



CODE CAN FLY

CODE CAN FLY

Cieężar odpowiedzialności. Lekkość współpracy.

Budujemy systemy kluczowe dla działalności organizacji. Własnym zespołem, od pierwszych decyzji po utrzymanie – z jakością i przewidywalnością, których wymaga długoterminowa odpowiedzialność.

ODPOWIEDZIALNI ZA REZULTAT

WYMAGANIA · ARCHITEKTURA · INŻYNIERIA · JAKOŚĆ · UTRZYMANIE

Firma wykonawcza dla systemów, od których naprawdę zależy organizacja.

Code Can Fly łączy kompetencje analityczne, architektoniczne, inżynierskie, jakościowe, operacyjne i zarządcze w jednym zespole odpowiedzialnym za rezultat.

Od 2020 roku rozwijamy organizację zdolną prowadzić pełny cykl życia systemu.

ZNACZENIE

Krytyczność nie jest etykietą branży.

System jest dla nas krytyczny wtedy, gdy jego niedostępność, błędne działanie albo niekontrolowana zmiana wpływają na podstawową działalność organizacji, jej klientów, użytkowników lub obywateli.



OBIETNICA

Jakość i przewidywalność prowadzą do odpowiedzialnej terminowości.

Nie oddzielamy terminu od jakości. Budujemy przewidywalność przez standardy, widoczne zależności, zarządzanie ryzykiem, jednoznaczne kryteria i regularną weryfikację rezultatów.

Requirements First - Great Solutions to Follow

Właściwe rozwiązanie zaczyna się przed pierwszą linią kodu.

Jakość systemu zaczyna się od rygorystycznego zrozumienia celu, użytkowników, kontekstu otoczenia, a następnie uwzględnienia ich od początku w architekturze, zakresie funkcjonalnym, wskaźnikach jakościowych i wydajnościowych oraz kryteriach odbioru. W Code Can Fly stosujemy podejście Requirements First - Great Solutions to Follow. Efekt? Mniej iteracji, niższe koszty utrzymania i szybsze osiągnięcie zamierzonej wartości. A Ty — od czego zaczynasz projekt IT?

Nasze podejście wyrasta z doświadczenia.

W Code Can Fly zaczynamy od organizacji — jej procesów, ludzi, ograniczeń i celów, a nie od technologii poszukującej zastosowania. Nasza fundamentalna zasada brzmi: zanim zaprojektujemy system, wspólnie ustalamy, co ma umożliwić, jakie procesy i stanowiska wesprzeć, w jakim otoczeniu będzie działać, jakimi ograniczeniami będzie się mierzyć oraz po czym poznamy, że właściwie pełni swoją rolę. To eliminuje ryzyko budowania rozwiązań technicznie poprawnych, a biznesowo obcych.	Nasza przewaga konkurencyjna tkwi w połączeniu dwóch obszarów, które zbyt często rozdzielają tradycyjne software house'y: świetnie zarządzane wymagania wyłonione z architektury biznesowej oraz dobór architektury IT pod ich adekwatną realizację. Działamy w oparciu o branżowe wzorce: ramy architektury korporacyjnej wyznacza TOGAF, proces zarządzania wymaganiami opieramy na praktykach IREB oraz wytycznych normy ISO/IEC/IEEE 29148; BPMN i UML uspoólniają narzędzie komunikacji pomiędzy interesariuszami projektu. Dzięki temu wymagania są nie tylko kompletne i spójne, ale realnie powiązane z procesami biznesowymi organizacji, a architektura służy realizacji tych wymagań, a nie odwrotnie.	Dobre wymagania to nie koszt na starcie, lecz inwestycja w decyzje, które nie okażą się później kosztownym błędem. Umożliwiają lepszą architekturę (dobór technologii i wzorców, które służą celom, a nie modzie), przewidywalność realizacji (świadome planowanie zakresu, wczesne ujawnianie zależności i ryzyk) oraz sprawiedliwe miary sukcesu (jasne kryteria odbioru łączące prace zespołu z rezultatem istotnym dla organizacji, a nie tylko „wykonaniem zadań”). To przekłada się na mniejszą liczbę iteracji, niższe koszty utrzymania i szybsze osiągnięcie zamierzonej wartości.
Wymagania same nie budują systemu, ale stanowią punkt odniesienia dla wszystkich innych dziedzin inżynierii oprogramowania — dzięki czemu analiza, architektura, implementacja, testy i utrzymanie dostarczają wartość na rzecz realizacji tego samego celu. Wymagania żyją i ewoluują razem z wiedzą o systemie, rynkiem i użytkownikami. Kluczem jest świadome zarządzanie tą zmianą: każda modyfikacja wymaga uzasadnienia, analizy wpływu i powiązania z decyzjami, na które wpływa. Dzięki temu elastyczność nie oznacza chaosu, a iteracje prowadzą w stronę lepszego rozwiązania.	Bazując na takim fundamencie, dostarczamy rozwiązania o najwyższej jakości, wykorzystując do tego celu technologie i procesy wytwórcze dobrane do jak najszybszego osiągnięcia celu projektu.	Potrąfimy mierzyć to, co wytwarzamy — m.in. z wykorzystaniem metod punktów funkcyjnych (COSMIC, IFPUG) — co pozwala nam z dużą precyzją szacować i kontrolować terminy oraz koszty wytwarzania.

Doświadczenie przekładamy na uporządkowany proces.

01

Nasza metoda nie powstała w prezentacjach ani w ramach jednorazowych wdrożeń. Została ukształtowana przez doświadczenie z analizą, architekturą, budową, testowaniem, wdrażaniem i utrzymaniem dużych systemów, w których największe problemy niemal zawsze miały swoje źródło w nieuzgodnionych celach, ukrytych założeniach, niedoskonale rozpoznanych zależnościach lub w wymaganiach, których nie dało się jednoznacznie zweryfikować. W Code Can Fly traktujemy te doświadczenia jako strategiczny atut: dzięki nim wiemy, gdzie i kiedy pojawiają się realne ryzyka, a przede wszystkim – jak je eliminować, zanim staną się kosztownym problemem.

02

Dlatego stosujemy własny, uporządkowany proces analizy, w którym łączymy świat biznesu z inżynierią w sposób, który jest zarówno rygorystyczny, jak i praktyczny. Pozyskujemy wiedzę od wszystkich kluczowych interesariuszy, porządkujemy ją i transformujemy we wspólną platformę wiedzy opisaną w jednolity sposób dla naszych klientów oraz zespołu realizacyjnego.

03

Dbamy o wspólne pojęcia, identyfikujemy założenia i uwidaczniamy zależności – dzięki czemu decyzje są podejmowane na podstawie faktów, a nie domysłów. Weryfikujemy i uzgadniamy wymagania z osobami posiadającymi wiedzę domenową oraz z zespołem odpowiedzialnym za architekturę, implementację, testy i późniejsze działanie rozwiązania, aby sprzeczności, luki i ryzyka można było rozwiązywać wtedy, gdy ich korekta jest jeszcze stosunkowo łatwa i tania.

04

Uznane standardy i metodyki, będące fundamentem naszych standardów pracy, potrafimy elastycznie dostosowywać do realiów przedsięwzięcia. Zamiast wdrażać je „na siłę”, dobieramy sposób pracy i poziom formalizacji do skali, ryzyka, etapu życia oraz otoczenia systemu – tak, aby wspierały proces oceny i decyzji, a nie zastępowały wiedzy domenowej, odpowiedzialności ani wyborów wynikających z faktycznej sytuacji organizacji. W Code Can Fly metodyka jest narzędziem do osiągania celu, nie celem samym w sobie; jej wartość mierzymy skutecznością, przewidywalnością i realnym wpływem na jakość dostarczanego rozwiązania.

PUNKTY ODNIESIENIA

Korzystamy z tych źródeł jako sprawdzianu i uzupełnienia własnej praktyki. Nie oznacza to certyfikacji ani stosowania jednego gotowego procesu.

TOGAF Ramy architektury korporacyjnej wspierające powiązanie celów organizacji, procesów, informacji, aplikacji i technologii.	BPMN Standaryzowana notacja procesów biznesowych, która tworzy wspólny język dla interesariuszy biznesowych i technicznych.	UML Standaryzowany język modelowania służący do opisu struktury, zachowania i współpracy elementów systemu.	ISO/IEC/IEEE 29148 Procesy, produkty pracy i cechy jakościowe inżynierii wymagań w całym cyklu życia systemu.	IREB Praktyki pozyskiwania, dokumentowania, walidowania i zarządzania wymaganiami.	COSMIC Metoda pomiaru funkcjonalnego rozmiaru oprogramowania oparta na przepływach danych i niezależna od technologii implementacji.	IFPUG Metoda pomiaru funkcjonalnego rozmiaru oprogramowania oparta na funkcjach danych i funkcjach transakcyjnych widocznych dla użytkownika.
--	---	---	---	--	--	---

Odpowiedzialność za system — od pierwszych decyzji po utrzymanie.

Zakres dobieramy do momentu, w którym znajduje się system.

01

Budowa nowego systemu

Dla organizacji, która potrzebuje nowego rozwiązania wspierającego jej podstawową działalność i chce zachować ciągłość odpowiedzialności od pierwszych decyzji.

- rozpoznanie celu, użytkowników, ograniczeń i kryteriów odbioru
- architektura rozwiązania, plan realizacji i zarządzanie ryzykiem
- budowa, integracje, kontrola konfiguracji i jakości kodu

02

Modernizacja i rozwój

Dla organizacji, której istniejący system ogranicza rozwój, generuje ryzyko albo nie odpowiada już wymaganiom biznesowym i technicznym.

- rozpoznanie systemu, zależności, ryzyk i priorytetów zmiany
- docelowa architektura oraz plan etapów i migracji
- modernizacja komponentów, danych i integracji

03

Przejęcie utrzymania i dalszego rozwoju

Dla organizacji, która potrzebuje nowego, przewidywalnego modelu odpowiedzialności za działający system.

- rozpoznanie stanu, zobowiązań operacyjnych i ryzyk przejęcia
- plan transferu wiedzy, dostępów, narzędzi i dokumentacji
- uporządkowanie środowisk, monitoringu, logów i procedur

Pełny łańcuch kompetencji. Jedna odpowiedzialność za rezultat.

Budujemy zespoły obejmujące kolejne etapy życia systemu. Kompetencje działają wspólnie, a nie jako osobny katalog ról i technologii.



Analiza i wymagania

Cele biznesowe, procesy, wymagania funkcjonalne i нефункционалне, integracje oraz kryteria odbioru.

Architektura rozwiązania

Granice systemu, komponenty, dane, integracje, decyzje technologiczne i atrybuty jakościowe.

Inżynieria oprogramowania

Implementacja, przeglądy kodu, zarządzanie konfiguracją i bezpieczna ewolucja rozwiązania.

Integracja systemów i danych

Kontrakty integracyjne, przepływy danych i współpraca w ekosystemach wielu systemów oraz dostawców.

Jakość i testy

Strategia jakości, testy funkcjonalne, integracyjne, regresyjne, wydajnościowe i bezpieczeństwa.

Bezpieczeństwo

Wymagania, projektowanie, kontrola zmian, weryfikacja i eksploatacja z uwzględnieniem ryzyka.

DevOps i obserwowalność

Automatyzacja budowania i wdrożeń, środowiska, monitoring, logi i gotowość operacyjna.

Zarządzanie realizacją

Portfel, projekt, zakres, zależności, ryzyka, komunikacja i przejrzysty przepływ decyzji.

Utrzymanie i rozwój

Obsługa zgłoszeń i incydentów, stabilizacja, planowanie zmian oraz długoterminowy rozwój.

Narzędzia wspierające pełny cykl realizacji

Dobieramy warsztat do celu i warunków działania systemu. Nie zaczynamy od stosu technologicznego. Architektura, technologie, narzędzia i sposób pracy wynikają z celu systemu, jego ograniczeń, wymagań jakościowych oraz warunków utrzymania. Poniższy katalog pokazuje reprezentatywny przekrój środowisk naszych realizacji i kompetencji zespołu – od analizy i organizacji pracy, przez budowę i integrację, po jakość, infrastrukturę, dane i rozwiązania AI.

01	Analiza i architektura	UML		BPMN		ArchiMate		Enterprise Architect											
02	Metodyki, sposoby pracy i projektowanie zorientowane na użytkownika	Scrum		AgilePM		PRINCE2 Project Management		Podejście predykcyjne (waterfall)		UX / Human-Centred Design									
03	Narzędzia wytwarzania, zarządzania pracą i CI/CD	JetBrains AI Assistant	Git	GitHub	GitLab CI/CD	Bitbucket	Bamboo	Jira	Confluence	Redmine	BlueOptima								
04	Backend i bazy danych	Java / OpenJDK		Spring		.NET .NET / C#		ORACLE Oracle Database		Microsoft SQL Server		IBM Db2	MongoDB	Couchbase					
05	Platformy aplikacyjne i uruchomieniowe	IBM WebSphere		JBoss EAP		Apache Tomcat		WildFly		Microsoft IIS		Liferay		IBM BAW					
06	Integracja, API i middleware	API REST		{...} SOAP / Web Services		OpenAPI / Swagger		Apache Camel		JBoss Fuse		IBM MQ		ActiveMQ	Apache Kafka				
07	Frontend	Angular		React		Vue.js		HTML5		CSS3		JavaScript							
08	Testy i jakość	JUnit		REST Assured		Selenium		Playwright		Robot Framework		SonarQube							
09	DevOps, infrastruktura i obserwowalność	Docker	Kubernetes	OKD		Red Hat OpenShift		Linux	Windows Server		vmware VMware		Proxmox VE	Zabbix		Grafana	Grafana Loki		
10	Dane, raportowanie, analityka i AI	ORACLE Oracle Discoverer		Power BI		SSIS / ETL		JasperReports		Amazon Redshift		Ollama		RAG (Retrieval-Augmented Generation)		LangChain		LangGraph	

Znaki identyfikują technologie i nie oznaczają partnerstwa ani certyfikacji CCF.

Systemy ważne dla działania instytucji i usług publicznych.

Trzy przykłady pokazują, jak łączymy analizę, architekturę, budowę, jakość, wdrożenie i utrzymanie w zależności od etapu oraz potrzeb konkretnego systemu.



MINISTERSTWO SPRAWIEDLIWOŚCI
OD 2020 · PRACE TRWAJĄCE

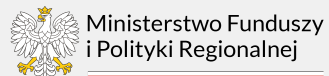
Cyfrowe usługi wymiaru sprawiedliwości

ZAANGAŻOWANIE CCF

Od 2020 roku CCF rozwija i utrzymuje wybrane komponenty tych rozwiązań. Zakres obejmuje analizę wymagań, rozwój architektury, implementację modyfikacji, testy, wdrożenia, zadania utrzymaniowe oraz bieżące wsparcie działających systemów.

ROZWIĄZANIA

Krajowy Rejestr Sądowy · Repozytorium Akt Rejestrowych · Rejestr Zastawów · e-Płatności · System Nieodpłatnej Pomocy Prawnej



MINISTERSTWO FUNDUSZY I POLITYKI REGIONALNEJ
2020–2026

Krajowy System Informatyczny SIMIK

ZAANGAŻOWANIE CCF

Budowa modułu e-Kontrola — W powierzonym zakresie uczestniczyliśmy w pełnym łańcuchu realizacyjnym: od analizy wymagań i projektu rozwiązania, przez wytworzenie i testy, po wdrożenie.

KSI SIMIK — Do 2026 roku zapewnialiśmy wsparcie utrzymania systemu i jego infrastruktury, realizując działania DevOps, obsługę operacyjną i zadania utrzymaniowe wspierające ciągłość działania.



ARiMR
2021–2022

Centralny System Obsługi Beneficjenta

ZAANGAŻOWANIE CCF

CCF miało przyjemność wspierać swojego kluczowego partnera Asseco Poland w inicjacji projektu CSOB dla ARiMR. Wspieraliśmy uruchomieniem przedsięwzięcia oraz realizację analizy wymagań i prac nad architekturą rozwiązania.



CODE CAN FLY

CODE CAN FLY · KONTAKT

Zaczniemy od tego, co system ma umożliwić Twojej organizacji.

Napisz, jeżeli planujesz nowy system, modernizujesz istniejące rozwiązanie albo przygotowujesz przejęcie utrzymania.

EMAIL

kontakt@codecanfly.com

WWW

<https://codecanfly.com>

LINKEDIN

[Code Can Fly](#)

SIEDZIBA

*ul. Gościńska 7
05-082 Blizne Łaszczynskiego
Polska*