patterns). His work emphasizes the importance of governance, supervision, and human-machine collaboration in modern software development.*

FairMind

Literatura uzupełniająca

Książki

Literatura pokrewna (Google Scholar):



[1] "How China Became Capitalist"

Ronald Coase

Springer | ISBN: 9781137019370

[2] "Markets and Hierarchies : Analysis and Antitrust Implications,"

Oliver Williamson

[3] "bounded rationality"

Oliver Williamson

[4] "Theory of transaction costs"

Oliver Williamson

[5] "The rational and social Foundations of music"

Max Weber

ISBN: 9780758136602

[6] "The Imperative of Responsibility"

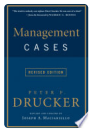
Hans Jonas

1984 | University of Chicago Press | ISBN: 9780226405964

[7] "Managing For Results"

Peter Drucker

2012 | Routledge | ISBN: 9781136009693



[8] "Management Cases, Revised Edition"

Peter F. Drucker

2009 | Harper Collins | ISBN: 9780061853760

[9] "Drug Discovery and Development, Third Edition"

James J. O'Donnell, John Somberg, Vincent Idemyor, James T. O'Donnell

2019 | CRC Press | ISBN: 9781351625135

Artykuły naukowe

Autorzy przywoływani:

[1] "FROM ‘BUREAUCRACY’"

Max Weber

2005 | Social Theory | DOI: 10.1515/9781474469654-021

Google Scholar

[2] "Expanding the Role of Tools in a Literate Programming Environment"

Kent Beck, Ward Cunningham

2000

Google Scholar

[3] "Agile management - an oxymoron?: who needs managers anyway?"

L. Anderson, Glen B. Alleman, Kent L. Beck, Joseph A. Blotner, Ward Cunningham

2003 | Conference on Object-Oriented Programming Systems, Languages, and Applications | DOI: 10.1145/949344.949410

Google Scholar

[4] "The Problem of Social Cost"

Ronald H. Coase

1960 | Palgrave Macmillan UK eBooks | DOI: 10.1002/9780470752135.ch1

Google Scholar

[5] "The New Institutional Economics"

Ronald H. Coase

2002 | Cambridge University Press eBooks | DOI: 10.1017/cbo9780511613807.002

Google Scholar

[6] "The Institutional Structure of Production"

Ronald H. Coase

2008 | DOI: 10.1007/978-3-540-69305-5_3

Google Scholar

[7] "Transaction-Cost Economics: The Governance of Contractual Relations"

Oliver E. Williamson

1979 | The Journal of Law and Economics | DOI: 10.1086/466942

Google Scholar

[8] "Comparative Economic Organization: The Analysis of Discrete Structural Alternatives"

Oliver E. Williamson

1991 | Administrative Science Quarterly | DOI: 10.2307/2393356

Google Scholar

[9] "The Economics of Organization: The Transaction Cost Approach"

Oliver E. Williamson

1981 | American Journal of Sociology | DOI: 10.1086/227496

Google Scholar

[10] "The New Institutional Economics: Taking Stock, Looking Ahead"

Oliver E. Williamson

2000 | Journal of Economic Literature | DOI: 10.1257/jel.38.3.595

Google Scholar

[11] "FROM 'POLITICS AS A VOCATION'"

Max Weber

1991 | Edinburgh University Press eBooks | DOI: 10.1515/9781474469654-020

Google Scholar

[12] "The Imperative of Responsibility: In Search of an Ethics in the Technological Age"

M. Granger Morgan, Hans Jonas

1985 | Journal of Policy Analysis and Management | DOI: 10.2307/3324683

Google Scholar

[13] "The Imperative of Responsibility: In Search of an Ethics for the Technological Age"

Hans Jonas

1985 | DigitalGeorgetown (Georgetown University Library)

Google Scholar

[14] "Philosophical Essays: From Ancient Creed to Technological Man."

Daniel S. Robinson, Hans Jonas

1974 | Philosophy and Phenomenological Research | DOI: 10.2307/2106643

Google Scholar

[15] **"Philosophical Reflections on Experimenting with Human Subjects"**

Hans Jonas

1979 | DOI: 10.1007/978-1-4615-6561-1_16

Google Scholar

[16] **"Vision Transformers and Bi-LSTM for Alzheimer's Disease Diagnosis from 3D MRI"**

Taymaz Akan, Sait Alp, Mohammad Alfrad Nobel Bhuiyan

2023 | DOI: 10.1109/csce60160.2023.00093

Google Scholar

[17] **"The Hidden Injuries of Class"**

Richard Sennett, Jonathan Cobb

1972 | London School of Economics and Political Science Research Online (London School of Economics and Political Science)

Google Scholar

[18] **"Building and Dwelling"**

R. Sennett

2023 | DOI: 10.2307/jj.5666735

Google Scholar

Artykuły pokrewne (Google Scholar):

[19] **"zIA: a GenAI-powered local auntie assists tourists in Italy"**

Alexio Cassani, Michele Ruberl, Antonio Salis, Giacomo Giannese, Gianluca Boanelli

2024 | arXiv (Cornell University) | DOI: 10.48550/arxiv.2407.11830

Google Scholar

Treść tworzy Fundacja Dobre Państwo.
Redaguje i publikuje APA ONE, autorski system redakcyjny Fundacji oparty na AI.
APA ONE Publishing System
Fundacja Dobre Państwo © 2026
Article ID: a53ce80a-8d08-4804-bf69-1d908154a2c1