



PARTNER
RAPORTU

Low Code

ZASTOSOWANIA | DOBRE PRAKTYKI | WYZWANIA | FAKTY I MITY

Podjęcie niskokodowe jest sposobem na **przyspieszenie** i **zmniejszenie kosztów** tworzenia aplikacji biznesowych i integracji, a także **modernizację** istniejących systemów. Sprawdź, jakie możliwości daje Low Code.

Poznaj **zastosowania**, ryzyka, **dobre praktyki** i sprawdzone **rozwiązania**.

iTWIZ.



Spis treści

str. 3 – 7

Low Code vs No Code
wojna dwóch światów

Przeczytaj →



str. 32 – 42

**No Code, Low Code,
Pro Code. Co wybrać?**
Accenture

Przeczytaj →



str. 8 – 15

Wywiad
z **Manoj Nair**
C&F

Przeczytaj →



str. 43 – 48

Jaki powinien być Low Code
do zastosowań korporacyjnych

Przeczytaj →



str. 16 – 19

Dlaczego Low Code
wkracza na salony?

Przeczytaj →



str. 49 – 56

Przykładowe
zastosowania Low Code
– zestawienie multicaso

Przeczytaj →



str. 20 – 26

Wywiad
z **Filipem Kobielskim**
EY Polska

Przeczytaj →



str. 58 – 60

Jakie rozwiązania warto
budować za pomocą platform
No Code, a jakich lepiej nie?

Przeczytaj →



str. 27 – 29

Kim jest Citizen Developer
i kto może nim zostać?

Przeczytaj →



str. 61 – 62

Trzy fazy rozwoju
oprogramowania w myśl
konceptu No Code

Przeczytaj →



str. 30 – 31

6 kroków w kierunku
wdrożenia koncepcji
Citizen Developer

Przeczytaj →



str. 63 – 68

13 faktów i mitów
o platformach Low Code

Przeczytaj →





Przeczytaj inspirujący wywiad z **Manoj Nair**

Head of Digital Transformation w C&F



Z wywiadu dowiesz się:

W jaki sposób koncepcja Low Code wspiera innowacje biznesowe, a także – jak zmienia proporcje wydatkowania budżetów na rozwój aplikacji

Kiedy warto sięgnąć po rozwiązania niskokodowe, a kiedy taki projekt nie znajduje uzasadnienia

Jakich wzrostów efektywności pracy deweloperów można oczekiwać od rozwiązań Low Code

Co sprawia, że popularność podejścia Low Code tak dynamicznie rośnie, także w silnie regulowanych sektorach gospodarki

Jak Low Code zmienia sposób pracy programistów, a także relacje pomiędzy zespołem IT i użytkownikami biznesowymi

[przejdź do wywiadu](#)





Low Code vs No Code

Wojna dwóch światów

Koncepcja niskokodowego lub bezkodowego tworzenia rozwiązań biznesowych nie jest nowa. **Pomimo wielu podobieństw, Low Code i No Code stanowią zupełnie odmienne podejścia, błędnie sprowadzane do jednej kategorii.**

Jakie zatem są różnice między Low Code i No Code.

W ślad za rosnącą złożonością oraz dynamiką procesów w większych branżach postępuje dziś skala wykorzystania rozwiązań automatyzujących wybrane operacje biznesowe. Rosną także potrzeby w zakresie modernizacji środowisk IT, która pozwoli dostosować je do obecnych realiów biznesowych. W wielu organizacjach tego typu potrzeby biznesowe względem warstwy oprogramowania zderzają się z barierą w postaci niedoboru specjalistów zajmujących się tworzeniem aplikacji.

Powodów tego stanu rzeczy jest dużo – jednak większość z nich sprowadza się do kwestii finansowych: w wielu firmach nie ma zgody albo środków na zwiększenie liczebności zespołów deweloperskich. W efekcie, dużo projektów IT jest odkładanych, a backlog tych niezrealizowanych puchnie w oczach. To zaś wprost przekłada się na wzmocnienie nieefektywności operacyjnej, którą przecież nowe rozwiązania mają eliminować.

Realną alternatywą dla zwiększania zatrudnienia w zespołach zajmujących się rozwojem aplikacji stało się dziś zaangażowanie w takie procesy użytkowników biznesowych wyposażonych w odpowiednie narzędzia. Narzędziami takimi są platformy deweloperskie, określane mianem niskokodowych (Low Code) lub bezkodowych (No Code). W obecnych realiach stanowią one ciekawą alternatywę dla tradycyjnych (wysokokodowych) procesów tworzenia oprogramowania.

Pomimo pewnych podobieństw użytkowych, środowiska Low Code i No Code to dwie odmienne kategorie rozwiązań. Co zatem je różni?

Czym jest Low Code?

Koncepcja Low Code zakłada zautomatyzowanie i uproszczenie prac związanych z tworzeniem kodu aplikacyjnego. Zakłada przy tym m.in. wykorzystanie prostych w obsłudze narzędzi graficznych, pozwalających np. na tworzenie kodu bez konieczności ręcznego kodowania. W ten sposób definiowane mogą być m.in. przepływy procesów i danych, warstwa obsługi oraz reguły biznesowe lub integracyjne.

Co jednak ważne, poza wspomnianą automatyzacją platformy Low Code – środowiska RAD (Rapid Application Development) zapewniają użytkownikom dostęp do warstwy kodu aplikacji lub obsługę dodatkowego języka skryptowego, co pozwala programistom szybko zbudować ogólne ramy aplikacji w sposób graficzny, a następnie dopracować je, zadbać o specyficzne wymagania czy kwestie integracyjne już na poziomie modyfikacji i uzupełnienia wygenerowanego automatycznie kodu.

W analogiczny sposób możliwe jest również uzupełnianie kodu aplikacyjnego fragmentami trudnymi lub niemożliwymi do wygenerowania, poprzez gotowe bloki konstrukcyjne dostępne w danej platformie Low Code.

Co istotne, **podejście Low Code nie jest ograniczone jedynie do tworzenia typowych aplikacji biznesowych. Jest wykorzystywane m.in. w kontekście automatyzacji procesów biznesowych, tworzenia stron internetowych, rozwiązań mobilnych, a także integracji oprogramowania i usług chmurowych.** Znajduje zastosowanie również w kontekście modernizacji przestarzałych aplikacji, pozwalając przykładowo na uzupełnienie funkcjonalności istniejącego oprogramowania, dobudowanie nowego interfejsu graficznego czy niezbędnych integracji.



Co to jest No Code?

Koncepcja No Code wpisuje się w założenia idei Low Code, z tym wyjątkiem, że w pełni automatyzuje proces tworzenia kodu aplikacyjnego – i nie pozwala użytkownikom na ingerowanie w wygenerowany kod źródłowy. Na podstawie zdefiniowanych przy użyciu narzędzi graficznych schematów automatycznie tworzone są wszystkie przepływy, reguły i cały kod niezbędny do uruchomienia aplikacji.

Z jednej strony zatem platformy zbudowane w myśl koncepcji No Code zapewniają możliwość tworzenia oprogramowania w pełni za pomocą narzędzi graficznych, bez potrzeby – i możliwości – dokonywania jakichkolwiek zmian na poziomie kodu oraz bez znajomości składni czy zasad programowania. Z drugiej jednak, ograniczają możliwości budowanych w ten sposób aplikacji jedynie do funkcjonalności dostępnych za pośrednictwem predefiniowanych elementów dostępnych w danej platformie No Code.

Rozwiązania No Code znajdują zastosowanie w obszarach, takich jak: tworzenie prostych aplikacji biznesowych bezpośrednio przez użytkowników, budowanie interaktywnych raportów czy kokpitów menedżerskich, tworzenie prostych automatyzacji procesowych oraz budowanie interfejsów graficznych lub aplikacji mobilnych z gotowych komponentów.


Jakie są różnice między Low Code i No Code?

Zgodnie z powyższymi wyjaśnieniami, **główną różnicą** pomiędzy środowiskami niskokodowymi i bezkodowymi jest skala automatyzacji.

Platformy Low Code, za sprawą narzędzi wizualnych, narzędzi automatyzacyjnych i wielu gotowych komponentów, zmniejszają złożoność i pracochłonność procesów tworzenia aplikacji. Pozwalają jednak na tworzenie własnego kodu.

Platformy No Code z kolei bazują wyłącznie na warstwie wizualnej, nie zapewniając możliwości dodawania lub zmiany kodu źródłowego tworzonego oprogramowania, ani nawet wglądu w tenże kod.

To sprawia, że **rozwiązania Low Code oraz No Code są skierowane do dwóch różnych grup użytkowników**. Dla użytkowników biznesowych i osób nietechnicznych, często określanymi mianem Citizen Developerów, platformy No Code znoszą bariery związane z potrzebą ręcznego kodowania. Takie podejście ogranicza jednak zakres funkcjonalności możliwych do zaimplementowania w generowanych w pełni automatycznie aplikacjach. To zaś sprawia, że w bardziej złożonych projektach częściej wykorzystywane jest podejście Low Code.



Nie oznacza to jednak, to, że platformy Low Code są zarezerwowane wyłącznie dla osób z kompetencjami deweloperskimi. Dobrze sprawdzają się bowiem w zastosowaniach wymagających współpracy między użytkownikami biznesowymi i zespołem IT. Zespoły biznesowe, na bazie dostępnych w środowiskach Low Code narzędzi graficznych, mogą stosunkowo łatwo nakreślić oczekiwaną funkcjonalność lub interfejs aplikacji, gwarantując przy tym podstawę do dalszych prac przy zaangażowaniu programistów.

Co istotne, **rozwiązania Low Code i No Code w pewnym stopniu zmieniają zadania stojące przed zespołami IT – a także zakres odpowiedzialności zespołów biznesowych.** Żadna z tych koncepcji nie zakłada eliminacji roli zespołów IT w organizacjach. Nadal niezbędne jest bowiem m.in. zapewnienie kontroli i utrzymanie dla środowisk aplikacyjnych tworzonych na podstawie podejścia nisko- lub bezkodowego.

Podsumowując

Koncepcja Low Code zakłada zautomatyzowanie i uproszczenie prac związanych z tworzeniem kodu aplikacyjnego.

Koncepcja No Code wpisuje się w założenia idei Low Code, z tym wyjątkiem, że w pełni automatyzuje proces tworzenia kodu aplikacyjnego – i nie pozwala użytkownikom na ingerowanie w wygenerowany kod źródłowy.

Główną różnicą pomiędzy środowiskami niskokodowymi i bezkodowymi jest skala automatyzacji.

Rozwiązania Low Code oraz No Code są skierowane do dwóch różnych grup użytkowników.

Rozwiązania Low Code i No Code w pewnym stopniu zmieniają zadania stojące przed zespołami IT – a także zakres odpowiedzialności zespołów biznesowych.



Piotr Waszczuk

Szybki

jak Low Code

Na konferencjach Low Code rozmawiamy dziś przede wszystkim o szybkości dostarczania aplikacji – mówi **Manoj Nair**, Head of Digital Transformation w C&F. To właśnie skokowy przyrost szybkości dostarczania aplikacji – a więc i efektywności pracy programisty – decyduje o coraz większej popularności Low Code, także w silnie regulowanych sektorach gospodarki.

Czy z perspektywy C&F obecna fala wdrożeń platform Low Code ma charakter przełomowy? Czy ich czas nadszedł ze względu na deficyt pracowników w IT, cyfryzację środowiska pracy i procesów biznesowych, popyt na przestrzeń do szybkich zmian, wprowadzanych również przez biznes?

Myślę, że jest nawet więcej czynników, które powodują, że rozwiązania Low Code spotykają się z zainteresowaniem firm. Pracujemy z tą technologią już 8 lat i owszem, zawsze wśród jego głównych zalet wymienia się właśnie szybkość wytwarzania oprogramowania. Ale Low Code to znacznie więcej: zdecydowanie niższy poziom kosztów wdrożenia, licencji i utrzymania, a także dużo prostszy sposób zarządzania zmianami i aktualizacjami, co jest znakomitą receptą na dług technologiczny.

Te obserwacje zdecydowały zresztą o podjęciu przez naszą firmę poszukiwań alternatywy dla rozwiązań powstających w tradycyjnych, wysokokodowych językach programowania. Nasze poszukiwania ostatecznie zwięźczyły współpracę z firmą OutSystems.

Co wyróżnia ofertę tego dostawcy?

Gotowy produkt i nastawienie na szybkie budowanie aplikacji

– tak, żeby biznes mógł wykorzystać swoje pomysły, dopóki są świeże. Takim wyróżnikiem jest więc m.in. radykalne przyspieszenie tzw. Time to Market. OutSystems pozwalał również od początku budować aplikacje bezpieczne i skalowalne, co dziś jest standardem w kategorii Low Code, ale 7–8 lat temu nie było tak oczywiste.

Najważniejsza jednak w Low Code jest szybkość budowania aplikacji...

Zdecydowanie – za pomocą takich rozwiązań **aplikacje powstają 4–5 razy szybciej, niż przy użyciu tradycyjnych narzędzi i języków programowania.** Gdyby chcieć utrzymać takie tempo tradycyjnymi metodami – i gdyby to było możliwe – to kolejne etapy życia aplikacji, a więc utrzymanie i rozwój, pochłonęłyby nieporównanie więcej zasobów zespołu IT. Mówiąc krótko – nie wystarczyłoby programistów i specjalistów od utrzymania.

Może zatem jeszcze lepiej byłoby, gdyby zamiast na Low Code postawić na No Code?

Niekoniecznie. Koncepcja No Code ma bardziej ograniczone zastosowanie. Trudno sobie przecież wyobrazić wytworzenie przez citizen developerów większego systemu o znaczeniu krytycznym.



Trudno też wyobrazić sobie, że nagle masowo zaczną powstawać fragmenty procesów, albo i całe procesy, osadzone poza ramami zdefiniowanej, zrównoważonej i zabezpieczonej architektury. **Low Code ma służyć zwiększeniu efektywności pracy deweloperów, a nie namaszczaniu na deweloperów osób, które nie mają do tego odpowiednich kompetencji**, ani nie mają na to czasu, ponieważ już realizują inne zadania. Chwila prawdy przychodzi również w momencie testów takich produktów i ich integracji – aplikacje powstałe w Low Code integrowane są w zasadzie automatycznie i natychmiastowo.

Jak to udoskonalenie odbierają sami programiści?

Opór wewnątrz firmy jest na początku niekiedy silniejszy niż zewnętrzna konkurencja, oferująca inne rozwiązanie. **Programiści obawiają się, że platforma Low Code zabierze im pracę. Jednak dość szybko okazuje się, że to środowisko nie tyle zabiera pracę, co po prostu mocno ją zmienia.** Z reguły na lepsze, bo eliminuje z niej elementy powtarzalne i żmudne.

To tylko obawy o utratę pracy, czy także o możliwości sprawowania kontroli nad wytwarzaniem oprogramowania i w efekcie – utratą bezpieczeństwa i spójności przez całość?

Oczywiście, **obawy dotyczą również ryzyka chaosu, jaki może wywołać Low Code za sprawą zbyt dużej szybkości wytwarzania aplikacji.** Z drugiej strony, spotykamy się często z niedowierzaniem, że w tym narzędziu można wytworzyć, przykładowo, system do obsługi kilkuset tysięcy użytkowników.

Aby przekonać potencjalnych klientów do możliwości Low Code, zabiegamy zawsze o zgodę na przeprowadzenie projektu PoC, który pozwoli zweryfikować, czy wskazane zadanie da się wykonać – i w jakim czasie. Pamiętam wiele projektów PoC, kiedy w ciągu kilku godzin byliśmy w stanie przedstawić rozwiązanie, którego wytworzenie przy tradycyjnym podejściu wymagałoby kilku dni pracy, a na bieżąco zgłaszane ewentualne zmiany w zakresie wprowadzaliśmy na bieżąco.

Na kanwie takich wniosków możemy szybko dojść do porozumienia z doświadczonymi programistami. Dla nich będzie to narzędzie automatyzacji pracy, przyspieszenia jej, ale także sposób na wprowadzanie do zespołu młodszych programistów. Co więcej, praca w Low Code to również sposób na rozwinięcie skrzydeł w nowych obszarach, np. w zakresie budowy aplikacji webowych i mobilnych, lokowania nowych aplikacji w chmurze, czy tworzenia aplikacji typu Cloud Native bez dodatkowych etatów dla specjalisty od sieci, chmury czy bezpieczeństwa.

Podejście niskokodowe w naturalny sposób wspiera też koncepcję DevOps, a poszczególne platformy Low Code najczęściej pozwalają na łatwe zarządzanie rozwiązaniami wytworzonymi przy ich użyciu,

Praca w Low Code to również sposób na rozwinięcie skrzydeł

w nowych obszarach, np. w zakresie budowy aplikacji webowych i mobilnych, lokowania nowych aplikacji w chmurze, czy tworzenia aplikacji typu Cloud Native bez dodatkowych etatów dla specjalisty od sieci, chmury czy bezpieczeństwa.

a także monitorowanie ich wydajności. Możliwość szybkiej identyfikacji ewentualnych błędów oraz ich usuwania w celu doskonalenia wydajności jest istotą podejścia Low Code.

Wiele firm oczekuje jednak możliwości dostosowania rozwiązań do specyficznych, często unikalnych potrzeb. Jak odnajduje się tu podejście Low Code?

Bardzo dobrze. Jeśli chodzi o możliwości dostosowywania rozwiązań do potrzeb i pomysłów biznesu na nowoczesnej platformie Low Code, to **nie ma w zasadzie żadnych ograniczeń**. Podstawowe elementy można dowolnie, na własne potrzeby oprogramować. Inaczej jest w przypadku No Code, gdzie producent platformy zapewnia najczęściej mocno ograniczoną liczbę szablonów dla danej usługi.

Od czego zależy efektywność podejścia Low Code?

Takich czynników jest kilka, ale najważniejsze znaczenie ma skala użycia. Proszę pamiętać, że w projektach IT, a takimi są projekty aplikacyjne, chodzi o uzyskanie jak najlepszej stopy zwrotu z inwestycji. **W przypadku platform Low Code wskaźnik ROI jest sensowny, gdy osiągnie się pewną skalę ich wykorzystania.** Przykładowo, nie ma sensu zaprzęganie platformy Low Code w celu zbudowania i rozwoju pojedynczej aplikacji. Korzyści zaczynają się wtedy, gdy wpisujemy tę technologię w strategię rozwoju środowiska aplikacyjnego, określając np., że większość aplikacji wewnętrznych ma powstawać w modelu niskokodowym. Należy przy tym pamiętać, że aplikacje zbudowane w podejściu Low Code powinny dobrze obsługiwać wewnętrzne procesy, automatyzować obszary pracy manualnej, zwiększać produktywność klienta wewnętrznego oraz poziom jego samoobsługi. Z drugiej strony, ich zadaniem niekoniecznie powinno być np. podnoszenie poziomu doświadczeń klienta (Customer Experience).

Czy platformy Low Code nadają się do modernizacji i integracji krytycznych systemów, opartych na starszych technologiach?

Oczywiście – i często są do tego wykorzystywane. Nie bez powodu. Proszę zauważyć, że dostawcy oprogramowania biznesowego coraz częściej postanawiają przenieść swoje rozwiązanie do chmury i zachęcają, a czasem wręcz zmuszają klientów, aby przeszli na model cloud computing. Klienci są w takiej sytuacji postawieni pod ścianą, a do tego okazuje się, że migracja nie jest – bo przecież nigdy nie jest – bezproblemowa. Koszty rosną.

Low Code oferuje tu alternatywę. Pozwala bowiem na łatwe skopiowanie funkcjonalności i integracji do nowych aplikacji. Rok temu zrealizowaliśmy projekt dla klienta, który na całym świecie miał ponad 15 systemów instalacji SAP i musiał przenieść je do środowiska SAP HANA, odtwarzając połączenia z kilkudziesięcioma kluczowymi systemami dla każdej instancji SAP. Po przekalkulowaniu czasu, kosztów i zasobów, niezbędnych do realizacji takiego projektu w tradycyjnym podejściu, zdecydowano się na obudowanie standardowej implementacji SAP HANA



Platformy Low Code nadają się do modernizacji i integracji krytycznych systemów, opartych na starszych technologiach.

własnymi, wytworzonymi z użyciem podejścia Low Code aplikacjami, które zapewniają niezbędne, dostosowane do biznesu funkcje i integracje. Dzięki temu możliwe było istotne skrócenie czasu realizacji projektu. Osiągnięto także znaczące oszczędności na kosztach licencji SAP.

Aplikacje Low Code często zastępują w ten sposób klastry funkcjonalne złożone z wielu, często przestarzałych systemów zbudowanych na przestrzeni lat do obsługi jakiegoś procesu. Często służą też do realizacji innowacyjnych pomysłów, które jeszcze nie są gotowym produktem. Aplikacja Low Code powstaje szybko i ma przez to większą szansę przeistoczyć się w faktyczny produkt, zdążyć przed konkurencją i wpisać się w rynkowy timing.

Zapewne koszty takich eksperymentów produktowych są również mniejsze – podobnie jak obawy przed ich zamknięciem w razie niepowodzenia...

To prawda. Takie jest generalnie nastawienie. Za sprawą podejścia Low Code możliwa jest też zmiana struktury kosztów w środowiskach aplikacyjnych.

Dawna struktura budżetu przy wdrożeniach systemów zakładała przeznaczenie 80% środków na utrzymanie i wsparcie, a jedynie 20% na rozwój aplikacji. Dziś te proporcje się odwróciły, również za sprawą Low Code. Innowacja stała się kluczowa z punktu widzenia klienta.

Czy aplikacje zbudowane na podstawie koncepcji Low Code można przenosić pomiędzy różnymi platformami?

To ważna kwestia dla wielu firm. Dlatego **od początku szukaliśmy rozwiązania, które zapewnia taką możliwość.** W 2015 roku w zasadzie jedynie OutSystems pozwalał na budowę aplikacji, które można przenieść do .NET lub Javy. Obecnie nadal wspierana jest transkrypcja do .NET. Sprawdzaliśmy jego możliwości dla dużego systemu i **w ciągu 4–5 dni udało się uzyskać bardzo przyzwoity wynik.**

Jest to oczywiście ważna opcja w przypadku tworzenia kluczowych systemów. Dotychczas jednak nie mieliśmy jeszcze okazji asystowania klientowi w takim powrocie. Żaden z naszych klientów nie zrezygnował z implementacji OutSystems ani wytworzonych na platformie systemów.

Decydująca była z pewnością kwestia komfortu, wygody, jaką oferuje ta platforma. Jest to bardzo ważny element całej układanki: **platformy Low Code oferują zwykle katalog gotowych konektorów z popularnymi systemami i usługami.** Jeżeli więc pojawia się pomysł na użycie ChatGPT, to można zrealizować od ręki. W ramach platformy OutSystems **można także samemu budować konektory do innych środowisk i platform.** Wówczas elementy naszego rozwiązania możemy wykonywać w technologiach, takich jak: .NET, HTML, CSS czy JavaScript – w efekcie powstają rozwiązania, które znacznie wykraczają ponad to, co oferuje platforma Low Code.



To wymóg ważny w przypadku dużych organizacji, potrzebujących skalowalnych aplikacji, które można łatwo dostosowywać do potrzeb i od podstaw rozwijać w architekturze kontenerowej lub chmurowej.

Platformy, które zaspokajają podobne potrzeby, są dziś niezwykle pożądane na rynku, ponieważ dzięki nim można skupić się na logice biznesowej aplikacji, a nie na upewnianiu się, że będzie w danym środowisku działać.

W dyskusji podczas spotkania CXO HUB pojawił się zarzut, że platformy Low Code mają krótki czas życia, a więc istnieje duże ryzyko wyboru takiej, której rozwój zakończy się w ciągu kilku lat.

Ile jest dziś na rynku pewnych, stabilnych platform Low Code, spośród których można wybierać?

Obecnie możemy mówić o kilku najbardziej stabilnych – i **najlepiej prosperujących – rozwiązaniach, które wymienia także Gartner: OutSystems, Mendix, Microsoft, Salesforce.** Platformy te obecne są na rynku od ponad 10 lat i mają wciąż rosnący portfel klientów.

Jakich wzrostów efektywności pracy deweloperów można oczekiwać od rozwiązań Low Code?

Aplikację, która w tradycyjny sposób powstaje w 6–12 miesięcy, w podejściu Low Code można zakończyć w 2–3 miesiące. Oczywiście nawet w tak krótkim czasie może się zdarzyć, że klient końcowy zażyczy sobie zmian w trakcie projektu. Low Code doskonale sobie z taką dynamiką projektową radzi – implementacja pojedynczych zmian zajmuje minuty, zaś tworzenie nowych funkcjonalności to kwestia godzin, maksymalnie tygodni.

Jak złożony jest proces aktualizacji platformy Low Code, na której przecież opierają się aplikacje biznesowe?

Firmy bardziej konserwatywne, które decydują się na wykorzystanie platformy Low Code w modelu on-premise, same decydują o czasie i trybie aktualizacji, oczywiście przy wsparciu dostawcy.

Sytuacja doskonale się upraszcza, kiedy mamy do czynienia z platformą Low Code działającą w modelu chmury obliczeniowej. Wówczas utrzymanie jest całkowicie pomijalne, a klient i my, jako partner wdrożenia, otrzymujemy tylko informację o planie wdrożenia aktualizacji. Jeśli nie zgłosimy uwag, to stosowna zmiana zostanie automatycznie wprowadzona w określonym czasie.

Jednocześnie, gdyby po takiej aktualizacji pojawił się problem, można natychmiast cofnąć się do kopii zapasowej sprzed wdrożenia.

Aplikację, która w tradycyjny sposób powstaje w 6–12 miesięcy,

w podejściu Low Code można zakończyć w

2–3
miesiące.

Te funkcje są opłacone w kosztach licencji – nie wymagają dodatkowych kosztów ani stresu związanego z wielką migracją. Najczęściej raz na 1–2 lata pojawia się nowa wersja platformy, ale wówczas jej aktualizacja także nie jest bardzo czasochłonna ani ryzykowna – **zajmuje przeciętnie 12–24 godziny**. To zupełnie inna skala, niż przy migracjach standardowego oprogramowania.

Każda migracja kojarzy mi się jednak z ryzykiem – w jaki sposób systemy zbudowane w Low Code mogą je niwelować?

Zawsze jest ryzyko, ale w Low Code jest też możliwość natychmiastowej reakcji. Platforma sama identyfikuje obszary i elementy, które źle przeniosły się do nowej wersji i wymagają naprawy. Najczęściej są to po prostu obszary i elementy, które zostały wytworzone poza szablonem i wobec tego wymagają ręcznej interwencji.

Takie podejście zdecydowanie skraca czas i problemy związane z migracją – wszelkie błędy obsługuje się w praktyce na bieżąco. Trudniejsze zmiany mogą być też eskalowane do producenta, ale ciągle jest to nieporównanie mniejsze obciążenie niż przy tradycyjnym podejściu. De facto nie testujemy też kodu aplikacji, ale całe procesy biznesowe.

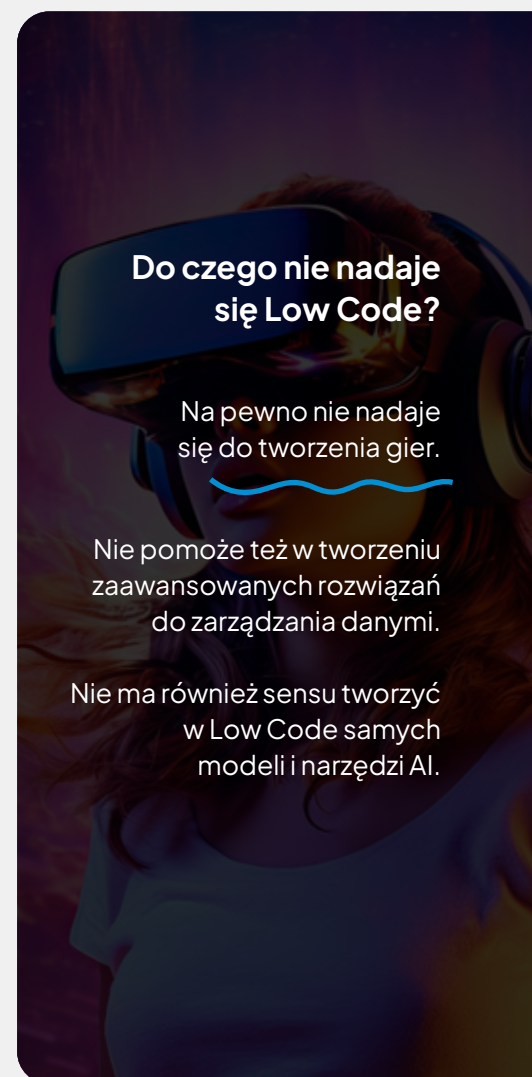
Czy największej konkurencji dla podejścia Low Code nie stanowią dziś rozwiązania oparte na generatywnej AI? Dlaczego programista miałby korzystać z platformy Low Code, skoro wirtualny asystent może wygenerować kod aplikacyjny za niego?

Moim zdaniem przyszłość to właśnie Low Code plus AI. Sztuczna inteligencja już dziś bardzo dobrze pomaga platformom Low Code w budowie aplikacji. Podpowiada gotowe fragmenty, które powinna zawierać. To wydatnie przyspiesza pracę. Szczególnie początkującym programistom. Pozwala również na zachowanie pełnej kontroli nad sposobem działania aplikacji.

Do czego na pewno nie nadaje się Low Code?

Na pewno nie nadaje się do tworzenia gier. Nie pomoże też w tworzeniu zaawansowanych rozwiązań do zarządzania danymi. Nie ma również sensu tworzyć w Low Code samych modeli i narzędzi AI – choć aplikacje powstałe w Low Code w naturalny sposób z takimi rozwiązaniami współpracują.

Co można powiedzieć o podejściu do koncepcji Low Code w Polsce?



Do czego nie nadaje się Low Code?

Na pewno nie nadaje się do tworzenia gier.

Nie pomoże też w tworzeniu zaawansowanych rozwiązań do zarządzania danymi.

Nie ma również sensu tworzyć w Low Code samych modeli i narzędzi AI.

Mam wrażenie, że o możliwość użycia Low Code **zabiegają zwłaszcza firmy z sektorów silnie regulowanych**. Przykładowo, szczególnie po doświadczeniu wojny na Ukrainie, widzimy wiele zapytań dotyczących wykorzystania Low Code w modelu chmurowym ze strony sektora bankowości.

Jest to o tyle istotne, że Low Code pozwala szybko wytwarzać aplikacje, które będą natywne w środowisku chmurowym. Oznacza to podwójną korzyść. Po pierwsze – jest to możliwość szybkiego wytworzenia działających narzędzi, które przejmować będą funkcje od starzejących się monolitycznych systemów on-premise. Po drugie, umieszczenie nowych aplikacji w chmurze pozwala zniwelować ryzyka zatrzymania biznesu wskutek, dajmy na to, sabotażu infrastruktury centrum danych.

Również rynek farmaceutyczny, medyczny – nawet silniej regulowane niż bankowy – mają coraz więcej doświadczeń i z ich strony pojawia się coraz więcej zapytań o Low Code. W ich przypadku podejście niskokodowe oznacza szansę nadrobienia opóźnień w transformacji cyfrowej.

Co ciekawe, na polskim rynku coraz częściej padają pytania o środowisko Low Code zintegrowane z AI. Prowadzimy rozmowy z klientami z branży spożywczej, którzy poszukują ciekawych scenariuszy użycia dla takich rozwiązań.

Wszystko to jedynie potwierdza prognozy Gartnera dla rynku Low Code, który z ok. 15 mld USD przed 5 laty urosnąć ma nawet do 50 mld w ciągu 2–3 kolejnych lat. Jest na to rynek, także w Polsce.

Jakie nowe kompetencje związane z podejściem Low Code są potrzebne, aby efektywnie z tego podejścia korzystać?

Jako dostawca, zapewniamy platformę, pomagamy zbudować pierwszą aplikację i wspieramy lokalny zespół. Może to być początek drogi do uruchomienia własnego centrum kompetencji Low Code, ale widać, że polskie firmy nie są zainteresowane tym, żeby utrzymywać powstające aplikacje we własnym zakresie. Nie jest zatem potrzebny liczny zespół czy specjalne kompetencje.

Na początku wystarczy doświadczony programista, znający architekturę i jeden czy dwóch deweloperów na poziomie nawet juniora. Praktyka pokazuje, że **po kilku miesiącach będą oni w stanie samodzielnie dostarczać i utrzymywać aplikacje,** zastępując de facto cały liczny dział oprogramowania.

Efektywność, która jest znakiem firmowym Low Code, ma dwa oblicza. Z jednej strony jest to zdecydowanie większa szybkość dostarczania aplikacji, a z drugiej – o wiele większa łatwość utrzymania wielu aplikacji przez nawet pojedynczego programistę.



Rozmawiał
Szymon Augustyniak

Dlaczego

Low Code

wkracza

na salony?

Za rosnącą powszechnością podejścia niskokodowego stoi naturalne dążenie do ułatwiania pracy, ale także bardzo konkretne wartości biznesowe. Walorów jest jednak więcej, a i okres popularyzacji koncepcji Low Code nie jest przypadkowy.

Popularność niskokodowego podejścia do rozwoju aplikacji rośnie nie bez powodu. Za wykorzystaniem rozwiązań pozwalających wprost zaangażować właścicieli biznesowych i przyszłych użytkowników aplikacji w proces ich tworzenia przemawia wiele aspektów. Przede wszystkim, podejście Low Code i No Code pozwala na odciążenie zespołów IT, a także skrócenie etapów przygotowania specyfikacji oprogramowania – a co za tym idzie – umożliwia wydatne przyspieszenie procesów rozwoju aplikacji.

Low Code

dla każdego

Low Code

dla każdego

Low C

Low Code

dla każdego

Low

Na rynku dostępnych jest dziś wiele platform klasy Low Code, a ich liczba ciągle rośnie. **Na łamach opublikowanego w sierpniu 2022 roku rankingu Gartner Magic Quadrant for Enterprise Low-Code Application Platforms w gronie liderów sklasyfikowane zostały platformy oferowane przez firmy: OutSystems, Mendix, Microsoft, Salesforce oraz ServiceNow.** Do grupy wizjonerów eksperci Gartnera zaliczyli natomiast dostawców, takich jak: Appian, Zoho i Pegasystems. Z kolei w kwadrancie pretendentów (challengers) znalazła się jedynie firma Oracle. Zdecydowanie liczniejsza jest zaś grupa dostawców niszowych, a – dodajmy – jest to jedynie czołówka rynku rozwiązań Low Code wytypowana przez analityków firmy Gartner. Wybierać można również spośród platform open source, a także tych oferowanych w modelu usługowym i w ramach publicznych środowisk chmury obliczeniowej. **Stopniowo znikają zatem typowe bariery związane z wdrożeniem, choćby testowym, środowisk umożliwiających tworzenie aplikacji w myśl idei programowania niskokodowego.**

Poszczególne środowiska Low Code różnią się m.in. stopniem zaawansowania, złożonością czy liczbą dostępnych konektorów integracyjnych. Istota ich działania jest jednak zbliżona – definiowanie podstawowej funkcjonalności i warstwy obsługi budowanych aplikacji odbywa się za pośrednictwem interfejsu graficznego, w którym użytkownicy układają wybrane komponenty, definiując dodatkowo ich zależności i parametry działania. Takie, typowe dla środowisk niskokodowych, podejście sprawia, że jest to sposób tworzenia aplikacji najłatwiejszy do opanowania ze wszystkich dotychczas stosowanych – zwłaszcza dla osób nietechnicznych. Jest to zarazem kluczowa cecha koncepcji Low Code.

Dlaczego

właśnie teraz?

Dlaczego

właśnie teraz?

Dlaczego

Dlaczego

właśnie teraz?

Dlaczego

Koncepcja Low Code i No Code nie jest nowością. Próby zredukowania technicznych wymagań towarzyszących procesowi tworzenia aplikacji są podejmowane od wielu lat. Już kilka dekad temu na rynku pojawiały się platformy wspierające modelowanie procesów biznesowych, a dodatkowo umożliwiające wygenerowanie prostego kodu aplikacyjnego, automatyzującego wybrany proces na bazie predefiniowanych elementów. Z takiej koncepcji wywodzi się m.in. platforma Oslo firmy Microsoft. W swoich założeniach to środowisko, zapowiedziane 15 lat temu, miało zapewniać fundament pozwalający osobom niebędącym programistami na tworzenie aplikacji z wykorzystaniem, dostępnych w stosownym repozytorium, modeli aplikacyjnych i metadanych. Według zapowiedzi, takie podejście do rozwoju oprogramowania miało ułatwić rozwój kodu aplikacyjnego, choć w swoich początkowych założeniach nie zakładało zaangażowania w ten proces osób niemających kompetencji technicznych. Pierwotnie na platformę Oslo składały się trzy kluczowe elementy: rodzina dedykowanych języków modelowania danych „M”, wizualne narzędzie deweloperskie oraz stosowne repozytorium. Ostatecznie jednak, w 2010 roku zaniechano dalszych prac nad tym środowiskiem - w użyciu pozostały jedynie stworzone pod tym kątem języki deweloperskie „M”. Oficjalnym powodem anulowania projektu miał być spodziewany niski poziom zainteresowania takim podejściem.

Ale obecnie realia są inne. Rosnąca złożoność procesów biznesowych, dynamika otoczenia gospodarczego, uległy dodatkowemu wzmocnieniu w realiach pandemii COVID-19 oraz zmian geopolitycznych. Ostatnie lata przyniosły bowiem ogromne przyspieszenie w zakresie wykorzystania cyfrowych zasobów i kanałów komunikacji. Podobnie, przyspieszyły projekty cyfrowej transformacji wielu firm i instytucji publicznych. **Potrzeba zapewnienia możliwości szybkiego adaptowania sposobu działania organizacji do działania w realiach pandemii czy przerwanych łańcuchów dostaw stała się dziś pewną normą. Zdolność do szybkiego wdrażania zmian biznesowych okazała się kluczowa dla przetrwania i powodzenia wielu przedsiębiorstw.** Niezbędne stało się zatem skrócenie czasu dostarczania nowych rozwiązań, w tym głównie aplikacji będących podstawą dla kolejnych usług czy produktów świata cyfrowego.

W poprzek takim potrzebom stanęły jednak realia światowego rynku pracy. Popyt na kompetencje techniczne wywindował stawki programistów do rekordowych poziomów. Poziomów, na które wiele organizacji nie mogło sobie pozwolić, bądź też - tego typu wydatki uznano za nieefektywne. Poszukiwano zatem podejścia alternatywnego. Koncepcja Low Code i No Code dojrzała we właściwym momencie - i świetnie się tu odnalazła. Poza tym podejście niskokodowe jest podstawą dla wielu inicjatyw z obszaru robotyzacji procesów biznesowych, a jako takie - wpisuje się w potrzeby związane z masową automatyzacją procesów, tj. hiperautomatyzacją.

Koncepcja
Low Code
i No Code
nie jest
nowością

Próby zredukowania technicznych wymagań towarzyszących procesowi tworzenia aplikacji są podejmowane od wielu lat. Już kilka dekad temu na rynku pojawiały się platformy wspierające modelowanie procesów biznesowych, a dodatkowo umożliwiające wygenerowanie prostego kodu aplikacyjnego, automatyzującego wybrany proces na bazie predefiniowanych elementów.

Co ciekawe, jak wynika z ubiegłorocznego opracowania firmy Microsoft: „Low-Code Trend Report 2022: Building a learning culture on a low-code platform”, zapewnienie użytkownikom biznesowym, których rola dotąd skupiała się jedynie na definiowaniu potrzeb i używaniu aplikacji stworzonych przez kogoś innego, dostępu do platform Low Code / No Code pozytywnie wpływa na ich zaangażowanie, zadowolenie, a nawet przekłada się na zmniejszenie obciążenia pracą.

Pracownicy działów biznesowych zyskują zatem narzędzie do niemal samodzielnego tworzenia aplikacji odpowiadających im indywidualnym potrzebom. **W realiach stałej presji na poprawę efektywności i redukcję kosztów działania, tego typu demokratyzacja podejścia do rozwoju oprogramowania coraz częściej staje się motorem innowacji biznesowych.** Pozwala bowiem na poszukiwanie punktowych usprawnień, szczególnie cennych tam, gdzie ogólna efektywność organizacji została już wielokrotnie zoptymalizowana i opomiarowana.

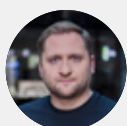
Teraz i na zawsze

Teraz i na zawsze

Tera

Podejście niskokodowe nie jest pozbawione ograniczeń. Platformy Low Code i No Code, nawet uzupełnione dedykowanymi fragmentami kodu aplikacyjnego, nie są obecnie zalecane do wykorzystania na potrzeby rozwoju najbardziej kluczowych aplikacji biznesowych, a także tych, o mocno rozbudowanej czy unikalnej funkcjonalności. **Istnieje jednak wiele scenariuszy, w których podejście Low Code dobrze uzupełnia dotychczasowe modele rozwoju oprogramowania, pozwalając jednocześnie na zwiększenie dynamiki prac.**

Użytkownicy biznesowi mogą bowiem z dużą dozą samodzielności definiować nowe interfejsy obsługi, proste automatyzacje, a nawet aplikacje stanowiące fundament usług czy produktów końcowych – bez potrzeby nadmiernego zaangażowania zespołu IT i oczekiwania na realizację zleconych prac. W praktyce Low Code jest narzędziem, które w rękach zaangażowanych użytkowników biznesowych może stać się źródłem ciekawych innowacji i realnej przewagi konkurencyjnej.



Piotr Waszczuk

Low Code

pozwała budować
skomplikowane aplikacje
rozwiązujące złożone
problemy biznesowe

Z **Filipem Kobielskim**, szefem zespołu Low-code/No-code Hub
w dziale Technology Consulting EY Polska rozmawiamy o:

praktycznych
aspektach
wykorzystania
rozwiązań
niskokodowych
i bezkodowych

potencjalnych
scenariuszach
użycia technologii
Low Code w realiach
odpowiedzialnego
biznesu

możliwościach użycia
takich narzędzi
do modernizacji
przestarzałych
środowisk
aplikacyjnych

wyzwaniach
i dobrych praktykach
związanych
z wdrażaniem
takich koncepcji

Czym jest Low Code i do jakich zastosowań się nadaje?

Low Code to klasa narzędzi, które przyspieszają możliwość wdrażania aplikacji poprzez wykorzystanie predefiniowanych elementów. Nadal potrzebne jest kodowanie, choć w mniejszej skali niż w przypadku tradycyjnego podejścia do programowania.

Dobrym przykładem platformy Low Code są rozwiązania ServiceNow, Pega i Microsoft PowerApps w części portalowej. Budowanie aplikacji polega na wypełnianiu pewnej matrycy gotowymi elementami i pisaniu kodu odpowiedzialnego za specyficzną logikę czy działanie interfejsu obsługi.

Do stworzenia pełnoprawnej aplikacji potrzebne jest uzupełnienie gotowych elementów dostępnych w wybranej platformie niskokodowej własnym kodem aplikacyjnym. Bez tego można zbudować jedynie proste, w praktyce niemal nieprzydatne, rozwiązania oparte wyłącznie na połączeniu elementów standardowo zaszytych w platformie Low Code.

Warto też rozróżnić platformy Low Code przeznaczone dla użytkowników indywidualnych, od tych, zaprojektowanych pod kątem wykorzystania w środowiskach korporacyjnych. Jest to ogromna, często niedoceniana różnica.

Rozwiązania skierowane do zastosowań indywidualnych pozwalają na szybkie tworzenie aplikacji. Nie nadają się jednak do wykorzystania tam, gdzie potrzebne są dodatkowe mechanizmy kontroli czy ochrony danych.

Na takie potrzeby odpowiadają rozwiązania przeznaczone do zastosowań klasy Enterprise. Mają one wbudowane mechanizmy bezpieczeństwa, funkcje zapewniające zgodność z regulacjami i wspierają wiele innych aspektów niezbędnych do tego, aby mówić o produkcyjnym wykorzystaniu aplikacji Low Code w poważnych organizacjach.

Zwykło się postrzegać rozwiązania Low Code jako narzędzia przeznaczone do tworzenia trywialnych funkcjonalnie aplikacji. Rzeczywistość jest jednak inna. **Dzisiejsze platformy Low Code nadają się do tworzenia skomplikowanych aplikacji rozwiązujących złożone problemy biznesowe.** Mówimy o rozwijających się, ale już dość dojrzałych technologiach, które stanowią pełnoprawną alternatywę dla tradycyjnego sposobu tworzenia aplikacji.

Takie rozwiązania faktycznie zapewniają alternatywę dla tradycyjnego procesu wytwarzania aplikacji do zastosowań biznesowych, w tym dla banków czy największych korporacji.

Jak na tym tle wypada koncepcja No Code?

Do grupy No Code zaliczają się narzędzia, które umożliwiają tworzenie aplikacji bez pisania ani jednej linii kodu. W EY mocno przyglądamy się platformie Unqork. Pozwala ona na tworzenie praktycznie dowolnych rozwiązań aplikacyjnych – od interfejsu użytkownika, przez obiegi dokumentów, po dynamiczne modele baz danych dostosowujące tryb

Platformy Low Code
stanowią pełnoprawną
alternatywę dla
tradycyjnego sposobu
tworzenia aplikacji.

działania do różnych scenariuszy użycia. Wszystko to bez potrzeby pisania jakiegokolwiek kodu. Co jednak ciekawe, architektura tej platformy zakłada, że tworzone przy jej użyciu rozwiązania są de facto plikami .json. Nie ma tam jednak ani grama kodu. Wszystko jest budowane za pomocą narzędzi wbudowanych w platformę Unqork. To oznacza, że **mając wszystkie potrzebne funkcjonalności pod ręką, w ramach jednej platformy, można tworzyć rozwiązania ograniczone niemal jedynie własną kreatywnością.**

Dlaczego warto przyjrzeć się takim rozwiązaniom?

Podójście bezkodowe otwiera nowe możliwości rozwoju oprogramowania dopasowanego do potrzeb użytkowników biznesowych. Jest jednak też problematyczne, ponieważ zmienia status quo dla deweloperów, którzy – w naturalny dla siebie sposób – będą skłaniali się ku wykorzystaniu własnego kodu. Mają taką możliwość, ale stoi to w sprzeczności z pryncypiami architektonicznymi środowisk No Code i zazwyczaj wpływa negatywnie na wydajność tworzonego rozwiązania.

Jeśli jednak deweloper nauczy się narzędzia No Code, wyboru potrzebnych funkcjonalności spośród wszystkich dostępnych w danej platformie mechanizmów oraz ich dobrej konfiguracji pod kątem specyfiki tworzonego rozwiązania, to będzie w stanie dużo szybciej budować aplikacje dla użytkowników końcowych.

Jest to prawdziwy Game Changer. Nie musimy zajmować się żadnymi aktualizacjami platformy, bo dba o nią dostawca. Podobnie, nie trzeba aktualizować kodu aplikacji, bo żaden fragment kodu nie został napisany. Nie trzeba również dbać o całkowity koszt użytkowania rozwiązania, bo korzystamy z niego w modelu usługowym. Jest to zupełnie inne spojrzenie na proces tworzenia aplikacji.

Czy zatem można powiedzieć, że No Code i Low Code to odrębne ścieżki wywodzące się z tej samej koncepcji, ale skierowane do różnych grup użytkowników?

Myślę, że tak. W EY mam przyjemność przewodzić zespołowi, w skład którego wchodzi zarówno osoby z zapleczem biznesowym – przydatnym do tego, aby dobrze rozumieć istotę problemu, który przychodzi nam rozwiązać – oraz osoby wywodzące się z obszaru technologii, które znają zasady poprawnego tworzenia aplikacji.

Low Code czy No Code to jedynie nowe narzędzia, przy użyciu których budujemy aplikacje. Aplikacje, które muszą zostać dobrze zaprojektowane i zintegrowane z resztą infrastruktury, a nade wszystko muszą działać w sposób prawidłowy. Wszystko to jest niezbędne nie tylko w kontekście realizacji celu, jakiemu ma odpowiadać tworzone rozwiązanie, lecz także w odpowiedzi na pewnego rodzaju brak zaufania do takich narzędzi. Istnieje jednak trzecia grupa użytkowników takich rozwiązań. Mowa o osobach działających na pograniczu kompetencji technologicznych i biznesowych. Działa to idealnie, ponieważ są w stanie połączyć te dwa

No Code to prawdziwy Game Changer

bez aktualizacji platformy.

Nie musimy zajmować się żadnymi aktualizacjami platformy, bo dba o nią dostawca. Podobnie, nie trzeba aktualizować kodu aplikacji, bo żaden fragment kodu nie został napisany.

światy. Rozumieją, co obie strony chcą osiągnąć, a często są to dość sprzeczne potrzeby. Pamiętajmy, że historycznie zawsze istniał pewnego rodzaju konflikt interesów oraz problem na linii komunikacji między IT i biznesem. Tymczasem rozwiązania niskokodowe i bezkodowe dają możliwość scalenia tych światów.

Co sprawia, że firmy i instytucje z całego świata otwierają się na koncepcje niskokodowe?

Nie da się ukryć, że mamy dziś na świecie problem z dostępnością kompetencji w zakresie rozwoju oprogramowania oraz kosztami takich kompetencji. Coraz mniej jest organizacji skłonnych ponosić wysokie koszty tradycyjnego developmentu. Nie widzą w tym wartości biznesowej, tym bardziej wiedząc, że dostępne są dziś alternatywne metody rozwoju oprogramowania. W okresie pandemicznym w pewnym sensie przelała się czara goryczy na rynku.

Podejście Low Code i No Code zmienia wiele paradygmatów i przyspiesza budowanie aplikacji. Nagle okazuje się, że rezultaty możemy zobaczyć po tygodniach czy miesiącach, a nie po latach. Oczywiście nie znaczy to, że aplikacje niskokodowe mogą być rozwijane, a następnie funkcjonować bez nadzoru ze strony zespołów IT. Ludzie, którzy wiedzą, jak należy tworzyć i projektować aplikacje są niezbędni. Skala ich zaangażowania może być jednak zmniejszona. Część typowych aktywności związanych z tworzeniem oprogramowania może być realizowana przez użytkowników biznesowych, którzy w sposób wizualny są w stanie łączyć pewne gotowe elementy i stworzyć przynajmniej pewien szkic działania aplikacji.

Jest to o tyle ważne, że **niemal każda firma ma obecnie problem z długim czasem oczekiwania na realizację projektów prowadzonych przez wewnętrzne zespoły deweloperskie.** Jest to zazwyczaj spowodowane bardzo dużym zapotrzebowaniem ze strony biznesu oraz niewystarczającą dostępnością ludzi. Na projekty aplikacyjne trzeba czekać miesiącami, a czasem nawet latami. A taka perspektywa już dawno nie jest akceptowalna...

Do tej pory nie było alternatywy. Za sprawą podejścia niskokodowego możemy zmniejszyć inercję organizacji nadążającej za zmieniającymi się realiami. Kluczowe znaczenie ma dziś też, naturalna dla podejścia niskokodowego, zwinność i łatwość w dostosowywaniu rozwiązań do potrzeb. Pozwala to na ograniczenie ryzyka związanego z wejściem w ślepej uliczki rozwoju aplikacji. Nawet jeśli potrzebne będzie wyrzucenie części funkcjonalności, bo w trakcie projektu zmieniają się założenia czy realia biznesowe, to można łatwo zmodernizować tylko potrzebny fragment aplikacji zamiast rezygnować z efektów wieloletniego projektu.

Jakie są główne wyzwania towarzyszące przechodzeniu na Low Code oraz No Code?

Myślę, że główną barierą jest brak wiedzy na temat rozwiązań typu Low Code/No Code, ich rodzajów oraz potencjalnych scenariuszy użycia. W wielu organizacjach ciągle nie ma świadomości, że takie technologie istnieją. W niektórych firmach Low Code jest postrzegany jako synonim dla narzędzi klasy RPA, które są tylko jednym z obszarów zastosowania technologii niskokodowych.

Niemal każda firma ma problem z długim oczekiwaniem na realizację projektów prowadzonych przez wewnętrzne zespoły deweloperskie.

Podejście Low Code i No Code

zmienia wiele paradygmatów i przyspiesza budowanie aplikacji.

Istnieją dziś przecież rozwiązania Low Code umożliwiające tworzenie integracji, stron internetowych czy platform e-commerce, ale są także środowiska umożliwiające budowanie kompleksowych aplikacji biznesowych. Jako EY specjalizujemy się właśnie w tej ostatniej grupie narzędzi.

Kolejnym wyzwaniem jest zbudowanie dobrego zespołu, który będzie pracował na bazie technologii Low Code. Powodzenie tych pierwszych, pionierskich inicjatyw niskokodowych ma bowiem ogromne znaczenie dla zmiany postrzegania tego paradygmatu w całych organizacjach.

Barierą na drodze do wykorzystania potencjału Low Code jest też właściwe zarządzanie zmianą. Fakt, że możemy realizować projekty szybciej i więcej osób ma możliwość zaangażowania w tworzenie aplikacji sprawia, że musimy lepiej zarządzać procesem wytwórczym. Rozwiązania budowane przy użyciu koncepcji Low Code funkcjonują przecież wspólnie z innymi rozwiązaniami i w jednym środowisku aplikacyjnym. Low Code to jedynie inne podejście do tworzenia aplikacji. Nie może być więc osobnym bytem w środowisku IT.

Jak zatem zmienia się rola zespołów IT w obliczu wdrażania Low Code?

Bez dwóch zdań **zaangażowanie zespołu IT w kontekście wdrażania i obsługi aplikacji niskokodowych jest niezbędne.** Musimy mieć ludzi, którzy wiedzą, jak się tworzy aplikacje, aby w modularny sposób i z duchem nowego narzędzia zbudować rozwiązanie działające sprawnie – także w danym środowisku – i w sposób rzeczywiście odpowiadający potrzebom.

Możemy jednak oddać część zadań związanych z projektowaniem aplikacji użytkownikom biznesowym, a tym samym uwolnić trochę mocy zespołu IT. Obecnie zdecydowana większość, bo 80–90% nakładów pracy koniecznych do stworzenia nowej lub rozbudowania istniejącej aplikacji dla biznesu leży po stronie zespołów IT. Rola biznesu sprowadza się dziś głównie do nakreślenia pewnej wizji oczekiwanego rozwiązania i oceny prototypów.

W reżimie niskokodowym możemy wydzielić zaangażowanie zespołów IT do ok. 30–40% całościowej pracochłonności związanej z budową aplikacji. Kolejne 30–40% nakładów pracy znajduje się po stronie wspomnianych już zespołów interdyscyplinarnych łączących kompetencje techniczne i biznesowe. Reszta odpowiedzialności za stworzenie aplikacji w środowiskach Low Code leży już stricte po stronie biznesu. Na ten obszar składa się np. zaprojektowanie warstwy procesowo-użytkowej.

W jakich obszarach Low Code szczególnie dobrze się sprawdza?

Odpowiem na bazie doświadczeń, które w ciągu ostatnich kilkunastu miesięcy zbudowaliśmy na własne potrzeby i w ramach projektów, które realizowaliśmy dla naszych klientów. Początkowo skupiliśmy się głównie na tworzeniu aplikacji, w których można łatwo zarządzać interfejsem użytkownika. Przykładowo, dla klientów z sektora ubezpieczeniowego zbudowaliśmy narzędzia ułatwiające brokerom

W reżimie niskokodowym
możemy wydzielić
zaangażowanie zespołów IT

do ok. **30–40%**

całościowej pracochłonności
związanej z budową aplikacji.

przekazywanie zapytań polisowych do underwriterów odpowiedzialnych za analizę ryzyka. Przygotowaliśmy więc niezbędny workflow, a także warstwę użytkownika oraz integrację z systemami ubezpieczeniowymi. Mamy zatem dynamiczny i atrakcyjny dla użytkowników aplikacji interfejs, łatwy do manipulacji, ale działający na bazie integracji ze starszymi technologicznie systemami.

Szeroki wachlarz możliwości platform Low Code/No Code oraz ich stosunkowo łatwa integracja z systemami legacy sprawia, że możemy również rozważyć użycie tych narzędzi do wytworzenia nowej, zwinnej warstwy operacyjnej w organizacji. Przy takim podejściu możemy zwiększyć możliwości adaptacji do zmiennych warunków biznesowych, zmienić funkcje istniejących w firmie rozwiązań dziedzinowych, budując swojego rodzaju „bazy danych na sterydach”, oraz zmniejszyć wydatki związane z utrzymaniem czy też modernizacją dedykowanych, standardowych systemów.

Ciekawe jest również rozwiązanie, które zbudowaliśmy we współpracy z zespołem prawnym z kancelarii EY Law. Nasze rozwiązanie, na podstawie udzielonych przez użytkownika odpowiedzi wskazuje obszary niezgodności z regulacjami, ryzyka związane z niedostosowaniem do konkretnych przepisów, a także podpowiada, jakie działania należy podjąć, aby te luki zniwelować. Co ważne, to narzędzie, gotowy produkt dostępny dla klientów, zbudowaliśmy w ciągu zaledwie kilku miesięcy. Jest to świetny przykład na to, jak zespół techniczny może połączyć siły z zespołem biznesowym, by szybko i zwinnie zaprojektować rozwiązanie dla użytkowników końcowych.

Kolejnym obszarem użycia Low Code są punktowe projekty dotyczące typowo niedoinwestowanych i zaniedbanych procesów, również z obszaru IT. Przykładem jest zarządzanie podatnościami. Jest to proces, który uwzględnia wiele źródeł informacji, interesów osób zaangażowanych w proces i musi być przeprowadzony bardzo szybko.

Staramy się zatem iść w kierunku nietypowych obszarów zastosowania środowisk niskokodowych, tak aby wykorzystać pełnię ich możliwości właśnie w kontekście integracji odmiennych obszarów biznesowych, a także zwinności i szybkości działania. Możemy więc np. budować całościowe aplikacje, ale też proste narzędzia wspierające procesy, które obecnie są realizowane na bazie arkuszy kalkulacyjnych czy poprzez wymianę korespondencji e-mail. Jednocześnie, do podstawowej funkcjonalności możemy łatwo dodawać zarządzanie wyjątkami i komunikacją, raportowanie, a także logikę niezbędną m.in. w kontekście czyszczenia danych. To wszystko jesteśmy w stanie zbudować przy użyciu Low Code/No Code.

Szeroki wachlarz możliwości platform Low Code/No Code

oraz ich stosunkowo łatwa integracja z systemami legacy sprawia, że możemy również rozważyć użycie tych narzędzi do wytworzenia nowej, zwinnej warstwy operacyjnej w organizacji.

Czy takie podejście dobrze sprawdza się na potrzeby unowocześnienia i uzupełnienia możliwości przestarzałych, ale ciągle niezbędnych systemów dziedzinowych?

Jak najbardziej. **Modernizacja systemów typu legacy to obszar, w którym widać naprawdę ogromny potencjał podejścia niskokodowego i bezkodowego.** Możemy stosunkowo szybko zbudować rozwiązanie wpasowujące się w istniejące środowisko

i dostarczające potrzebną funkcjonalność na bazie istniejących już, być może przestarzałych, ale nadal działających systemów. Low Code, podobnie jak No Code, bardzo dobrze integruje się z innymi rozwiązaniami i usługami, także typu legacy.

Proszę pamiętać, że firma przymierzająca się do unowocześnienia przestarzałego technologicznie systemu, który nadal ma kluczowe znaczenie dla prowadzonej działalności, staje najczęściej przed decyzją o wydatkowaniu znaczących środków, a jednocześnie zgodą na utratę części kustomizacji czy własnych procesów.

Alternatywnie, jeśli chcemy zmienić lub udoskonalić proces biznesowy, aby lepiej odpowiadał potrzebom dzisiejszych czasów, możemy wykorzystać narzędzia niskokodowe do stworzenia dodatkowej warstwy operacyjnej. Nie mówimy więc o migracji systemu legacy do rozwiązania Low Code czy No Code, tylko o dalszym wykorzystaniu poczynionych kiedyś inwestycji w istniejące rozwiązanie oraz zmianie sposobu jego używania.

Rezygnujemy zatem z używania warstwy operacyjnej istniejących starszych systemów kluczowych, która zapewne nie do końca odpowiada aktualnym potrzebom, pozostawiając w nich całą warstwę przetwarzania danych i przenosząc operacje do nowego, dynamicznego środowiska zbudowanego na podstawie koncepcji Low Code/No Code. Operacyjnie warstwa ta może być lepiej dopasowana do dzisiejszych realiów i potrzeb biznesu, działać sprawniej i zapewniać użytkownikom lepsze doświadczenia, ale faktycznie bazować na sprawdzonym zapleczu.

Na jakim etapie drogi klientów w kierunku Low Code i No Code może pomóc zespół EY?

Możemy pomóc na dowolnym etapie. Globalne doświadczenie pozwala nam realizować każdy projekt – od zapewnienia inspiracji i podpowiedzi, dlaczego warto zainteresować się podejściem Low Code/No Code, przez wskazanie problemów biznesowych, dla których tę koncepcję warto zastosować oraz zaprojektowanie, konfigurację i wdrożenie aplikacji, po zarządzanie nią w dalszym etapie oraz przeszkolenie personelu klientów, aby mogli samodzielnie takie rozwiązania budować. Na koniec dnia to zaangażowanie użytkowników decyduje o powodzeniu tego rodzaju projektów. Takie doświadczenia mamy także w Polsce.

Ważne jest również tłumaczenie i obalenie pewnych mitów. Punktem wyjścia nie jest jedynie zoptymalizowanie procesu, ale też szybkie wprowadzenie nowego rozwiązania, uruchomienie nowych możliwości biznesowych.

Low Code i No Code świetnie wspierają innowacje biznesowe, a jednocześnie pomagają w zaangażowaniu szerszej grupy pracowników w inicjatywy rozwojowe.



Rozmawiał
Piotr Waszczuk

Globalne doświadczenie pozwala nam realizować każdy projekt

od zapewnienia inspiracji i podpowiedzi, dlaczego warto zainteresować się podejściem Low Code/No Code, przez wskazanie problemów biznesowych, dla których tę koncepcję warto zastosować oraz zaprojektowanie, konfigurację i wdrożenie aplikacji, po zarządzanie nią w dalszym etapie oraz przeszkolenie personelu klientów, aby mogli samodzielnie takie rozwiązania budować.



Kim jest

Citizen Developer

i kto może nim zostać?

Koncepcja Low Code/No Code pozwala zdemokratyzować proces tworzenia i rozwoju programowania, a także – choćby częściowo – przekazać go w ręce przyszłych użytkowników. Na rosnącą skalę użycia takich rozwiązań wpływają m.in. rosnący popyt na oprogramowanie i niedobór wykwalifikowanych programistów, a także wysokie koszty ich pracy.

Kim jest więc Citizen Developer i kto może nim zostać?

Citizen Developer to najczęściej pracownik działu biznesowego dysponujący umiejętnościami dostosowanymi do pełnionej w organizacji roli, ale nie mający wiedzy na temat kodowania – lub mający ją na niskim poziomie – który tworzy aplikacje dla swojego zespołu lub innych osób, korzystając z platform niskokodowych lub bezkodowych. Co ważne, mówimy tu o platformach zatwierdzonych i usankcjonowanych przez firmowy dział IT, który – w zależności od podejścia – nie bierze bezpośredniego udziału w tworzeniu danych aplikacji lub angażuje się w ograniczonym zakresie, ale nadzoruje proces ich powstawania. Co równie istotne, Citizen Developer to osoba, a nie pełniona rola.

W czasach, gdy firmom brakuje wyspecjalizowanych programistów i zasobów IT niezbędnych do wdrażania projektów cyfryzacyjnych, stosowanie idei rozwoju programowania obywatelskiego okazuje się mieć sens, zwłaszcza że według różnych badań, nawet 60% aplikacji jest obecnie tworzone poza tradycyjnym działem IT. Ponadto profesjonalni deweloperzy mają coraz więcej zajęć, a na ich barkach spoczywa coraz większa odpowiedzialność – nie tylko bowiem tworzą, zarządzają i rozwijają aplikacje, lecz także realizują często skomplikowane projekty transformacyjne. Citizen Developerzy mogą więc zdjąć z nich część obowiązków w zakresie tworzenia aplikacji, z których następnie będą sami korzystać.

Warto przy tym podkreślić, że osoby takie nie przejmują roli zespołów IT, ale współpracują z nimi, aby odciążyć specjalistów i przyspieszyć proces rozwoju aplikacji i umożliwić im skupienie się na bardziej złożonych projektach. Istotny jest również fakt, iż deweloperzy obywatelscy to członkowie zespołów pracujących po stronie biznesu. Są to więc osoby, które doskonale znają potrzeby i wyzwania z perspektywy tzw. biznesu. Mogą zatem – w dużej mierze samodzielnie – podjąć inicjatywę i opracować potrzebne narzędzia we współpracy z zespołem IT, który zadba m.in. o to, aby nowa aplikacja spełniała firmowe standardy m.in. w obszarze bezpieczeństwa.



Jakimi umiejętnościami powinien dysponować Citizen Developer?

Aby odnieść sukces jako programista obywatelski, przede wszystkim trzeba być osobą otwartą na pozyskiwanie nowych umiejętności oraz pracę z nowymi technologiami. Niezbędna jest też otwartość na naukę i współpracę z innymi. Ważne są również umiejętności komunikacyjne.

Kolejna istotna kwestia to zdolności analityczne. Programiści obywatelscy mają często do rozwiązania konkretne problemy – muszą więc opracować koncepcję rozwiązania na bazie dostępnych informacji. Citizen Developer powinien także umieć krytycznie myśleć o wspomnianych problemach w kontekście usprawnienia procesów w organizacji czy wykorzystania istniejących narzędzi.

Jakie korzyści Citizen Developerzy zapewniają swoim firmom?

Sprawdź korzyści, które przynosi organizacjom stosowanie idei obywatelskiego rozwoju oprogramowania. Należą do nich:



Oszczędność zasobów i zmniejszenie zaległości IT

Jest to szczególnie ważne, biorąc pod uwagę ciągły niedobór programistów. Według Forrester Research, do 2024 roku w Stanach Zjednoczonych będzie ich brakować nawet 500 tys. W Polsce zaś brakować ma nawet 150 tys. programistów. Citizen Developerzy mogą więc pomóc wypełnić tę lukę.

Obniżenie kosztów

Podejście Citizen Development obniża koszty rozwoju oprogramowania i pozwala lepiej wykorzystać istniejące w organizacji zasoby oraz kompetencje.

Poprawa współpracy między biznesem a IT

Citizen Developerzy – reprezentujący stronę biznesową – dobrze rozumieją wyzwania, przed którymi stoją ich zespoły. Wiedzę tę wnoszą do procesu tworzenia aplikacji czy przepływów pracy, sprawiając, że są one dopasowane do potrzeb biznesowych.



Mikołaj Marszycki

Źródła: Agilepoint, Creatio, Gartner, Kissflow, Pipefy, TechTrend

Przyspieszenie rozwoju nowych aplikacji

Według Project Management Institute (PMI), programowanie obywatelskie pomaga firmom przejść od zwinnych do hiperzwinnych modeli rozwoju oprogramowania. Taki sposób pracy może prowadzić do znacznego obniżenia kosztów i poprawy dynamiki wdrażania innowacji. Istnieją analizy wskazujące, że platformy Low Code /No Code skracają czas tworzenia aplikacji o 50–90%. Innymi słowy, programiści obywatelscy pomagają zespołom biznesowym rozwiązywać problemy szybciej i wydajniej niż tradycyjne metody rozwoju aplikacji.

Poprawa wydajności operacyjnej

Citizen Developerzy odgrywają kluczową rolę w skróceniu czasu rozwiązywania problemów biznesowych, co przekłada się na poprawę efektywności oraz zwinności firmy. Jest to istotne również w kontekście realizacji złożonych projektów innowacyjnych, które mogą być sprawniej realizowane przez mniej przeciążone pracą zespoły IT.

Poprawa elastyczności biznesowej

Dzięki odpowiednim narzędziom, obywatelscy programiści mogą sprawniej budować rozwiązania zapewniające funkcjonalności lub przepływy pracy odpowiadające bieżącym oczekiwaniom klientów, działaniom konkurencji czy pojawiającym się wyzwaniom biznesowym, co ma duże znaczenie dla usprawnienia dynamiki i elastyczności organizacji.

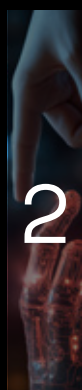
kroków

w kierunku wdrożenia konceptji Citizen Developer

Według Gartnera 61% firm już ma lub planuje wdrożenie funkcjonalnych inicjatyw na rzecz rozwoju obywatelskiego. Do końca 2023 roku liczba Citizen Developerów w dużych organizacjach przewyższy liczebność profesjonalnych deweloperów nawet w stosunku cztery do jednego.

Tymczasem wiele mniejszych firm wciąż nie jest do końca przekonanych do sensu funkcjonowania programistów obywatelskich lub nie wie, jak się zabrać do wprowadzenia w życie tej idei.

Prezentujemy 6 głównych kroków ułatwiających wdrożenie koncepcji obywatelskiego rozwoju oprogramowania w organizacjach biznesowych.



2

Wybieraj właściwe pomysły

Aby idea Citizen Developmentu przyniosła oczekiwane efekty, niezbędny jest właściwy wybór projektów – powinny być to inicjatywy realne do realizacji, przydatne i zachęcające do kolejnych prac na fali osiągniętych sukcesów.



4

Zadbaj o szkolenia dla pracowników

Podstawowe szkolenia techniczne, a także te związane z obsługą wybranej platformy Low Code/No Code są niezbędne, aby ułatwić pracownikom zespołów biznesowych faktyczne zaangażowanie w tworzenie wartościowych aplikacji. Poza szkoleniami warto też prowadzić cykliczne testy indywidualnych kompetencji.



6

Postaw na współpracę

Sprawna komunikacja pomiędzy wszystkimi zainteresowanymi stronami ma kluczowe znaczenie dla efektywności projektów rozwoju aplikacji w myśl koncepcji Citizen Development. Warto zatem zbudować wygodne środowisko współpracy pomiędzy użytkownikami biznesowymi zaangażowanymi w tworzenie aplikacji i zespołem IT, jasno określić zadania obu stron, a także zbudować reguły działania i je egzekwować.



Mikołaj Marszycki



1

Stwórz kulturę eksperymentowania z technologiami

Warto zachęcać swoich pracowników do eksperymentowania z technologią, podejmowania ryzyka, a przede wszystkim do dzielenia się swoimi pomysłami. Takie podejście wzmacnia atmosferę wspierającą innowacyjność i stanowi pierwszy krok do bezpośredniego zaangażowania pracowników działów biznesowych w rozwój aplikacji w reżimie Low Code/No Code. Konieczne jest upowszechnienie w organizacji myślenia projektowego.



3

Wybierz właściwą platformę Low Code/No Code

Rynek jest pełen takich narzędzi, ale tylko jedno z nich może być idealne dla konkretnej firmy. Od czego zależy wybór właściwej platformy? Przede wszystkim od tego, jak łatwo użytkownicy biznesowi z danej organizacji będą mogli tworzyć aplikacje. Należy wybrać platformę bezpieczną, mającą istotne dla firmy funkcjonalności, a także dopasowaną do posiadanego budżetu. Kluczem jest analiza potrzeb oraz możliwości. Warto skupić się na rozwiązaniach przeznaczonych do zastosowań korporacyjnych, po te bardziej konsumenckie sięgając jedynie na potrzeby testów.



5

Zapewnij wsparcie ze strony biznesu i IT

Należy bezwzględnie monitorować proces tworzenia aplikacji i w miarę potrzeb wspierać Citizen Developerów, aby zmniejszyć liczbę błędów i pomyłek na wszystkich etapach rozwoju aplikacji niskokodowych. Przed stworzeniem aplikacji programiści obywatelscy powinni dokładnie wiedzieć, dla kogo ją tworzą, jaki jest jej cel i pożądane rezultaty, a proponowane przez nich rozwiązania muszą pasować do istniejących celów biznesowych. Powinny też zostać poddane ocenie ze strony właścicieli konkretnych procesów biznesowych oraz zespołów IT. Pomoże to ocenić przydatność konkretnych inicjatyw Citizen Developer, wytypować najlepsze z nich do realizacji, a także zapewnić zgodność tworzonych narzędzi z obowiązującymi regulacjami.

No Low Pro

Code Code Code

- Co wybrać?
- Czym różnią się te koncepcje?

Koncepcja od No Code, przez Low Code, po Pro Code – synergie dostępnych podejść

Obecnie na znaczeniu mocno zyskują platformy łączące funkcjonalności typu No Code i Low Code, które pozwalają zespołom IT zaoszczędzić czas i skupić się na bardziej zaawansowanych zadaniach i aspektach projektów. Takie rozwiązania zapewniają również szerokie możliwości zespołom biznesowym i osobom poza IT, umożliwiając włączenie w proces tworzenia aplikacji tzw. Citizen Developerów.

Istnieje jednak trzecia koncepcja – Pro Code. Nowoczesne platformy powinny wspierać podejścia od No Code do Pro Code, z poziomu tego samego rozwiązania.

Wyjaśniamy na czym polegają różnice i synergie pomiędzy nimi.

Środowiska No Code i Low Code to przede wszystkim platformy, które pozwalają budować skomplikowane rozwiązania aplikacyjne i mają funkcjonalności umożliwiające przekazanie części – nierzadko znaczącej – procesu twórczego w ręce użytkowników biznesowych.

Dlaczego podejście niskokodowe zyskuje dziś na popularności?

Aby odpowiedzieć na tak postawione pytanie, zastanówmy się jakie grupy osób w danej organizacji najlepiej znają procesy biznesowe? Z pewnością są to osoby na co dzień zaangażowane w konkretne procesy, uczestniczące w przepływach informacji i pracy. To właśnie takie osoby najlepiej rozumieją specyfikę konkretnych procesów, problemów oraz obszarów, które mogą funkcjonować lepiej. Znają też własne potrzeby. Możliwość zaangażowania ich bezpośrednio w proces tworzenia aplikacji jest więc ogromną wartością. Co więcej, w tego rodzaju projektach często mamy do czynienia nie tylko z konkretnymi osobami, lecz także kilkuosobowymi zespołami użytkowników biznesowych z najlepszą wiedzą o tym, w jaki sposób zdigitalizować lub zmodernizować konkretny proces.

W bardziej tradycyjnym podejściu do budowania aplikacji zaangażowanie użytkowników również jest znaczące i bardzo ważne. Nie jest jednak tak bezpośrednie, jak w przypadku podejścia niskokodowego. Analitycy biznesowi najczęściej nie są w stanie z tak dużym poziomem dokładności przełożyć potrzeby i pomysły użytkowników na wymagania techniczne. Bezpośrednie zaangażowanie „hands-on” użytkowników biznesowych/analityków w proces budowania aplikacji znosi to ograniczenie. W modelu Low Code, w czasie zbliżonym do tego, ile zajmuje zebranie i opisanie wstępnych wymagań przy tradycyjnym podejściu, użytkownicy zamiast definiowania wymagań mogą samodzielnie stworzyć makietę aplikacji, która następnie zostanie udokumentowana, rozbudowana i zwalidowana przy zaangażowaniu zespołu IT.

Pozwala to na znaczące przyspieszenie innowacyjności organizacji, ale też zmniejszenie obciążeń po stronie zespołów deweloperskich. Niezależnie jednak od zastosowanego podejścia, także takie projekty powinny być realizowane w ściśle uporządkowany sposób. Oznacza to m.in. potrzebę prowadzenia całościowej dokumentacji projektowej oraz przejrzystego procesu zarządzania projektem. Chodzi m.in. o to, aby – eliminując wąskie gardła w procesach deweloperskich – uniknąć potencjalnych problemów związanych np. z rotacją pracowników.

< Czym zatem jest Pro Code?

Dla zaawansowanych projektów, gdzie elastyczność No Code i Low Code może okazać się niewystarczająca, istnieje podejście Pro Code.

Zespoły IT mogą korzystać z pełnej swobody w budowie rozwiązań, jednak to podejście wymaga głębszej wiedzy IT/programistycznej, a tym samym zaangażowanie deweloperów jest niezbędne.

Jednocześnie coraz częściej obserwujemy synergię pomiędzy dostępnymi strategiami. Połączenie podejścia Pro Code z No Code oraz Low Code otwiera drzwi do nowych możliwości w tworzeniu aplikacji.

Podejście Pro Code może wykorzystywać komponenty niskokodowe, ale uzupełnia je pełnoprawnymi zadaniami programistycznymi, obejmującymi np. uruchomienie dodatkowych mechanizmów bezpieczeństwa czy zapewnienie skomplikowanych integracji pomiędzy usługami chmurowymi i systemami on-premise. Ważne jednak, aby te funkcjonalności były możliwe i łatwe do połączenia ze standardowymi elementami platformy niskokodowej. Jednocześnie, tworzone w ten sposób komponenty mogą być następnie udostępniane szerzej Citizen Developerom z zachowaniem odpowiednich procesów Governance.

Przykładowo w myśl podejścia Pro Code programiści mogą budować dodatkowe komponenty, które będą potem wykorzystywane na zasadzie No Code/Low Code bezpośrednio przez użytkowników biznesowych zainteresowanych digitalizacją i automatyzacją procesów. W taki sposób rozwijane mogą być m.in. elementy związane z użytecznością aplikacji czy integracjami. Mówimy tu więc o łączeniu dwóch światów – IT i biznesu. Firmy korzystające z synerгии tych możliwości mogą skutecznie dostosować podejście do konkretnych potrzeb projektu, osiągając optymalne rezultaty.





Pomimo różnic, nowoczesne rozwiązania No Code i Low Code pozwalają na tworzenie aplikacji na bazie gotowych elementów, umożliwiając skrócenie prac projektowych. Ponadto, narzędzia wbudowane w platformy niskokodowe znacząco ułatwiają i przyspieszają prace nad nowymi aplikacjami.

Nowoczesne rozwiązania udostępniają możliwość automatycznego tworzenia struktury danych i tabel, bazując na istniejących arkuszach kalkulacyjnych, a następnie umożliwiają zdefiniowanie przepływu informacji, warstwy komunikacji oraz warunków akceptacji wydatków za pośrednictwem przejrzystego interfejsu typu „drag & drop”, którego obsługa nie wymaga żadnej wiedzy programistycznej. Po takich krokach, w ramach podejścia Low Code, mamy praktycznie gotową pierwszą wersję nowej aplikacji.

Idąc dalej, draft aplikacji może być tworzony automatycznie, na bazie dostarczonych informacji. Użytkownicy biznesowi, odpowiadając kolejno na pytania dotyczące m.in. modelu danych, procesu decyzyjnego i bezpieczeństwa, mogą w zautomatyzowany sposób uzyskać pierwszą makietę nowej aplikacji. Proces wytwórczy może być również wspierany przez generatywną sztuczną inteligencję, która jest w stanie udzielać użytkownikom podpowiedzi na każdym etapie tworzenia, rekomendując poszczególne akcje, kroki.

Można zatem przyjąć, że **Low Code zapewnia podstawy do tego, aby zautomatyzować proces tworzenia aplikacji w organizacjach. Pozwala bowiem łatwo definiować drzewa decyzyjne odpowiadające istocie konkretnych procesów**, a następnie – w prosty sposób – wdrażać je w środowisku biznesowym. Niech to będzie np. proces obsługi dostępu do firmowych miejsc parkingowych w rozproszonej geograficznie organizacji czy proces akceptacji wydatków, gdzie – poniżej pewnej kwoty – pracownicy mają pełną swobodę wyboru.



Jak wygląda zarządzanie pomysłami na wykorzystanie platform niskokodowych?

Dobłą praktyką jest wdrożenie procesu Idea – > Demand – > Project. Chodzi o zapewnienie pracownikom możliwości łatwego zgłaszania pomysłów, innowacji, głosowania i późniejszej ich oceny pod kątem potencjału biznesowego oraz technicznych możliwości realizacji. Często podczas takiej walidacji okazuje się, że wspierające rozwiązania już istnieją w firmie, ale nie są używane w sposób odpowiadający nowym pomysłom.

Koncepcje wstępnie zakwalifikowane do realizacji trafiają do fazy dalszych analiz, gdzie oceniane są kwestie związane z ładem korporacyjnym, realnym zapotrzebowaniem, bezpieczeństwem czy skalowalnością. W wielu organizacjach działania te są w dużej mierze zautomatyzowane. Potem rozpoczyna się proces faktycznego budowania aplikacji na bazie platformy niskokodowej początkowo przez użytkowników biznesowych – z odpowiednimi materiałami szkoleniowymi i przygotowaniem – a następnie z udziałem IT.

◀ Rola zespołu IT w projekcie niskokodowym

Nie jest jednak tak, że proces wytwórczy aplikacji – w myśl koncepcji No Code/Low Code – jest realizowany bez zaangażowania zespołu IT. Z kolei pomysłodawca danego rozwiązania nie zawsze faktycznie bierze udział w jego wykonaniu. W tym miejscu warto rozwinąć temat zaangażowania zespołu IT w tworzenie aplikacji niskokodowych. Przede wszystkim do uruchomienia samej platformy niskokodowej i tworzonych przy jej użyciu aplikacji niezbędna jest infrastruktura IT – zasoby, o które ktoś musi zadbać.

Jeżeli organizacja korzysta z platformy Low Code działającej w modelu usługowym, to najczęściej to jej dostawca gwarantuje zasoby sprzętowe niezbędne do samego środowiska niskokodowego, a także budowanych w nim aplikacji. To w pewnym stopniu ułatwia sprawę. Co jednak z tworzeniem kodu aplikacyjnego?

◀ Podejście Pro Code z synergiami No Code/Low Code, a rzeczywistość

Model Pro Code, czyli pełna swoboda programistyczna, może efektywnie współpracować z narzędziami No Code i Low Code, otwierając drzwi do większej elastyczności i szybkości w procesie tworzenia aplikacji. Model Pro Code zakłada uzupełnienie koncepcji Low Code kompetencjami deweloperskimi, wszędzie tam, gdzie gotowe komponenty platformy niskokodowej okazują się niewystarczającymi. Można założyć, że w takim ujęciu jeden „pro” deweloper jest w stanie wspierać od 5 do nawet 20 użytkowników zajmujących się rozwojem aplikacji niskokodowych. Jest to możliwe, ponieważ platformy typu No Code/Low Code tak naprawdę prowadzą użytkowników za rękę przez cały proces budowania aplikacji.

Co ważne, deweloperzy Pro Code uczestniczą w procesie oceny pomysłu danej aplikacji, oceny priorytetów czy definiowania dostępu dla użytkowników, którzy mają zająć się tylko budową rozwiązania. Takie decyzje w naturalny sposób dotyczą m.in. kwestii bezpieczeństwa danych i środowiska IT.

Faktem jest, że w przypadku mniej doświadczonych użytkowników zaangażowanie deweloperów będzie wyższe niż w przypadku osób, które wcześniej budowały aplikacje w reżimie No Code lub Low Code.

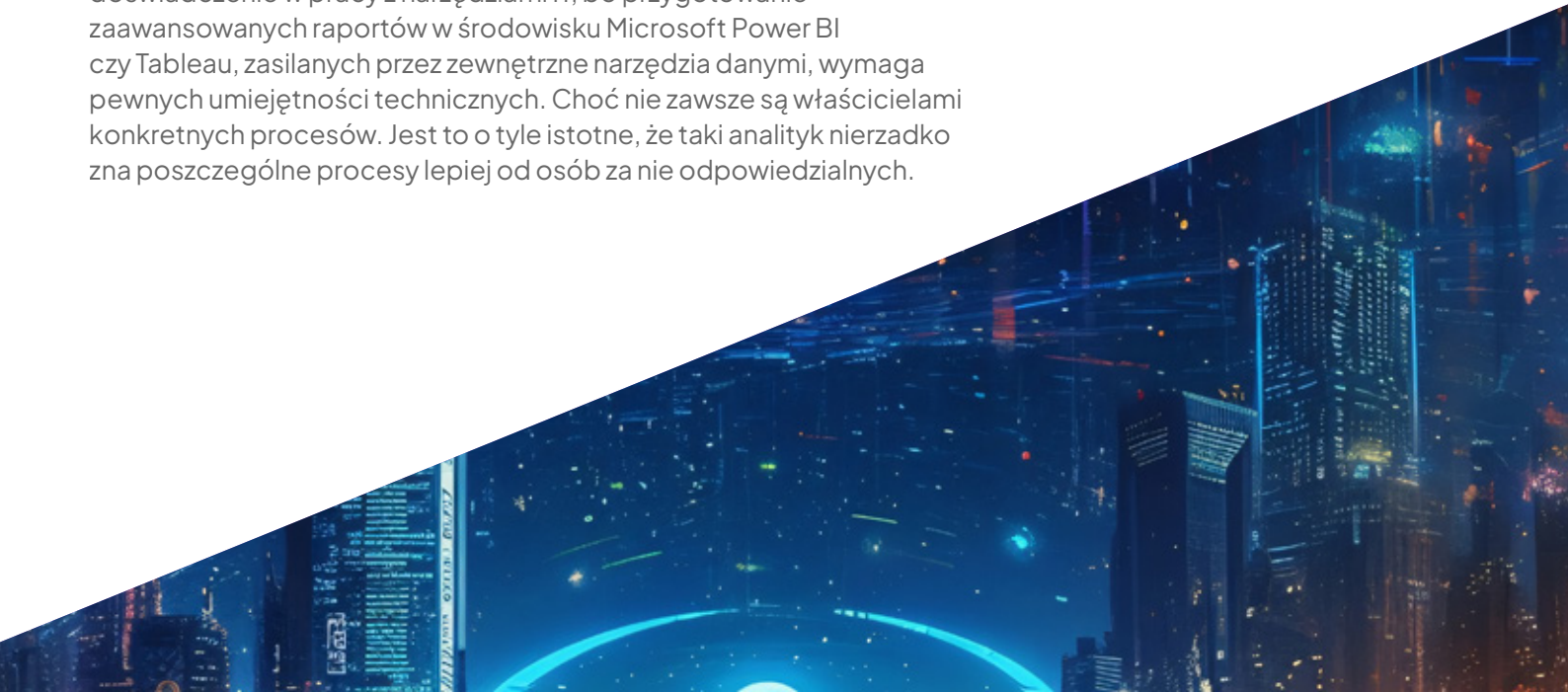
Wiele zależy też od dojrzałości danej organizacji. Tam, gdzie proces walidacji, przyznawania zasobów i dostępów, a także budowy aplikacji niskokodowych jest realizowany w dużej mierze ręcznie, początkowe zaangażowanie deweloperów sięga 40–60% czasu pracy nad projektem i spada, w miarę przechodzenia do kolejnych etapów.

Z kolei w tych firmach, które mają zautomatyzowane procesy – np. walidacji koncepcji aplikacyjnych pod kątem wydajności, zabezpieczeń i potencjału biznesowego – rolą zespołu IT jest głównie ocena jakości budowanego rozwiązania oraz cykliczna kontrola. Wówczas zaangażowanie wyspecjalizowanych deweloperów w procesie wytwórczym takiej aplikacji nie przekracza 20% czasu jej stworzenia. Jest to zatem ogromna oszczędność czasu specjalistów z obszaru IT, a zarazem możliwość bezpośredniego zaangażowania interesariuszy danego obszaru biznesowego w tworzenie rozwiązania dla nich samych.

◀ Kto może budować aplikacje niskokodowe, zostać Citizen Developerem?

Praktycznie każdy. **Największe szanse mają najbardziej zaangażowani użytkownicy biznesowi, którzy od początku starają się być najbliżej wdrożenia platformy niskokodowej czy bezkodowej.** Są to osoby żywo zainteresowane jej działaniem i efektami. Tym ważniejsze staje się zidentyfikowanie takich czempionów w ramach organizacji. Oczywiście osoba z jakimkolwiek doświadczeniem IT będzie miała niższy próg wejścia w technologie No Code/Low Code, ponieważ rozumie podstawy, ryzyka i limitacje działania systemów IT.

W większości departamentów zawsze się znajduje osoba, która chce być bliżej nowego rozwiązania od innych. Świetnymi kandydatami są np. analitycy biznesowi. Zazwyczaj mają pewne doświadczenie w pracy z narzędziami IT, bo przygotowanie zaawansowanych raportów w środowisku Microsoft Power BI czy Tableau, zasilanych przez zewnętrzne narzędzia danymi, wymaga pewnych umiejętności technicznych. Choć nie zawsze są właścicielami konkretnych procesów. Jest to o tyle istotne, że taki analityk nierzadko zna poszczególne procesy lepiej od osób za nie odpowiedzialnych.



◀ Ryzyka podejścia Low Code

Największe ryzyka dotyczą bezpieczeństwa, procesowania wrażliwych danych i zgodności z regulacjami, a więc właściwego zaprojektowania ich architektury, a także wydajności aplikacji niskokodowych.

Tych obszarów dotyczą też główne obawy wpływające na to, że nie wszystkie organizacje są otwarte na wdrożenie koncepcji Low Code.

Z drugiej strony, coraz bardziej naturalnym elementem procesu tworzenia aplikacji jest dziś automatyzacja testów. Takie podejście może być z powodzeniem stosowane także w przypadku środowisk niskokodowych. Automatyzacja nie neutralizuje wszystkich problemów, ale minimalizuje ryzyko braku skalowalności, niewystarczającego bezpieczeństwa czy niezgodności z obowiązującymi w danej firmie zasadami.

Niezależnie od wielkości organizacji, będąc osobą decyzyjną w sprawie wyboru koncepcji rozwoju aplikacji, rekomendowałbym wybranie rozwiązania, które pozwala na połączenie koncepcji No Code, Low Code i Pro Code. Zapewni ono fundament dla nowoczesnego podejścia do budowania aplikacji dla biznesu, również z wykorzystaniem możliwości, które daje nam generatywna sztuczna inteligencja.

Z dużą dozą prawdopodobieństwa zdecydowałbym też o wyborze rozwiązania dostarczanego w modelu chmurowym, ponieważ takie podejście zapewnia dodatkowo warstwę bezpieczeństwa i skalowalność na poziomie zasobów. Zawsze jednak warto na początku rozważyć rozwiązanie, które w konkretnej firmie będzie charakteryzować się niskim progiem wejścia.

◀ Najłatwiej wykorzystać istniejące technologie

Najniższy próg wejścia w rozwiązania niskokodowe istnieje w tych organizacjach, które korzystają już z rozwiązań, świetnie sprawdzających się jako podstawa dla środowisk No Code/Low Code.

Takimi rozwiązaniami są np. platformy Salesforce, ServiceNow czy Microsoft Power Apps. Ciekawym rozwiązaniem jest również skupiona wokół korporacyjnych zastosowań Low Code, platforma OutSystems.

Wychodzą one z różnych światów, ale zapewniają solidne możliwości w zakresie niskokodowego podejścia do tworzenia aplikacji biznesowych.

Salesforce historycznie wywodzi się z obszarów typowych dla rozwiązań wspierających zarządzanie relacjami z klientami. ServiceNow zaś z obszaru zarządzania usługami IT. Takie rozwiązania, raz wdrożone, obejmują m.in. kwestie z zakresu zarządzania łańcem korporacyjnym oraz mają spójny, bazowy model danych. Dużo łatwiej jest więc rozszerzyć skalę ich użycia na inne obszary, o ile pozwala na to funkcjonalność danej platformy. W przypadku środowiska ServiceNow można powiedzieć, że obecnie ponad 80% organizacji wykorzystuje tę platformę także poza obszarem IT, choćby do obsługi procesów HR, zakupowych, wsparcia posprzedażowego klientów, zarządzania ryzykiem, a ostatnio również z obszarem ESG oraz wsparciem end-to-end regulacji DORA w sektorze finansowym. Analogicznie jest w przypadku podejścia Low Code.

Skoro w danej firmie funkcjonuje już np. platforma ServiceNow, obsługująca wybrane procesy biznesowe, ryzyka oraz IT, udostępniająca centralny portal pracownika, za pomocą którego można zgłosić zapotrzebowanie na nowego laptopa, incydent bezpieczeństwa czy wniosek urlopowy, to dołożenie kolejnych procesów z wykorzystaniem modułów No Code / Low Code platformy jest bardzo proste i szybsze niż onboarding nowego rozwiązania i jego integracja.

Konsolidacja platform jest dobrym kierunkiem również z perspektywy kosztów, ale pamiętajmy, że jedna platforma nie jest w stanie pokryć wszystkich potrzeb organizacji, np. platforma ServiceNow sprawdza się świetnie jako „engagement layer”, który umożliwia zarządzanie procesami i usługami poprzez integrację z istniejącymi danymi i core systemami firmy, natomiast nie zastąpi bazowych System of Records.

◀ Praktyka dostawcy rozwiązań No Code / Low Code / Pro Code

W Accenture jestem liderem praktyki ServiceNow w Polsce, więc na bieżąco pracuję z tą właśnie platformą, ale mam też kontakt z innymi środowiskami, choćby na potrzeby zapewnienia niezbędnych integracji. Wraz z zespołem realizujemy projekty oparte na rozwiązaniach ServiceNow dla klientów z całego świata. Nie brakuje wśród nich ciekawych inicjatyw z obszaru No Code i Low Code.

Jednym z takich projektów było stworzenie aplikacji do zarządzania tożsamością i dostępem dla jednej z globalnych organizacji, wykorzystującej platformę ServiceNow w obszarach IT, HR, Procurement czy Finansach. Chodziło m.in. o centralne administrowanie tożsamościami, profilami, kontami użytkowników, audytem, certyfikacją, a także obsługą procesów zarządzania dostępami do wewnętrznych i zewnętrznych aplikacji cloud oraz on-prem, integrując to rozwiązanie również z procesami onboardingu i offboardingu.

Zrealizowany projekt umożliwił kontrolę nad uprawnieniami oraz kontami użytkowników w ustrukturyzowany sposób, z odpowiednim procesem akceptacyjnym, audytem oraz kwartalną certyfikacją dostępów do kluczowych środowisk. Brak kontroli nad tym obszarem niesie konsekwencje związane m.in. z bezpieczeństwem czy kosztami licencji.

Dla jednego z największych europejskich banków, wykorzystując mechanizmy platformy ServiceNow i synergię sięgającą od No Code po Pro Code, zbudowaliśmy rozwiązanie optymalizujące procesy zatrudnienia w kilkudziesięciu krajach, od momentu werbalnego przyjęcia oferty, poprzez generację, podpisywanie, zatwierdzanie końcowej umowy, badania lekarskie, dokumentację, po przygotowanie niezbędnych narzędzi, szkoleń i dostępuów dla rozpoczęcia pierwszego dnia pracy.

Z kolei dla jednej z firm produkcyjnych uruchomiliśmy system elektronicznego obiegu dokumentów i akceptacji umów. Rozwiązanie, które zintegrowaliśmy z popularnymi narzędziami e-signature typu DocuSign czy Adobe, zapewniło wsparcie dla procesów przechodzących przez kilka departamentów, aż do poziomu zarządu.

Do budowy takich aplikacji wykorzystujemy komponenty platformy ServiceNow No Code/Low Code, często bez potrzeby dodatkowego kodowania, a czasem z dobudowaniem skomplikowanych funkcjonalności, liczących tysiące linii kodu, np. na potrzeby zapewnienia złożonej i unikalnej integracji z systemami trzecimi. Co ważne, takie dedykowane rozszerzenia tworzymy tylko raz, a później możemy wykorzystywać je szeroko jedynie za sprawą prostej parametryzacji, tabel decyzyjnych i przy użyciu graficznego interfejsu platformy ServiceNow. Na tym właśnie polega podejście synerгии Pro Code z No Code/Low Code.

W ten właśnie sposób dla jednego z klientów podjęliśmy się digitalizacji procesów z obszaru Cloud Management, finansów, zaopatrzenia i obsługi dostawców/partnerów. W ramach zespołu projektowego zapewniliśmy rozwiązanie dla osób obsługujących zgłoszenia, płatności, użytkowników końcowych korzystających z portalu pracownika/dostawcy oraz automatyzacji w celu kontroli nad bazą CMDB i procesem bezobsługowej orkestracji usług chmurowych, w tym wirtualnych maszyn, czy operacji na poziomie firewall, z uwzględnieniem ryzyka operacyjnego. Niektóre z potrzebnych funkcjonalności, integracji a nawet procesów, musieliśmy zbudować od zera.

W ramach tego projektu jako Accenture skupiliśmy się na kilkudziesięciu kluczowych usługach. Poza departamentem IT, w pracach aktywnie uczestniczyli członkowie zespołów biznesowych firmy, dla której realizowaliśmy ten projekt. Te właśnie osoby, w kolejnych etapach rozbudowywały uruchomione wspólnie środowisko z wykorzystaniem podejścia No Code /Low Code, poprzez funkcjonalności i szablony zbudowane przez nas w początkowej fazie prac, zapewniając obsługę kolejnych, często całkiem nowych procesów już bez naszego udziału.

◀ Wsparcie od koncepcji po utrzymanie

W podobne projekty możemy zaangażować się praktycznie na każdym etapie. Często podpowiadamy, w jaki sposób warto organizować procesy, układamy Governance od poziomu zarządzania innowacjami, przez walidację techniczną oraz biznesową, po akceptację.

Oczywiście zapewniamy również wsparcie dla faktycznego procesu wytwarzania, wdrażania i utrzymania koncepcji No Code i Low Code, także w połączeniu z Pro Code. Automatyzujemy testy bezpieczeństwa,

wydajnościowe, użyteczności UI i wdrożenia, ale też układamy procesy zarządzania zmianą w organizacji. Często zapewniamy także rolę „mediatora”, która na bieżąco wspiera klienta na poziomie wewnętrznych dyskusji z działami IT i biznesem o ich potrzebach.

Dopiero w kolejnym kroku pomagamy dobrać odpowiednie do potrzeb rozwiązanie. Nie można też powiedzieć, że to właśnie platforma ServiceNow będzie najlepszą bazą do obsługi wszystkich procesów w każdej firmie. Przykładowo, w obszarze HR do przechowywania danych pracowników oraz zarządzania talentami lepiej sprawdzają się systemy, takie jak Workday, SAP SuccessFactors czy Cornerstone. Platforma ServiceNow z podejściem No Code/ Low Code może być jednak akceleratorem, który na bazie funkcjonalności zaszytych w systemach dziedzinowych pozwoli na przyspieszenie procesów i poprawę doświadczeń ich użytkowników.



Low Code sprawdza się także w złożonych procesach

Wiele osób niepotrzebnie odrzuca możliwość wykorzystania koncepcji Low Code do obsługi bardziej złożonych procesów. Tymczasem, niezależnie od tego, czy w procesie istnieją 4 kroki, czy 40, czy rozwiązanie powinno być dostępne z poziomu przeglądarki internetowej, czy również w ramach aplikacji mobilnej, to taki proces można zautomatyzować z użyciem podejścia Low Code. Wówczas jedynie rośnie prawdopodobieństwo tego, że w ramach danej platformy nie istnieją wszystkie potrzebne funkcjonalności, komponenty, integracje i należy je dołożyć, działając w ramach koncepcji Pro Code.

Nie należy też wdrażać podejścia Low Code na siłę. Niektóre organizacje mają m.in. własne standardy aplikacji biznesowych podążające za technologiami niewspierającymi koncepcji niskokodowych. W takich przypadkach bez ważnego powodu biznesowego przenoszenie istniejących, poprawnie działających aplikacji krytycznych do platform Low Code nie ma sensu.

W Accenture nie rekomendujemy też przepisywania funkcjonalności kluczowych, dedykowanych systemów do środowisk niskokodowych.

Uważamy, że najlepsza jest synergia takich rozwiązań. Podobnie jest w przypadku platform Salesforce i ServiceNow. Pierwsza z nich zapewnia świetne wsparcie procesów sprzedażowych. ServiceNow ma zaś usprawnić procesy, które następują po zawarciu transakcji sprzedażowej, również w ramach wsparcia technicznego dla sprzedanych usług oraz produktów. Znow, podstawą takiej synergii jest przemyślana integracja. Integracja tym łatwiejsza, że wyróżnikiem nowoczesnych rozwiązań niskokodowych są predefiniowane konektory zapewniające integrację z naprawdę wieloma wykorzystywanymi na świecie rozwiązaniami biznesowymi. W przypadku ServiceNow jest to ok. 300 dostępnych konektorów pozwalających np. na dodanie obsługi podpisów

DocuSign, pozyskania danych o klientach z bazy Salesforce, utworzenia zgłoszenia szkody komunikacyjnej na platformie GuideWire czy wykorzystania ChatGPT w ramach wbudowanego wirtualnego agenta. Oczywiście, dla pewnych, unikalnych scenariuszy predefiniowane konektory okazują się niewystarczające. Wówczas możemy je rozszerzyć poprzez dodanie własnych akcji/kodu, który uruchomi dodatkowe zadanie lub wykona skomplikowaną transformację danych wymaganą przez poszczególny proces.



Paweł Pietryka

Cloud First ServiceNow

Practice Lead w Accenture w Polsce

Jaki powinien być Low Code do zastosowań korporacyjnych?

Koncepcja Low Code otwiera nowe możliwości rozwoju aplikacji biznesowych. Demokryzacja podejścia do tworzenia oprogramowania stwarza jednak dodatkowe ryzyka m.in. w zakresie jakości kodu, wykorzystania danych czy integracji. Odpowiedzialne zastosowanie rozwiązań niskokodowych wymaga zatem wypracowania procedur i polityk, które pozwolą na zapewnienie wysokiego poziomu bezpieczeństwa oraz zgodności z przepisami.

Choć podejście Low Code wręcz zachęca zaangażowanych użytkowników do samodzielnego eksperymentowania i budowania aplikacji odpowiadających na konkretne, nawet chwilowe potrzeby użytkowników, **to w realiach korporacyjnych należy precyzyjnie oddzielić eksperymenty od systemów faktycznie używanych na tzw. produkcji.** Platforma, która ma stać się podstawą tworzenia, wykorzystywanych w codziennej pracy, aplikacji niskokodowych musi spełniać wymagania typowe dla środowisk korporacyjnych – w szczególności w zakresie bezpieczeństwa i zgodności z regulacjami.

Niezbędne staje się zagwarantowanie, że zarówno deweloperzy – profesjonaliści oraz użytkownicy biznesowi bezpośrednio zaangażowani w tworzenie aplikacji – będą przestrzegać obowiązujących w organizacji reguł i standardów, a także procedur bezpieczeństwa. Podobny nadzór powinien istnieć również w odniesieniu do samych rozwiązań budowanych na bazie koncepcji niskokodowych – rozwiązań tworzonych w myśl zasad Low Code, a więc szybciej i w sposób angażujący szerszą grupę osób, które powinny podlegać szczególnej kontroli w zakresie zapewnienia zgodności z regulacjami. To zaś **wymaga przemyślenia i właściwego uporządkowania procesów, odpowiednio do potrzeb i możliwości konkretnej firmy.** Jest to o tyle istotne, że choć podejście Low Code opiera się na wykorzystaniu automatyzacji, to jednak dotyczy ona przede wszystkim tworzenia kodu aplikacyjnego.

Low Code z funkcjami korporacyjnymi

Na rynku dostępne są dziś rozmaite środowiska pozwalające na tworzenie oprogramowania w reżimie Low Code. Obok stosunkowo prostych narzędzi, istnieją

rozwiązania bardzo złożone, wspierające także inne obszary działalności największych organizacji. **W realiach biznesowych warto skupić się jednak głównie na platformach wyposażonych w mechanizmy mające na celu zapewnienie kontroli w obszarze uprawnień, dostępu, funkcjonalności oraz zgodności z regulacjami.**

Dojrzałe środowiska Low Code do zastosowań korporacyjnych w sposób natywny gwarantują możliwość kontrolowania najważniejszych aspektów związanych z procesem wytwórczym oprogramowania, a także poszczególnych etapów tegoż procesu. Bezpośrednio w ramach takich platform można, przykładowo, określić odpowiedzialność za weryfikację pomysłów, zatwierdzanie kolejnych etapów rozwoju aplikacji. W analogiczny sposób można również osadzić środowiska Low Code w ramach istniejących w organizacji polityk oraz dobrych praktyk związanych z wykorzystaniem zasobów IT, usług chmurowych czy bieżącym funkcjonowaniem zespołu IT.

Odpowiedzialność plus zaangażowanie

Warto podkreślić, że ogromne znaczenie dla właściwego osadzenia platformy Low Code w ramach firmy ma właściwe zrozumienie roli oraz możliwości podejścia niskokodowego. **Ważne jest, aby wszystkie osoby zaangażowane w proces tworzenia aplikacji biznesowych w modelu Low Code – programiści, użytkownicy i analitycy biznesowi, a także właściciele poszczególnych projektów – byli zaangażowani w proces wdrożenia takiego środowiska na możliwie wczesnym etapie.** Dzięki temu, w dłuższej perspektywie, będą w stanie bardziej trafnie identyfikować potrzeby biznesowe możliwe do obsłużenia w ramach platformy niskokodowej.

Wdrożenie koncepcji Low Code nierzadko wymaga też zmiany przyzwyczajeń i ograniczenia apetytów na kustomizacje i indywidualne dostosowania kodu aplikacyjnego. Istota tego podejścia zakłada bowiem maksymalne wykorzystanie gotowych elementów, predefiniowanych lub zbudowanych na potrzeby wcześniejszych projektów.

Low Code pod kontrolą

Dzisiejsze realia w zakresie cyberbezpieczeństwa wymuszają podjęcie kroków mających na celu zwiększenie odporności całego środowiska IT na ewentualne zagrożenia. W sposób oczywisty dotyczy to aplikacji – również tych tworzonych w reżimie niskokodowym. Co więcej, możliwa za pomocą koncepcji Low Code **demokratyzacja podejścia do tworzenia oprogramowania stwarza dodatkowe ryzyka na poziomie jakości kodu, sposobu wykorzystania danych oraz integracji z innymi aplikacjami, bazami lub źródłami danych.**

Wszystko to przekłada się na konieczność właściwego zaprojektowania uprawnień w zakresie dostępu do poszczególnych funkcji środowiska Low Code oraz danych, tak aby ograniczyć ewentualne ryzyka naruszenia bezpieczeństwa danych lub infrastruktury. Są to kwestie, które powinny pozostać w domenie zespołu IT. **Rolą doświadczonych programistów powinno być m.in. opracowanie i egzekwowanie właściwej architektury aplikacji, weryfikacja funkcjonalności tworzonej w ramach platformy Low Code, a także kontrola dostępu, ocena bezpieczeństwa oraz sposobu i zakresu wykorzystania danych oraz integracji.**

Zapewnienie bieżącego nadzoru nad procesem rozwoju aplikacji Low Code i precyzyjny podział odpowiedzialności za kwestie funkcjonalne, techniczne i operacyjne takich rozwiązań jest niezbędne również przez wzgląd na potencjalnie sprzeczne interesy poszczególnych grup użytkowników platform niskokodowych.

O ile bowiem użytkownikom biznesowym i właścicielom konkretnych procesów zależy może na jak najszybszym dostarczeniu możliwie najszerszej funkcjonalności, o tyle zespół IT powinien, działając w swoim obszarze odpowiedzialności, pilnować zgodności i bezpieczeństwa, racjonalizując zamiary deweloperów wywodzących się z zespołów biznesowych.



Realne scenariusze użycia i realne ryzyka

Aby dobrze zrozumieć znaczenie kwestii związanych z zapewnieniem bezpieczeństwa i zgodności z regulacjami, warto przyjrzeć się typowym obszarom wykorzystania aplikacji niskokodowych.

— Strony internetowe i aplikacje webowe

podjęcie Low Code znacząco usprawnia budowanie portali oraz aplikacji internetowych wspierających różnego rodzaju interakcje z klientami. Rozwiązania takie znajdują się więc w pierwszej linii potencjalnych ataków. Nierzadko pośredniczą też w wymianie wrażliwych danych, związanych, przykładowo, z obsługą zamówień i płatności.

— Aplikacje biznesowe

zapewniające funkcjonalności wymagane do realizacji konkretnych procesów biznesowych lub stanowiące fundament dla oferowanych usług i produktów. Z racji na swoją istotę, mniej narażone na zagrożenia zewnętrzne, jednak często stanowią o możliwości realizacji procesów istotnych dla bieżącego funkcjonowania danej organizacji. Warto zaznaczyć, że rozwiązania Low Code najczęściej nie znajdują zastosowania przy budowaniu najbardziej kluczowych systemów biznesowych – ale są wykorzystywane na potrzeby rozszerzenia możliwości istniejących systemów.



— Aplikacje mobilne

Łączące możliwości interfejsów webowych i dedykowanych narzędzi biznesowych – zarówno tych, dostępnych dla klientów, jak i tych, stosowanych na wewnętrzne potrzeby organizacji.

— Automatyzacja obiegu informacji

Podjęcie Low Code sprawdza się jako sposób na tworzenie zautomatyzowanych, nierzadko punktowych narzędzi wspierających gromadzenie danych oraz tworzenie raportów i dostarczanie ich właściwym użytkownikom. Takie rozwiązania cyfryzujące specyficzne procesy biznesowe mają zatem ogromny wpływ na podejmowanie decyzji biznesowych. Analogicznie, koncepcja Low Code znajduje zastosowanie również na potrzeby zapewnienia integracji i wymiany danych pomiędzy konkretnymi aplikacjami, mikrousługami oraz źródłami danych, takimi jak: bazy danych, systemy biznesowe, ale też czujniki i urządzenia brzegowe wchodzące w skład infrastruktury internetu rzeczy.

Potencjał biznesowy koncepcji Low Code decyduje o wysokim zainteresowaniu takimi rozwiązaniami – także w najbardziej regulowanych branżach i firmach przetwarzających wysoce wrażliwe dane. Faktem jest, że podejście niskokodowe zakłada pewne uproszczenia procesu tworzenia aplikacji. Nie stoi to jednak na przeszkodzie do wykorzystania modelu Low Code w wielu obszarach biznesowych. Co więcej, **przy zachowaniu pewnych zasad, rozwiązania niskokodowe świetnie uzupełnią dotychczasowe podejście do rozwoju środowiska aplikacyjnego wielu przedsiębiorstw, otwierając przy tym dodatkowe możliwości tworzenia innowacji biznesowych.**



Piotr Waszczuk



6 przykładów użycia rozwiązań niskokodowych

Co łączy różne przykłady zastosowania
podejścia niskokodowego?

Motywacje stojące za wykorzystaniem podejścia niskokodowego są różne, w większości sprowadzają się m.in. do poprawy szybkości realizacji projektów, ograniczenia ich kosztów czy możliwości zaangażowania użytkowników biznesowych wprost do tworzenia aplikacji, z których będą następnie korzystać.

**Doświadczenia i skutki zastosowania podejścia Low Code
w różnych organizacjach warto pokazać na przykładach.**

OutSystems

w DNB Bank Polska SA

Wdrożenie platformy OutSystems w DNB Bank Polska SA było związane z unowocześnieniem systemu CRM. Działające dotychczas rozwiązanie nie pozwalało na poprawę ani w dłuższej perspektywie utrzymanie odpowiedniego poziomu doświadczenia korporacyjnych klientów. Jednocześnie założeniem modernizacji było przeprowadzenie projektu bez potrzeby zmian i zakłócania pracy systemu głównego banku, który jest rozwiązaniem legacy i nie było planów jego zmiany – przy jednoczesnej pełnej integracji.

Udało się to dzięki wykorzystaniu platformy Low Code, na bazie której firma C&F zbudowała nowe rozwiązanie CRM, osiągając to w ciągu niespełna 6 miesięcy, to jest w czasie o połowę krótszym, niż pierwotnie zarezerwowano na tę zmianę. Podstawowym parametrem, wyznaczającym istotę poprawy, było przyspieszenie o 70% wskaźnika time-to-market usług i produktów dostarczanych w ramach CRM.

OutSystems przyspiesza proces tworzenia aplikacji poprzez uproszczenie i automatyzację wielu aspektów rozwoju oprogramowania, co nie tylko przynosi korzyści zespołom IT pod względem produktywności i zwinności, lecz także lepiej dostosowuje IT do celów biznesowych. Szybka implementacja funkcjonalności oraz łatwość zmian skutkują szybszym feedbackiem zwrotnym, co w podejściu iteracyjnym pozwala wyeliminować nieścisłości już na starcie. Ponadto skupiamy się na implementacji funkcjonalności biznesowych dla użytkownika końcowego, resztę zapewnia platforma OutSystems. Nawet w przypadku pierwszych projektów poszczególne ich fazy są dostarczane szybciej, wyprzedzając harmonogram i nierzadko wymuszając przestoje, aby zczekać, aż zmiany w systemach współpracujących zostaną wykonane

Michał Witek

Solution Architect
w firmie C&F.

Platforma OutSystems stała się zarazem podstawowym środowiskiem dostarczania kolejnych aplikacji zintegrowanych z wieloma innymi systemami i źródłami danych w banku, ponieważ zapewniają one w ocenie banku zdecydowanie lepsze doświadczenie klientom-użytkownikom. Nowy CRM jest również kluczowym źródłem danych dla głównego systemu banku.

Konrad Jęczeń, CIO DNB Bank Polska określił następujące kryteria wyboru platformy OutSystems: liczne referencje sektorowe oraz pozycja lidera w rankingach, elastyczność i szybkość rozwoju aplikacji, łatwość wykorzystania platformy nawet przy ograniczonych zasobach deweloperskich oraz integracji z różnymi systemami i źródłami danych. Cenne były też referencje dotyczące jakości doświadczenia użytkowników, a także dostępność partnera dysponującego doświadczeniem wdrożeniowym.

||
Nowe rozwiązanie
CRM w 6 miesięcy.

Czas wykonania o połowę
krótszy niż planowano.

||

OutSystems jako platforma Low Code i C&F jako partner doświadczony w modernizacji corowych/legacy aplikacji niosą specjalistyczne umiejętności i wiedzę, które mogą znacząco zmniejszyć złożoność, ryzyko i koszty modernizacji starych systemów. Jak w przypadku każdej technologii, doświadczenie jest bezcenne. Zawsze doradzamy klientom wspólne rozpoczęcie przygody z OutSystems, dzieląc się najlepszymi praktykami i błędami, które kiedyś popełniliśmy. W miarę rozwoju projektu klient nabywa doświadczenia oraz pewności w działaniu – to moment, w którym może zacząć działać na własną rękę lub pozostać we współpracy z partnerem, który będzie w stanie utrzymać jakość, zapewnić support oraz zaalokować zasoby projektowe w najbardziej efektywny sposób

Michał Witek

Solution Architect
w firmie C&F.



Mendix w Siemensie

Kiedy globalny potentat Siemens w dziedzinie inżynierii, oprogramowania dla przemysłu, elektrotechniki i elektroniki przejął platformę Mendix, jednym z powodów była potrzeba usprawnienia i udoskonalenia własnego środowiska i procesów wytwarzania oprogramowania. **Siemens znacząco zredukował czas rozwoju, zwiększył współpracę między zespołami biznesowymi i IT oraz stworzył innowacyjne aplikacje dla swoich klientów.**

Przykładem zastosowania Low Code do wewnętrznych celów było wytworzenie aplikacji do obsługi procesu planowania sprzedaży w Siemens Digital Industries. Zespół sprzedażowy w tym oddziale Siemens potrzebował nowego podejścia do optymalizacji procesu zintegrowanego planu krajowego (ICP). Proces ICP polega na uspołnieniu celów sprzedażowych i rynkowych, na podstawie których poszczególne obszary określają własne strategie i plany działania.

Dotychczasowe rozwiązanie opierało się na arkuszach kalkulacyjnych, co czyniło proces niewygodnym i niewydajnym. Informacja biznesowa była gromadzona w różnych formatach i w wielu systemach. Co więcej, poszczególne departamenty sprzedażowe w oddziałach zaczęły tworzyć własne, luźno lub wcale niezintegrowane plany, wykorzystując własne metodyki i pomysły. Trudno w tej sytuacji było mówić o spójności danych i budowaniu na ich bazie współpracy.

Podstawą do wypracowania nowego, wspólnego rozwiązania stała się platforma Low Code Mendix, której właścicielem Siemens jest od 2018 roku.

Zaplanowano stworzenie aplikacji, która miała realizować proces ICP. Podejście Agile połączyło zespoły obejmujące różne funkcje, co w dalszym kroku pozwoli na przyspieszenie cyfryzacji we wszystkich obszarach Siemens Digital Industries Software.

W roku 2020 przystąpiono do tworzenia rozwiązania na podstawie platformy Low Code, dostępnej już za pośrednictwem wewnętrznej platformy rozwiązań biznesowych Siemens Xcelerator. **Zespół skierowany do tego zadania liczył zaledwie dwie osoby bez doświadczenia z platformą Mendix i trzech programistów, również bez znajomości Mendix, pod kierownictwem programisty, który takie doświadczenie posiadał.**

W pierwszej kolejności zespół ten przeszedł 2-tygodniowe szkolenie. Przystawanie narzędzi i zasad działania Mendix od podstaw następowało bardzo szybko – poza wspomnianym projektem aplikacji ICP, po 6 miesiącach jego członkowie potrafili już wytwarzać interfejsy API do systemu SAP. **Projekt POC dla aplikacji ICP został przeprowadzony przez dwóch zaledwie członków zespołu w ciągu tygodnia, a w ciągu dwóch miesięcy powstała aplikacja, która następnie została udostępniona 300 użytkownikom.**

Nowa aplikacja pozwoliła także wytworzyć prawdziwe role w zakresie zarządzania danymi i wyłonić właścicieli danych i procesów ich przetwarzania. Zapewnia też kontrolę wersji zestawów danych, wielopoziomowe zarządzanie dostępem do danych oraz ich jakością w całej organizacji. Proces IC, którego reprezentację stanowi nowa aplikacja, stał się ponadto żywym, angażującym obszarem – użytkownicy dostrzegli korzyści, jakie dostarcza, szczególnie w kontekście możliwości precyzyjnego prognozowania i planowania działań oraz podejmowania decyzji.

||

Projekt POC dla aplikacji ICP został przeprowadzony przez dwóch zaledwie członków zespołu w ciągu tygodnia.

W ciągu dwóch miesięcy powstała aplikacja, która następnie została udostępniona 300 użytkownikom.

||



Appian wsparciem dla usług 5G, kanadyjskiego operatora telekomunikacyjnego

Telus to kanadyjski operator telekomunikacyjny, obsługujący ponad 11 mln klientów. Od roku 2018 firma przygotowywała się do udostępnienia usług opartych na technologii 5G. **Uruchomienie nowej technologii miało dokonać się szybko i na pełną skalę, a zarazem spełnić podstawowe wymogi do zarządzania zmienionymi parametrami ruchu IP, który dzięki 5G generować mieli użytkownicy.**

Audyt działających systemów wykazał, że mogą nie sprostać nowym wyzwaniom skali, dostępności i wydajności. Według szacunków Telus, oddanie do użytku usług 5G przynieść miało nawet 10-krotne zwiększenie potrzeb w zakresie tych podstawowych parametrów, w stosunku do potrzeb, jakie generowała obsługa klientów w dotychczasowych technologiach telekomunikacyjnych. Firma uznała, że potrzebuje nowej, specjalnie zaprojektowanej do zarządzania i obsługi platformy, która przejąć miałaby wszystkie zadania rozproszone pomiędzy dotychczas wykorzystywane systemy i aplikacje.

Na bazie platformy Appian, Telus zbudował rozwiązanie Maestro do obsługi wszystkich usług realizowanych przez 5G. Obecnie zarządza ponad 10 tys. niezbędnych działań i funkcji, przejmując je od 11 równoległych działających dotychczas systemów. **Wdrożenie rozwiązania opartego na Low Code pozwala także na rozbudowę i szybkie dostarczanie kolejnych usług i produktów opartych na 5G. Czas ich tworzenia w nowym środowisku jest bowiem 10 razy krótszy.**

||

Wdrożenie rozwiązania opartego na Low Code pozwala także na rozbudowę i szybkie dostarczanie kolejnych usług i produktów opartych na 5G.

—

Czas ich tworzenia w nowym środowisku jest bowiem 10 razy krótszy.

||

Oracle Apex w Grupie VAN

Grupa VAN zatrudnia 700 pracowników i współpracuje z grupą ponad 3000 przewoźników, obsługując w Polsce i krajach Europy Środkowo-Wschodniej kilka tysięcy klientów. Przychody na poziomie 1 mld zł kwalifikują ją w gronie 10 największych firm branży TSL w Polsce.

Wyzwania rynku w związku z cyfryzacją – konieczność dostosowania do oczekiwań klientów, konkurencja, w szczególności zagrożenie ze strony platform – spowodowały, że firma postanowiła zbudować nowy, centralny system do obsługi biznesu. **Ewoluuując od rozwiązań clipperowych z lat 90., przez Oracle Forms, firma przeszła do modelu Low Code, ekspresowego budowania aplikacji na bazie Oracle APEX.** Decyzja o budowie w tym podejściu nowego systemu TMS wynikała z konieczności dopasowania rozwiązania do biznesu. Nowa wersja systemu e-VAN umożliwia budowanie aplikacji webowych, szybki development, a jednocześnie rozwój i utrzymanie nie wymagają dużych nakładów finansowych. Współpraca z partnerem – Pretius Low Code – zdecydowała o powodzeniu wdrożenia, ale także upowszechnienia podejścia w całej firmie.

W ciągu 6 miesięcy od wdrożenia produkcyjnych modułów „Rozliczenia” oraz „Windykacja” nastąpił znaczny spadek wartości faktur „zagrożonych”. Oszczędności przy zakupie i utrzymaniu infrastruktury wyniosły ok. 20% i kosztów licencji na serwery aplikacyjne o 10% w skali roku.

Korzyści jest więcej. Obejmują zwiększenie szybkości wytwarzania oprogramowania oraz określenie nowych ról w tym procesie, poprawę jakości dzięki zastosowaniu inżynierii oprogramowania, częstsze, ale nie wpływające na pracę aktualizacje oprogramowania i środowiska. Nowy system stwarza też możliwości dalszej cyfryzacji środowiska pracy i procesów w Grupie VAN. Oznacza to kolejne korzyści dla firmy w postaci przyspieszenia procesów, zwiększenia konkurencyjności, oszczędności.

||

Firma przeszła do modelu Low Code ekspresowego budowania aplikacji na bazie Oracle APEX.

W ciągu 6 miesięcy od wdrożenia produkcyjnych modułów „Rozliczenia” oraz „Windykacja” nastąpił znaczny spadek wartości faktur „zagrożonych”.

||

Microsoft Power Platform przy budowie tunelu w Szwajcarii

Firma budowlana, realizująca kontrakt na 300-metrowy tunel w Szwajcarii, którego budowa wymaga drążenia i wysadzania skały, posługuje się ustaloną metodyką i procedurami inżynierskimi. **Przed wszystkim budowa tunelu wymaga przestrzegania sekwencji działań i prac w ramach powtarzalnego procesu, odcinek za odcinkiem powstającej inwestycji.**

Śledzenie postępów w tym procesie wymagało wypełniania różnego rodzaju formularzy papierowych, gromadzących różnego typu informacje techniczne i organizacyjne (na trzy zmiany przypadało 6 różnych papierowych formularzy). Następnie informacja z formularzy była przetwarzana w arkuszach kalkulacyjnych. **Ręczny proces zbierania i przetwarzania informacji pochłaniał dużo czasu, a co gorsza – powodował błędy.** W wielu wypadkach informacja nie była przetwarzana dostatecznie szybko, a pomyłki wymagały ponownego sprawdzania i weryfikowania danych wejściowych. W efekcie menedżerowie i zespoły odpowiedzialne za kolejne etapy nie dysponowały aktualną ani pewną informacją.

Naturalnym rozwiązaniem było uporządkowanie i scyfrizowanie procesu, połączone z automatyzacją zbierania i przetwarzania danych. Żadne z gotowych rozwiązań na rynku nie odpowiadało oczekiwaniom zespołu. Zdecydowano więc o stworzeniu własnego rozwiązania; zostało to poprzedzone analizą procesu budowlanego pod kątem przepływu danych. Na tej podstawie rozpoczęto pracę nad rozwiązaniem, w której udział wzięło 6 osób, w tym tylko 2 osoby o kwalifikacjach Citizen Developerów. Toczyła się ona równolegle do prowadzonych prac budowlanych.

W wyniku tego zbudowano mobilną aplikację obsługującą wprowadzanie danych na kolejnych wskazanych etapach budowy, przekazywanie do dedykowanej bazy, a następnie jej przetwarzanie w określonym zakresie i przedstawianie w formie graficznej. Wszystkie prace wykonano w środowisku Microsoft Power Platform, korzystając m.in. z predefiniowanych tam konektorów źródeł danych oraz interfejsu graficznego wbudowanego w PowerApps.

Do generowania analiz i raportów oraz wykonania zapytań business intelligence aplikacja wykorzystuje Power BI, który pobiera potrzebne dane i zestawia je stosownie do potrzeb, w tym wizualizacji, z Microsoft Lists. **Użytkownicy – zespół pracujący na kontrakcie – ma w efekcie narzędzie umożliwiające na żywo dostęp do przetworzonej informacji o postępach prac, własnych zadaniach i efektywności prac.**

||

Naturalnym rozwiązaniem było uporządkowanie i scyfrizowanie procesu, połączone z automatyzacją zbierania i przetwarzania danych.

Zbudowano mobilną aplikację obsługującą wprowadzanie danych na kolejnych wskazanych etapach budowy, przekazywanie do dedykowanej bazy, a następnie jej przetwarzanie w określonym zakresie i przedstawianie w formie graficznej.

||



OutSystems wsparciem dla firmy farmaceutycznej

Firma farmaceutyczna, organizująca ponad 400 spotkań i konferencji dla około 20 tys. lekarzy, farmaceutów i decydentów spółek farmaceutycznych, jest zobligowana do raportowania przejrzystości swoich działań Europejskiej Federacji sektorów i stowarzyszeń farmaceutycznych (EFPIA).

Firma potrzebowała rozwiązania do zarządzania komunikacją i relacjami z tą grupą w czasie rzeczywistym, w tym zbierania i udostępniania odpowiednim organom niezbędnej informacji. Co więcej, model i procesy oparte były w większości na niescyfryzowanej, papierowej formie obiegu dokumentów i zadań.

Szybkie dostarczenie takiego rozwiązania, które ponadto rokowałoby możliwość dalszej rozbudowy i wprowadzania zmian, wymagało sięgnięcia po metody niskokodowe.

Powstał jeden, zintegrowany system był w stanie zastąpić wiele procedur i aplikacji dotychczas niezbędnych do realizacji podstawowego procesu obsługi zaproszonych gości oraz właściwego raportowania o relacjach do EFPIA. Ta mozaika powstawała na przestrzeni dekady i w znacznym stopniu utraciła możliwość zmiany i realizacji swoich zadań. Nowe rozwiązanie zbudowano natomiast w ciągu pół roku na platformie OutSystems. Jej wytworzenia i wdrożenia w spółce podjęła się firma C&F, partner OutSystems. System przejmuje także stopniowo kolejne aplikacje i procedury, związane z autoryzacją wydatków, dostępem do informacji czy nawet wydrukami. Tym samym wiele procesów w firmie zostało dzięki temu dodatkowo scyfryzowanych w ramach spójnej, zintegrowanej i przyjaznej do korzystania aplikacji.

||

Szybkie dostarczenie rozwiązania do zarządzania, które ponadto rokowałoby możliwość dalszej rozbudowy i wprowadzania zmian, wymagało sięgnięcia po metody niskokodowe.

Powstał jeden, zintegrowany system był w stanie zastąpić wiele procedur i aplikacji dotychczas niezbędnych do realizacji podstawowego procesu obsługi zaproszonych gości oraz właściwego raportowania o relacjach do EFPIA.

Podsumowanie

Powyższe przykłady pokazują, że umiejętnie wykorzystana koncepcja Low Code zapewnia wymierne korzyści biznesowe, również w obszarach mocno wykraczających poza ułatwienie procesu tworzenia oprogramowania.

Tym samym, platformy Low Code dobrze **wpisują się w aktualne potrzeby** dotyczące, przykładowo, wysokiej efektywności i dynamiki tworzenia aplikacji, modernizacji rozwiązań typu legacy i wdrażania innowacji na styku z klientami.



Szymon Augustyniak

nAxiom

powered by **OPTeam**

Uniwersalna i elastyczna platforma **low-code**

Zmieniaj pomysły w gotowe rozwiązania

nAxiom to platforma low-code służąca do szybkiego budowania aplikacji biznesowych metodą drag and drop, umożliwiającą optymalizację i robotyzację procesów.

Platforma low-code nAxiom to:

- szybkie wdrożenie funkcjonalności niedostępnych w użytkowanych systemach
- implementowanie innowacyjnych pomysłów i procesów
- spójne i zarządzalne środowisko aplikacyjne zamiast rozproszonych i nieefektywnych rozwiązań
- szybkie modyfikacje procesów w reakcji na zmiany w otoczeniu biznesowym




Pobierz e-booka i dowiedz się więcej



nAxiom



naxiom.com

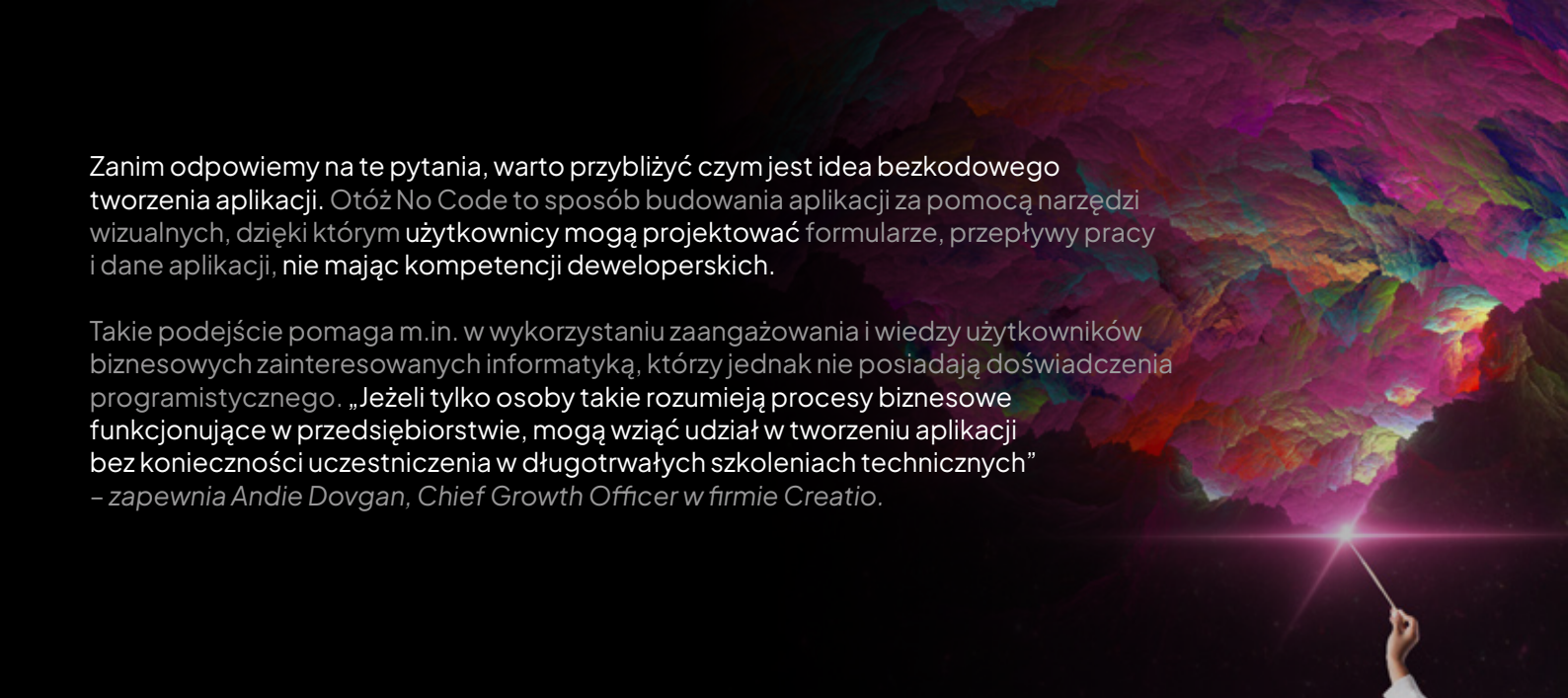


jakie rozwiązania warto budować

za pomocą platform No Code,
a jakich lepiej nie?

Branża oprogramowania znajduje się obecnie w fazie rewolucji spowodowanej szybkim wzrostem popularności różnych narzędzi wspierających programowanie, w tym platform typu No Code. Dzięki nim tworzeniem aplikacji mogą zajmować się także osoby niemające wiedzy z dziedziny programowania.

W jakich zatem obszarach takie podejście się sprawdza, a w których niekoniecznie?



Zanim odpowiemy na te pytania, warto przybliżyć czym jest idea bezkodowego tworzenia aplikacji. Otóż No Code to sposób budowania aplikacji za pomocą narzędzi wizualnych, dzięki którym użytkownicy mogą projektować formularze, przepływy pracy i dane aplikacji, nie mając kompetencji deweloperskich.

Takie podejście pomaga m.in. w wykorzystaniu zaangażowania i wiedzy użytkowników biznesowych zainteresowanych informatyką, którzy jednak nie posiadają doświadczenia programistycznego. „Jeżeli tylko osoby takie rozumieją procesy biznesowe funkcjonujące w przedsiębiorstwie, mogą wziąć udział w tworzeniu aplikacji bez konieczności uczestniczenia w długotrwałych szkoleniach technicznych” – zapewnia Andie Dovgan, Chief Growth Officer w firmie Creatio.

Jakie rozwiązania można budować z użyciem platform No Code?

Podejście No Code może odgrywać znaczącą rolę w transformacji cyfrowej konkretnych organizacji. Typowe przypadki wykorzystania tej koncepcji obejmują:

- automatyzację powtarzalnych operacji, zmniejszającą obciążenie pracowników i poprawiającą efektywność kluczowych procesów. Firmy mogą również wykorzystywać No Code do tworzenia niestandardowych aplikacji, które usprawniają ich operacje — takie jak automatyzacja powtarzalnych zadań lub tworzenie spersonalizowanych pulpitych nawigacyjnych i raportów w celu monitorowania ich wydajności;

- modernizację warstwy obsługi starszych systemów biznesowych;

- tworzenie rozwiązań cyfrowych, które poprawiają jakość obsługi klienta (np. aplikacje mobilne, portale klienckie czy chatboty);

- usprawnienie procesów wewnętrznych, takich jak HR, finanse, marketing oraz IT, m.in. na poziomie przepływu pracy, co również poprawia efektywność i zmniejsza obciążenie zespołów wsparcia;

- zapewnienie integracji starszych systemów z nowymi koncepcjami i technologiami, takimi jak chmura obliczeniowa, AI czy internet rzeczy.

Rozwiązania No Code mogą też sprawdzić się w szeroko pojętej analityce danych, wspierając firmy w tworzeniu rozwiązań, które pasują do ich konkretnych potrzeb — takich jak wizualizacja danych, eksploracja danych i uczenie maszynowe. Często ułatwiają również integrację narzędzi analizy danych z innymi systemami biznesowymi, np. CRM, ERP czy platformy handlu elektronicznego, zapewniając wgląd w czasie rzeczywistym we wszystkie aspekty ich działalności.

Integracja taka odnosi się też do łączenia różnych narzędzi i usług oprogramowania. Pozwala to automatyzować procesy i łatwo udostępniać dane bez pisania kodu. Co ważne, zapewnienie takiej integracji na bazie rozwiązań No Code jest zwykle możliwe za pomocą prostego kreatora wizualnego. W analogiczny sposób można także budować nowoczesne, przyjazne dla użytkownika interfejsy dla starszych aplikacji, ułatwiając w ten sposób ich użytkowanie i poprawiając wrażenia użytkownika, bez konieczności przeprowadzenia złożonej migracji czy modernizacji systemów typu legacy. W tym kontekście można również łączyć starsze aplikacje z nowymi technologiami, a także dodawać nowe funkcjonalności związane m.in. z obsługą technologii mobilnych czy analizy danych.

Szeroki wachlarz zastosowań sektorowych

Reasumując, rozwiązania No Code mogą być wykorzystywane w różnych branżach i na różne sposoby. Przykładowo, w bankowości i usługach finansowych często służą do tworzenia niestandardowych aplikacji do zarządzania ryzykiem, wykrywania oszustw i zgodności z przepisami. Dodajmy, że badanie przeprowadzone przez Forrester Research przewiduje, że do końca 2023 roku No Code będzie wykorzystywany w 70% aplikacji rozwijanych w tym sektorze.

Kolejną branżą, w której szerokie zastosowania znajdują technologie bezkodowe, jest przemysł. Tu platform No Code używa się do tworzenia aplikacji wspierających specyficzne procesy z zakresu zarządzania zapasami, łańcuchem dostaw czy utrzymania predykcyjnego. Z kolei branża opieki zdrowotnej może wykorzystywać No Code do tworzenia niestandardowych systemów EHR i usprawniania opieki nad pacjentem, zaś branża detaliczna — do rozwoju unikalnych funkcjonalności e-commerce.

Jakie ograniczenia ma No Code?

Należy jednak pamiętać, że bazujące wyłącznie na warstwie wizualnej platformy bezkodowe nie umożliwiają dodawania lub zmiany kodu źródłowego budowanych aplikacji i mogą opierać się jedynie na mechanizmach wbudowanych w konkretne platformy. Nie stosuje się ich zatem do tworzenia złożonych rozwiązań o znaczeniu krytycznym. Poza tym, jak wskazują praktycy, ograniczeniem platform No Code często staje się ich wysoka standaryzacja, zwykle ograniczony interfejs użytkownika oraz domyślne szablony aplikacyjne. Podobnie, jeśli chodzi o funkcjonalność — problem zaczyna się, gdy potrzebujemy konkretnej funkcji, której w wybranej platformie nie ma. Kolejnym ograniczeniem może być migracja z jednej platformy No Code na drugą.

Co istotne, chociaż przeznaczone do zastosowań korporacyjnych platformy No Code zapewniają generalnie wysoki poziom bezpieczeństwa, to najczęściej gwarantują jedynie ograniczoną kontrolę nad działaniem aplikacji.

Ograniczeniem dla wykorzystania koncepcji No Code może być też skalowalność poszczególnych platform, która może okazać się niewystarczająca, gdy liczba użytkowników przekroczy pewien próg. Warto zatem uprzednio zweryfikować możliwości wydajnościowe poszczególnych rozwiązań.



Mikołaj Marszycki



5 fazy rozwoju

oprogramowania w myśl
konceptji No Code

Model No Code ma pewne wspólne cechy z tradycyjnym sposobem tworzenia oprogramowania – takie jak choćby praktyki zwinne czy DevOps – ale nie można traktować go dokładnie w ten sam sposób.

Z jakich etapów składa się zatem cykl rozwoju aplikacji No Code?

Jak wskazują eksperci, unikalne wyróżniki platform No Code to: możliwość szybkiego wprowadzania aplikacji do użycia i łatwego jej rozwoju. Jest to kluczowa zaleta, ponieważ pozwala budować wśród użytkowników przekonanie, że ich opinia jest szybko uwzględniana i wpływa na rozwój produktu.

Podejście No Code angażuje osoby bez doświadczenia programistycznego, więc wymaga nadzoru i wsparcia ze strony zespołu IT tak, aby tworzone aplikacje odpowiadały specyficznym potrzebom danej organizacji m.in. w obszarze bezpieczeństwa czy zgodności z regulacjami.

faza pierwsza

Dotyczy szeroko rozumianego projektowania. Na tym etapie definiowane są wymagania biznesowe oraz warunki brzegowe dotyczące realizowanych w ramach projektu procesów, jak też jego interesariuszy. Analizowane są również możliwości istniejących już aplikacji, komponentów No Code oraz potrzeby i możliwości opracowania kolejnych. Platformy No Code są w tym zakresie bardzo pomocne. Umożliwiają korzystanie z gotowych komponentów i szablonów, a nie rzadko także ich dostosowywanie. Jednocześnie, na etapie projektowania rozwiązań No Code powstaje prototyp aplikacji, a nie jedynie jej specyfikacja. Należy więc uprzednio zdefiniować wszystkie potrzebne parametry aplikacji.

faza druga

Obejmuje budowanie i udostępnienie wąskiej grupie użytkowników początkowej wersji aplikacji zbudowanej w myśl koncepcji Minimal Viable Product, MVP. Jak wskazują eksperci, nie musi być ona idealna, a wręcz nie powinna, ponieważ dążenie do perfekcji mogłoby odwrócić uwagę od głównego schematu działania. Warto natomiast rozpocząć zbieranie uwag od użytkowników wersji testowej, gdyż będą one potrzebne przy dopracowywaniu produktu. Krytycznym momentem jest upewnienie się, że aplikacja jest zgodna z przyjętymi wcześniej założeniami i jest gotowa do udostępnienia szerokiej grupie użytkowników. Takiej oceny warto dokonywać na podstawie różnego rodzaju list kontrolnych uwzględniających wszystkie zdefiniowane wcześniej aspekty. Ostatnim etapem tej fazy jest udostępnienie aplikacji użytkownikom końcowym.

faza trzecia

Dotyczy m.in. testów wydajnościowych i wprowadzania dalszych usprawnień techniczno-funkcjonalnych. Istotne jest tu zatem utworzenie sprzężenia zwrotnego pomiędzy użytkownikami aplikacji, także tymi zaangażowanymi w jej rozwój. Na tym etapie aplikacja powinna być już regularnie wykorzystywana przez grupę docelowych użytkowników końcowych.

Każdy z przedstawionych etapów odgrywa kluczową rolę w cyklu życia oprogramowania budowanego i rozwijanego w myśl idei No Code.

Przy stosowaniu takiego podejścia ważne jest m.in. zachowanie pewnego rygoru pracy i zrozumienie unikalnych aspektów No Code, aby lepiej wykorzystać potencjał tej koncepcji w procesie cyfrowej transformacji organizacji.



Mikołaj Marszycki

faktów i mitów

o platformach Low Code

Organizacje z różnych sektorów, z myślą o usprawnieniu pracy i sprostaniu potrzebom klientów, coraz częściej sięgają po rozwiązania klasy Low Code. Popularność platform niskokodowych wynika nie tylko z trudnego dostępu do doświadczonych programistów, lecz także z chęci skrócenia czasu potrzebnego na realizację nowych projektów.

Warto jednak zastanowić się czym jest, a czym nie jest koncepcja Low Code.

Wiedza na temat rozwiązań typu Low Code **ułatwi wybór platformy dopasowanej do potrzeb danej organizacji** i pozwoli wykorzystać pełnię jej możliwości. Warto zatem rozwiązać powszechne mity na temat rozwiązań Low Code:

1

Platformy Low Code przeznaczone są wyłącznie dla małych firm i prostych aplikacji

Mit

Nie jest to może całkowity mit, ponieważ zbudowanie platformy programistycznej zdolnej do dostarczania złożonych aplikacji przy jednoczesnej maksymalizacji produktywności nie jest łatwe. Jednak błędne jest też założenie, że wszystkie platformy Low Code są sobie równe i przeznaczone do tworzenia nieistotnych aplikacji w małych firmach.

Platformy Low Code wykorzystywane są bowiem obecnie do tworzenia także bardziej złożonych aplikacji korporacyjnych, i to w organizacjach różnej wielkości na całym świecie — wymagają one jednak większych umiejętności technicznych. Istnieje wiele globalnych, dużych firm, które z powodzeniem działają na platformach i rozwiązaniach Low Code. Korzystają z nich m.in. Microsoft, Google, DHL, SAP, Siemens czy Volkswagen.



Low Code to rozwiązanie tylko dla osób nietechnicznych

Mit

Choć wielu osobom wydaje się, że platformy Low Code skierowane są wyłącznie do tzw. Citizen Developerów – czyli osób z niewielkim przeszkoleniem technicznym — to pozwalają one na tworzenie aplikacji również przez zaawansowanych użytkowników (mających już pewne umiejętności techniczne), a nawet przez profesjonalnych programistów.

Korzystając z tych platform, mogą oni dostarczać zespołowi w pełni funkcjonalne i unikalne rozwiązania, rozszerzając często dowolne warstwy platformy czy przenosząc swój tradycyjny kod, aby integrować go z aplikacjami niskokodowymi.

Należy przy tym podkreślić, że choć platformy Low Code sprawiają, że tworzenie aplikacji jest bezproblemowe i szybsze w porównaniu z tradycyjnymi metodami, mimo wszystko wymagają one od Citizen Developerów podstawowej wiedzy, funkcjonalnej znajomości wymagań biznesowych oraz opanowania korzystania ze wszystkich funkcji danej platformy.

2



3

Platformy Low Code to „zamknięte narzędzia”

Mit

Owszem, część architektury niskiego kodu opiera się na pewnych zastrzeżonych stosach, co utrudnia ich rozszerzenie, zmianę lub integrację. Jednak wiele platform Low Code zaprojektowano tak, aby pozwolić na bezproblemowe i skuteczne współpracowanie z innymi rozwiązaniami. Platformy te integrują się więc z istniejącymi już narzędziami, obsługują strategiczne interfejsy API, umożliwiają integrację tradycyjnego kodu i potrafią korzystać z kreatywnego wkładu społeczności programistów. Ponadto, nawet po zrezygnowaniu z subskrypcji, umożliwiają korzystanie z istniejących danych poprzez migrowanie ich czy integrowanie z innymi aplikacjami. Ważne jest tylko, aby przed wyborem platformy sprawdzić, czy jej dostawca rzeczywiście pozwala eksportować pełne dane w momencie zakończenia usługi.

Platformy typu Low Code blokują eksport kodu

Fakt

4

To akurat prawda, ale nie ma w tym nic wyjątkowego. Chociażby ze względu na to, że sercem każdej platformy Low Code jest unikalny silnik, który najczęściej stanowi tajemnicę handlową producenta. W efekcie eksport kodu aplikacji zbudowanej w jednej platformie do innego środowiska Low Code na niewiele by się zdał. Nawet jeśli byłoby to możliwe, nadal trzeba by zatrudnić programistów, tylko po to, aby wprowadzili w wyeksportowanym kodzie zmiany, które umożliwią jego uruchomienie w innej platformie. Oczywiście, istnieją platformy, które umożliwiają taki eksport konfiguracji.

5

Platformy Low Code powodują powstawanie „Shadow IT”

Mit

Termin Shadow IT odnosi się do praktyki wykorzystywania zasobów informatycznych, oprogramowania lub aplikacji w przedsiębiorstwie bez wsparcia zespołu IT lub bezpieczeństwa. W ujęciu Low Code oznacza to oczywiście złe zarządzanie i korzystanie z aplikacji oraz jej niekontrolowany rozwój.

Ponieważ tworzenie aplikacji dzięki platformom Low Code staje się łatwiejsze, panuje przekonanie, że może z nich korzystać każdy i to bez nadzoru, co prowadzi często do tworzenia silosów aplikacyjnych w organizacji i do bałaganu w jej systemie. Jeśli natomiast platforma taka używana jest prawidłowo, a więc pod nadzorem centralnego zespołu IT, umożliwi ona wszystkim jej użytkownikom – od Citizen Developerów, poprzez tych częściowo zaawansowanych pracowników, po doświadczonych programistów – uzyskanie dokładnie tego, czego szukają. A wspomniany centralny zespół IT odpowiada za wyeliminowanie jakiegokolwiek szansy na powstanie zjawiska Shadow IT, np. poprzez ustalenie dostępu do odpowiednich zasobów.

6

W przypadku platform Low Code niemożliwa jest ich skalowalność

Mit

Choć rzeczywiście w przypadku wczesnych wersji platform niskokodowych ich skalowalność była mocno utrudniona lub niemożliwa. Z kolei najnowsze, zaawansowane wersje platform Low Code, oparte na chmurze, oferują nieskończony potencjał pod względem wydajności i skalowalności. Dzięki nim firmy są w stanie zautomatyzować zarówno swoje procesy podstawowe, jak i te pomocnicze. Organizacje z różnych branż wybierają je po to, aby przyspieszyć swoją transformację cyfrową.

Platformy Low Code obejmują tylko fazę rozwoju aplikacji

Mit

Platformy Low Code wspierają pełny cykl życia aplikacji. Koncentrują się nie tylko na fazie kompilacji, lecz także na wszystkich innych aspektach, takich jak: projektowanie, testowanie, wdrażanie oraz utrzymanie. Platformy te oferują również funkcje zwinnego zarządzania projektami, współpracy społecznej, pętli informacji zwrotnych od użytkowników itd. Wszystkie te funkcje razem pomagają zachować płynność w całym cyklu życia aplikacji.

7



8

Platformy Low Code zapewniają wysoką elastyczność

Fakt

Platformy Low Code tworzone są właśnie w celu zapewnienia elastyczności zarówno osobom fizycznym, jak i firmom, aby można było rozwiązywać różne problemy i tworzyć aplikacje za pomocą takich rozwiązań. Gwarantują użytkownikowi swobodę oraz elastyczność w przekształcaniu swoich unikalnych pomysłów w aplikację w krótkim czasie. A w przeciwieństwie do gotowego oprogramowania, pozwalają użytkownikom modyfikować lub rozszerzać funkcjonalność aplikacji w dowolnym momencie.

9

Platformy Low Code nie są bezpieczne

Mit

Poziom zabezpieczeń zależy oczywiście od możliwości wybranej platformy Low Code, ale najbardziej zaawansowane z nich zapewniają rozbudowane mechanizmy bezpieczeństwa, a także pozwalają na zapewnienie zgodności z regulacjami czy politykami z obszaru governance. I tak, najbardziej zaawansowane środowiska Low Code gwarantują m.in. odpowiednie kontrole na każdym etapie działania — w tym projektowania UI/UX, rozwoju, hostingu, dostarczania i konserwacji. Każda platforma Low Code przeznaczona do zastosowań korporacyjnych dostarcza również szczegółowych informacji o stosowanych rozwiązaniach z zakresu cyberbezpieczeństwa. Poza tym dostawcy takich platform poddawani są audytowi kontroli bezpieczeństwa pod kątem najlepszych standardów bezpieczeństwa dostępnych na rynku.

Platformy typu Low Code są drogie w utrzymaniu

Mit

Podobnie jak wszystkie platformy chmurowe, platformy Low Code pracują na zasadach pay-as-you-go. Oznacza to, że płaci się za ich korzystanie, a kiedy tego nie robimy, to po prostu nie płacimy. Dzięki temu ponosimy koszt tylko za to, z czego naprawdę korzystamy. Ponadto zawsze istnieje kilka poziomów usługi dopasowanych do potrzeb danej organizacji. Koszty utrzymania platform Low Code są więc uzależnione od skali ich użycia, a także skali wykorzystania zbudowanych za ich pomocą aplikacji. Nierzadko są to zatem koszty niższe od tych związanych z utrzymaniem starszych rozwiązań lokalnie.

11

Platformy Low Code utrudniają współpracę

Mit

Wykazaliśmy już wcześniej, że współpraca jest integralną częścią platform Low Code. Ba, ten rodzaj podejścia do rozwoju aplikacji wymaga nawet większej współpracy niż tradycyjny, w którym programiści pracują samodzielnie, aby napisać wiele linii kodu. Z drugiej strony, przy rozwiązaniach niskokodowych użytkownicy spotykają się, by zbudować produkt końcowy — oczywiście będąc pod nadzorem centralnego zespołu IT — i wszyscy mają w tym procesie coś do powiedzenia. Wrzeczywistości bywa nawet, że w tworzeniu aplikacji w ujęciu Low Code uczestniczy cała organizacja.

10

12

Platformy Low Code to tylko kolejny trend

Mit

Pewnie mało kto o tym wie, ale historia platform Low Code sięga lat 80. XX wieku. Ponownie do użytku zostały one wprowadzone w 2012 roku i od tego czasu dynamicznie się rozwijają. Według GlobeNewswire, globalny rynek platform Low Code wart był w 2021 roku 16,3 mld USD. Do roku 2030 osiągnie zaś wartość 148,5 mld USD. Z kolei według niedawnego badania Gartnera, do 2024 roku ok. 65% rozwoju aplikacji będzie powstawać przy użyciu technologii niskokodowych.

Na platformach Low Code nie ma możliwości korzystania z własnego backendu

Mit

Kolejny mit. Dzięki różnym integracjom (API, OAuth, SQL itp.) można łatwo połączyć frontend zbudowany w środowisku Low Code z tradycyjnym, lokalnym środowiskiem backend. Na przykład AppGyver nie oferuje żadnych usług backend. Możemy tylko zbudować frontend aplikacji mobilnej, a wszystkie dane powinny być dostarczane przez API.

13



Mikołaj Marszycki

WE FIND the WAY

on the roads
to Digital
Transformation

Delivering custom-developed solutions using
high performance Low-Code platform
and cloud native technology.



candf.com

