

Umowy IT. Specyfika, tworzenie i negocjowanie + wzory do pobrania

Przejdź do produktu na ksiegarnia.beck.pl

Część I. Rodzaje umów IT i ich specyfika

Mateusz Kozieł

Rozdział I. Podstawy prawne umów IT w Polsce

1. Uwagi wstępne

Jak pokażą dalsze rozdziały niniejszej publikacji, umowy IT to barwna mozaika rozmaitych stosunków umownych, mających za przedmiot różnego rodzaju aspekty i etapy projektu informatycznego. Przedmiot i cel poszczególnych umów cechują się dużą różnorodnością, a wielorakie świadczenia wykonywane przez strony umów IT powodują, że kontrakty te bardzo często nie dają się jednoznacznie zakwalifikować pod kątem prawnym.

Nie oznacza to jednak, że kontrakty IT to „dziki zachód”, niepodlegający jakimkolwiek przepisom, w którym treść umowy stanowi źródło obowiązującego dla stron „prawa” (co po części ma miejsce na gruncie prawa anglosaskiego). Tworząc i negocjując umowę IT musimy brać pod uwagę wiele regulacji prawnych, zarówno tych o charakterze ogólnym (takich jak chociażby KC), jak i tych o charakterze *stricte* szczegółowym lub wręcz branżowym, specyficznym często nawet nie dla całego sektora IT, ale konkretnej gałęzi produktów informatycznych¹.

Umowy dotyczące świadczenia usług IT, w tym usług utrzymania, rozwoju oprogramowania czy *outsourcingu*, nie są bowiem w polskim systemie prawnym uregulowane w sposób kompleksowy w jednym akcie normatywnym. Ich konstrukcja opiera się na zasadzie swobody umów oraz na zastosowaniu wielu przepisów ogólnych i szczególnych, pochodzących z różnych gałęzi prawa. W praktyce oznacza to konieczność uwzględnienia za-

¹ Inną siatkę regulacyjną zastosujemy bowiem tworząc np. kontrakt na wdrożenie systemu informatycznego realizującego obsługę kadrowo-płacową (gdzie dodatkowo w grę wejdzie konieczność zapewnienia zgodności efektów projektu również z regulacjami prawa pracy i prawa podatkowego), a inną w przypadku tworzenia umowy dotyczącej realizacji aplikacji mobilnej, umożliwiającej realizację programu lojalnościowego skierowanego do konsumentów (gdzie dodatkowo umowa powinna uwzględnić przepisy konsumenckie i możliwość ich realizacji przez oprogramowanie).

równy regulacji prawa cywilnego, jak i przepisów dotyczących własności intelektualnej, ochrony danych osobowych oraz – coraz częściej – prawa unijnego, regulującego obrót danymi i usługi cyfrowe.

Nie próbując nawet pokusić się o stworzenie enumeratywnej listy przepisów koniecznych do stworzenia umowy funkcjonującej w prawie polskim, warto jednak w niniejszym rozdziale stworzyć krótką „ściągawkę” regulacyjną dla kontraktów IT.

Niezależnie od tego, warto również uzmysłowić sobie, jak istotna w zakresie umów IT jest specyfika umów nienazwanych oraz jak duży wpływ mają dobre praktyki kontraktowe (w tym praktyki wynikające z metodyki realizacji umowy) na treść, zasady ukształtowania i realizacji umowy IT.

2. Przepisy ogólne

Podstawowym aktem prawnym mającym zastosowanie do umów IT jest KC. Stanowi on fundament konstrukcyjny większości kontraktów informatycznych, w szczególności w zakresie zasad zawierania umów, odpowiedzialności kontraktowej, wykonania zobowiązań (w tym zasad współdziałania stron w wykonaniu umowy) oraz skutków ich niewykonania lub nienależytego wykonania. **Na gruncie KC kształtowane są m.in. postanowienia dotyczące przedmiotu umowy, wynagrodzenia, terminów realizacji, kar umownych, odpowiedzialności odszkodowawczej, a także klauzule dotyczące siły wyższej czy odstąpienia od umowy.** W przypadku usług utrzymania znaczenie mają również przepisy regulujące umowy o świadczenie usług oraz zasady należytej staranności, które stanowią punkt odniesienia dla oceny jakości realizowanych świadczeń.

Kolejnym kluczowym aktem jest PrAut, o fundamentalnym znaczeniu dla umów dotyczących wytwarzania, modyfikacji, utrzymania i eksploatacji oprogramowania. **Na podstawie PrAut kształtowane są postanowienia umowne dotyczące autorskich praw majątkowych, licencji, pól eksploatacji, zasad przenoszenia praw oraz uprawnień do tworzenia i korzystania z utworów zależnych.** W kontekście usług utrzymania istotne są również regulacje dotyczące modyfikacji oprogramowania, dostępu do kodu źródłowego oraz uprawnień do korzystania z efektów prac rozwojowych, wykonywanych w ramach realizacji umowy.

Istotną rolę w umowach IT odgrywiają przepisy dotyczące ochrony danych osobowych, w szczególności RODO. Regulacja ta znajduje zastosowanie wszędzie tam, gdzie w ramach realizacji umowy dochodzi do przetwarzania danych osobowych, co w praktyce dotyczy znacznej części usług IT. **Na gruncie RODO kształtowane są m.in. postanowienia dotyczące ról stron (administrator – podmiot przetwarzający), zasad przetwarzania danych, obowiązków w zakresie bezpieczeństwa informacji, procedur reagowania na naruszenia ochrony danych oraz odpowiedzialności stron.** W przypadku usług utrzymania szczególnego znaczenia nabierają zapisy dotyczące dostępu do środowisk produkcyjnych, logów systemowych oraz danych użytkowników końcowych.

Coraz większe znaczenie dla umów IT mają również regulacje unijne dotyczące danych i usług cyfrowych, w tym **Data Act, wpływający na sposób kształtowania postano-**

wień umownych związanych z dostępem do danych, ich udostępnianiem, ponownym wykorzystaniem oraz przenoszalnością pomiędzy dostawcami usług. W praktyce oznacza to konieczność uwzględnienia w umowach IT zapisów dotyczących praw do danych generowanych w trakcie korzystania z systemów informatycznych, zasad ich udostępniania klientom lub podmiotom trzecim oraz obowiązków dostawcy w zakresie interoperacyjności i migracji usług.

3. Umowy IT jako umowy nienazwane

Z punktu widzenia przepisów prawa cywilnego (przede wszystkim KC), **umowa IT będzie najczęściej typem umowy nienazwanej (tj. nieuregulowanej wprost w przepisach)**. To od pełnej i niczym nieskrępowanej woli stron umowy IT zależy więc charakter umowy, jaką zdecydują się zawrzeć w celu unormowania swej współpracy. Zgodnie bowiem z przepisami KC, „strony zawierające umowę mogą ułożyć stosunek prawny według swego uznania, byleby jego treść lub cel nie sprzeciwiały się właściwości (naturze) stosunku, ustawie ani zasadom współżycia społecznego”². Jak podkreśla się w literaturze, w ramach swobody kształtowania stosunku obligacyjnego strony mogą zawrzeć umowę nienazwaną, z kolei w przypadku wybrania jednej z umów nazwanych (uregulowanych ustawowo) ustalić treść konkretnej umowy w taki sposób, że postanowienia jej będą odbiegać od wzorca normatywnego³.

W relacjach IT typ umowy, w tym wynikające z niej prawa i obowiązki stron powinny być ściśle dostosowane do rodzaju podejmowanej współpracy. Należy rozstrzygnąć w szczególności, czy jest to współpraca projektowa, czy bieżąca, i odpowiedzieć na pytanie, czego oczekujemy od personelu, czy chodzi o osiągnięcie określonego rezultatu, jak choćby budowa lub rozbudowa systemu w ramach wdrożenia/świadczenia usług (np. w ramach realizacji zgłoszeń serwisowych). Te okoliczności będą determinowały typ umowy.

4. Podstawowe typy umów

Z punktu widzenia prawnego należy wskazać, że umowa IT może przyjąć następujące postacie, uregulowane wprost przepisami KC:

- 1) umowa zlecenia;
- 2) umowa o dzieło;
- 3) umowa o świadczenie usług.

² Art. 353¹ KC.

³ P. Nazaruk, (w:) B. Bajor, D. Bierecki, J. Bocianowska, J. Ciszewski, M. Ciszewski, G. Karaszewski, J. Knabe, J. Mucha-Kujawa, G. Sikorski, B. Sitek, R. Tanajewska, P. Nazaruk, Kodeks cywilny. Komentarz aktualizowany, LEX/el. 2024, komentarz do art. 353¹.

Oznacza to możliwość dostosowywania klauzul do danych umów, dzięki którym umowa IT może mieć cechy:

- 1) umowy zlecenia, ze względu na cechę ciągłości współpracy ze stroną, staranne działanie, osobiste wykonanie czy też brak podporządkowania (większa swoboda);
- 2) umowy o dzieło, ze względu na postanowienia, które określają wymagania dotyczące jakości i terminów realizacji określonego zadania;
- 3) umowy o świadczenie usług, która jest najbliższej klasycznej umowy IT, decyduje o tym np. świadczenie główne dłużnika polegające na czynieniu.

Ważne

Należy zwrócić uwagę na reguły interpretacji umów, gdyż o typie (rodzaju) umowy decyduje zgodny zamiar stron i cel umowy, który może (ale wcale nie musi) wyrażać się także w nazwie umowy.

Zasadniczo wyróżniamy dwa alternatywne modele umowy cywilnoprawnej, stosowane najczęściej w relacjach kontraktowych:

- 1) umowa o dzieło – gdy oczekujemy od wykonawców określonych rezultatów prac (np. fragment systemu IT, określony, wymierny cel);
- 2) umowa o świadczenie usług (często określana zamiennie zleceniem) – gdy oczekujemy, że usługobiorca będzie w sposób staranny i sumienny realizował określone czynności prawne/usługi zdefiniowane w umowie (np. obsługę zgłoszeń serwisowych w ramach helpdesku).

Wskazówek dotyczących rozróżnienia poszczególnych typów umów cywilnoprawnych doszukać się można w bogatym orzecznictwie sądów powszechnych.

Orzecznictwo

Istotą umowy o dzieło jest osiągnięcie określonego, zindywidualizowanego rezultatu w postaci materialnej lub niematerialnej. W przypadku wadliwego albo sprzecznego z umową wykonywania dzieła zamawiający może wezwać przyjmującego zamówienie do zmiany sposobu wykonania umowy i wyznaczyć mu w tym celu odpowiedni termin, a po jego bezskutecznym upływie odstąpić od umowy lub powierzyć poprawienie lub dalsze wykonanie dzieła osobie trzeciej.

Umowa o świadczenie usług jest zaś umową starannego działania, której przedmiotem jest wykonywanie określonych czynności. Czynności te mogą ale nie muszą zmierzać do osiągnięcia rezultatu, nie muszą też być wykonywane stale i systematycznie. Jeśli zatem zleceniobiorca zachowa należyta staranność przy wykonywaniu powierzonych mu czynności, wówczas nie ponosi odpowiedzialności za niewykonanie zobowiązania (wyr. SA w Lublinie z 28.8.2013 r., III Aua 610/13, Legalis).

Konsekwencje wyboru danego typu umowy przekładają się w szczególności na następujące kwestie:

- 1) przedmiot umowy i wynagrodzenie:
 - a) w modelu usługowym – przedmiotem jest wykonanie usług, wynagrodzenie należy się za świadczenie usług,
 - b) w modelu dziełowym – przedmiotem jest wykonanie dzieła, wynagrodzenie należy się za dostarczenie rezultatu;

- 2) sposób realizacji umowy:
 - a) w modelu usługowym – wykonanie czynności faktycznych, co do zasady samodzielnie (art. 738 § 1 KC),
 - b) w modelu dziełowym – wykonanie oznaczonego dzieła, co do zasady możliwość skorzystania z podwykonawców (art. 356 § 1 w zw. z art. 645 § 1 KC);
- 3) zasady i zakres odpowiedzialności:
 - a) w modelu usługowym – odpowiedzialność za staranne działanie, odpowiedzialność na zasadach ogólnych,
 - b) w modelu dziełowym – odpowiedzialność za osiągnięcie rezultatu, odpowiedzialność za zgodność rezultatu z rezultatem oznaczonym w umowie, a także co do zasady rękojmia za wady dzieła;
- 4) możliwość, tryb i skutki jednostronnego rozwiązania umowy:
 - a) w modelu usługowym – wypowiedzenie, skutek na przyszłość; umownie można ograniczyć wypowiedzenie do ważnych powodów, zasadniczo obydwie strony mogą wypowiedzieć umowę w każdym czasie,
 - b) w modelu dziełowym – odstąpienie, skutek wsteczny (zasadniczo, można uregulować w umowie), przepisy określają sytuacje, kiedy odstąpienie jest dopuszczalne, a także można przewidzieć umowne prawo odstąpienia.

5. Znaczenie dobrych praktyk kontraktowych

Dobre praktyki kontraktowe odgrywają kluczową rolę w kształtowaniu stabilnych i efektywnych relacji między stronami umów IT. Złożoność projektów informatycznych, dynamiczny charakter technologii oraz asymetria wiedzy technicznej między wykonawcą a zamawiającym sprawiają, że klasyczne podejście do konstruowania umów często okazuje się niewystarczające. W tym kontekście dobre praktyki pełnią funkcję narzędzi porządkujących, pozwalających na precyzyjne określenie zakresu odpowiedzialności stron, zasad współpracy, mechanizmów zarządzania ryzykiem oraz sposobów rozwiązywania sporów. Ich stosowanie sprzyja ograniczeniu niepewności kontraktowej i zwiększa przewidywalność realizacji świadczeń.

Jednym z najważniejszych elementów dobrych praktyk kontraktowych w obszarze IT jest odejście od nadmiernego formalizmu na rzecz jasnego, funkcjonalnego opisu świadczeń i procesów. **W dobrze skonstruowanych umowach IT szczególną wagę przywiązuje się do precyzyjnego zdefiniowania pojęć technicznych, ról projektowych, procedur odbiorowych oraz zasad zarządzania zmianą.** Praktyka rynkowa pokazuje, że brak spójnego języka umowy lub posługiwanie się terminologią niejednoznaczną prowadzi do trudności interpretacyjnych zwiększając ryzyko *vendor lock-in*⁴, które w przypadku sporów kontraktowych może mieć istotne konsekwencje prawne i finansowe. Dobre praktyki zakładają zatem ścisłe powiązanie warstwy prawnej umowy z realiami operacyjnymi świadczenia usług IT.

⁴ Uzależnienie od dostawcy (tu: oprogramowania). W przypadku umów, wprowadzenie w nich specyficznych pojęć i instytucji, unikalnych dla kontraktu spowoduje, że podmiot trzeci nie będzie w stanie kontynuować jego realizacji w miejsce np. pierwotnego dostawcy IT.

Szczególne znaczenie w tym obszarze ma ITIL (ang. Information Technology Infrastructure Library), będący jednym z najbardziej rozpowszechnionych zbiorów dobrych praktyk w zakresie zarządzania usługami IT. ITIL powstał w latach 80. XX w., z inicjatywy administracji publicznej w Wielkiej Brytanii, jako odpowiedź na potrzebę standaryzacji i podniesienia jakości usług informatycznych świadczonych przez podmioty publiczne. Z czasem ITIL ewoluował w kierunku uniwersalnego, międzynarodowego standardu dobrych praktyk, znajdującego zastosowanie zarówno w sektorze publicznym, jak i prywatnym. W aktualnej wersji ITIL 4 podejście to koncentruje się na współtworzeniu wartości, elastyczności oraz integracji zarządzania usługami IT z celami biznesowymi organizacji. Z perspektywy kontraktowej, **ITIL dostarcza spójnego aparatu pojęciowego oraz ram organizacyjnych, które można bezpośrednio wykorzystać przy konstruowaniu umów dotyczących świadczenia, utrzymania i rozwoju usług IT**, w szczególności w zakresie definiowania usług, poziomów świadczenia (SLA), odpowiedzialności stron oraz mechanizmów ciągłego doskonalenia.

Odwołanie się do ITIL w treści umów IT stanowi przykład świadomego wykorzystania dobrych praktyk jako narzędzia ograniczania ryzyka kontraktowego. Standaryzacja pojęć i procesów oparta na ITIL pozwala na ujednoczenie oczekiwań stron oraz ułatwia interpretację postanowień umownych w przypadku sporów. Jednocześnie ITIL, jako zbiór dobrych praktyk, nie narzuca sztywnych rozwiązań, lecz oferuje elastyczne ramy, które mogą być dostosowane do specyfiki danego projektu lub usługi. W efekcie jego zastosowanie w umowach IT sprzyja równowadze pomiędzy formalnymi wymaganiami prawa zobowiązań a potrzebą adaptacyjności i ciągłego doskonalenia charakterystyczną dla nowoczesnych usług informatycznych.

Obok ITIL istotne znaczenie dla kształtowania dobrych praktyk kontraktowych w umowach IT mają również **międzynarodowe normy oraz ramy zarządcze, w szczególności ISO**, a zwłaszcza ISO/IEC 20000, oraz COBIT. **Norma ISO/IEC 20000 stanowi formalny standard systemu zarządzania usługami IT** i w wielu aspektach pozostaje komplementarna wobec ITIL, oferując bardziej normatywne i audytowalne podejście do organizacji procesów usługowych. W kontekście kontraktowym odwołanie się do tej normy bywa wykorzystywane jako obiektywny miernik należytej staranności dostawcy usług oraz jako punkt odniesienia przy ocenie jakości realizowanych świadczeń. Z kolei **COBIT, jako rama zarządzania i nadzoru nad IT, znajduje zastosowanie przede wszystkim w umowach o charakterze strategicznym**, gdzie istotne są kwestie odpowiedzialności decyzyjnej, zgodności z regulacjami oraz kontroli ryzyka. Jego elementy mogą być wykorzystywane przy formułowaniu postanowień dotyczących ładu korporacyjnego, raportowania oraz odpowiedzialności kadry zarządzającej za funkcjonowanie systemów informatycznych.

Na uwagę zasługują również dobre praktyki wywodzące się z obszaru zarządzania projektami oraz wytwarzania oprogramowania, takie jak PMBOK, PRINCE2, a także praktyki DevOps i Agile/Scrum. Ramy te wpływają na treść umów IT poprzez kształtowanie sposobu planowania, realizacji i kontroli prac projektowych. PMBOK i PRINCE2 dostarczają uporządkowanych modeli zarządzania projektem, które sprzyjają precyzyjnemu określeniu ról, odpowiedzialności i kamieni milowych w kontraktach o charakterze klasycznym. Z kolei DevOps oraz Agile akcentują ciągłość dostarczania, automatyzację oraz ścisłą

współpracę zespołów technicznych i biznesowych, co przekłada się na potrzebę elastycznych zapisów umownych dotyczących zmian, odbiorów częściowych i odpowiedzialności za stabilność środowisk. **Wspólnym mianownikiem tych dobrych praktyk jest przesunięcie ciężaru umowy z samego rezultatu na proces współpracy, co w nowoczesnych umowach IT coraz częściej stanowi klucz do ich skuteczności i trwałości.**

6. Metodyka realizacji projektu a treść kontraktu

Realizacja projektów IT z wykorzystaniem metodyk zwinnych (*agile*) staje się w ostatnim czasie powszechnie obowiązującym standardem rynkowym, wypierając klasyczne, kaskadowe metody wdrożeniowe (*waterfall*). Charakterystyczny dla prac IT realizowanych w podejściu *agile* brak sztywnych zasad dotyczących planowania, odbioru oraz budżetowania projektu rodzi istotne wyzwanie dla osób przygotowujących oraz wykonujących umowy na tego rodzaju prace developerskie. Jednocześnie *agile* wprowadza wiele postanowień swoistych dla tego rodzaju metodyki, które powinny znaleźć swe odzwierciedlenie w umowie.

W modelu kaskadowym (*waterfall*) treść kontraktu IT opiera się na założeniu sekwencyjnej realizacji projektu, podzielonego na wyraźnie wyodrębnione etapy. Umowy zawierane w tej metodyce posługują się zazwyczaj pojęciami takimi jak: **etap, faza projektu, kamień milowy (*milestone*), harmonogram, dokumentacja projektowa czy odbiór etapu.** Kluczowe znaczenie mają **szczegółowo opisany zakres prac oraz precyzyjnie określony rezultat końcowy**, który podlega formalnej procedurze odbioru. Odpowiedzialność wykonawcy jest w tym modelu ściśle powiązana z dostarczeniem kompletnego i zgodnego ze specyfikacją efektu danego etapu, a rozliczenie finansowe następuje zazwyczaj po jego zatwierdzeniu. Kontrakt *waterfallowy* akcentuje stabilność zakresu oraz przewidywalność kosztów i terminów, co sprzyja jednoznacznemu przypisaniu ryzyk kontraktowych, ale jednocześnie ogranicza elastyczność w reagowaniu na zmiany biznesowe.

Odmienny charakter ma **metodyka zwinna (*agile*), w której projekt jest realizowany w sposób iteracyjny i przyrostowy**, a zakres funkcjonalny usługi lub produktu ulega stopniowej konkretyzacji w trakcie trwania współpracy. Umowy oparte na podejściu *agile* muszą uwzględniać specyficzne pojęcia oraz mechanizmy właściwe dla tej metodyki, takie jak **sprint, backlog, user story czy zespół *scrumowy*.**

Ważne

„Sprint” oznacza ograniczony w czasie, najczęściej kilkutygodniowy okres pracy, w ramach którego zespół realizuje określony zestaw zadań.

„Backlog” stanowi uporządkowany zbiór wymagań, funkcjonalności lub zadań przewidzianych do realizacji w projekcie, który może być na bieżąco modyfikowany w odpowiedzi na potrzeby biznesowe.

„User story” to opis funkcjonalności z perspektywy użytkownika końcowego, formułowany w sposób zwięzły i ukierunkowany na wartość biznesową, a nie na szczegółową specyfikację techniczną.

Z punktu widzenia kontraktowego istotne jest, że **w metodyce agile odpowiedzialność stron koncentruje się** nie tyle na dostarczeniu z góry określonego rezultatu końcowego, ile **na zapewnieniu właściwego przebiegu procesu wytwórczego, transparentności prac oraz regularnego dostarczania przyrostów funkcjonalnych**. Umowy *agile*owe często odchodzą od klasycznych procedur odbiorowych na rzecz mechanizmów akceptacji przyrostów realizowanych w ramach sprintów oraz bieżącej współpracy stron. W konsekwencji większego znaczenia nabierają postanowienia dotyczące zasad zarządzania *backlogiem*, ról projektowych (np. *product owner*), sposobu podejmowania decyzji o zmianach zakresu oraz rozliczeń opartych na czasie i zaangażowaniu zespołu, a nie na sztywno określonym efekcie końcowym.

Uzupełnieniem podstawowych pojęć metodyk zwinnych są dalsze terminy, które coraz częściej znajdują swoje odzwierciedlenie w treści umów IT. Do najistotniejszych należy **„product owner”**, rozumiany jako **osoba odpowiedzialna za maksymalizację wartości biznesowej produktu oraz za priorytetyzację backlogu**. W kontrakcie jego rola bywa powiązana z kompetencją do akceptacji efektów prac oraz podejmowania decyzji wiążących dla zamawiającego. Istotne znaczenie ma również pojęcie **„definition of done” (DoD)**, które określa **zestaw kryteriów, jakie musi spełniać dana funkcjonalność, aby została uznana za ukończoną**. W ujęciu kontraktowym pełni ono funkcję odpowiednika kryteriów odbioru, choć o bardziej elastycznym i iteracyjnym charakterze. Kolejnym kluczowym pojęciem jest **„increment”**, czyli **przyrost produktu dostarczany po zakończeniu sprintu**, który powinien przedstawiać samodzielną wartość użytkową i podlegać bieżącej ocenie przez zamawiającego. W praktyce umownej spotyka się także **pojęcie „velocity”**, odnoszące się do **tempa pracy zespołu**, wyrażonego ilością zrealizowanych elementów *backlogu* w danym sprincie. Choć ma ono charakter wskaźnika operacyjnego, bywa wykorzystywane pomocniczo przy planowaniu zakresu i budżetu. Całość tych pojęć podkreśla procesowy i współtwórczy charakter metodyki *agile*, w której treść kontraktu musi uwzględniać nie tylko rezultat prac, lecz również sposób ich organizacji, decyzyjność stron oraz mechanizmy akceptacji i zmiany zakresu.

W praktyce kontraktowej związanej z realizacją projektów IT w metodykach zwinnych pojawia się wiele dalszych pojęć, które – choć wywodzą się z praktyki projektowej – mają istotne znaczenie dla prawidłowego ukształtowania praw i obowiązków stron. Do takich pojęć należy **„minimum viable product” (MVP)**, rozumiany jako **minimalna wersja produktu umożliwiająca jego faktyczne użycie i weryfikację założeń biznesowych**. W umowach IT MVP bywa traktowane jako pierwszy punkt odniesienia dla oceny postępów projektu lub jako etap decyzyjny co do dalszego finansowania prac. Istotne znaczenie ma również **„release”**, czyli **wydanie obejmujące zestaw ukończonych przyrostów, które mogą zostać wdrożone do środowiska produkcyjnego**. W ujęciu kontraktowym *release* często wiąże się z określonymi skutkami finansowymi, odbiorowymi lub gwarancyjnymi.

Kolejnym pojęciem jest **„refinement backlogu” (ang. backlog refinement)**, oznaczający **cykliczny proces doprecyzowywania i porządkowania elementów backlogu**. Z perspektywy umownej ma on znaczenie jako mechanizm wspólnego uzgadniania zakresu prac i ograniczania ryzyka sporów dotyczących niejednoznaczności wymagań. W tym kontekście pojawia się także termin **„estymacja”**, odnoszący się do szacowa-

nia nakładu pracy niezbędnego do realizacji poszczególnych elementów *backlogu*. W metodykach zwinnych estymacja ma charakter przybliżony, co wymaga od umów IT odejścia od sztywnego budżetowania na rzecz modeli rozliczeń opartych na czasie lub zaangażowaniu zespołu.

Warto również wskazać pojęcie „*time and materials*” (T&M), często stosowane jako model rozliczeniowy w projektach *agile*, w którym **wynagrodzenie wykonawcy uzależnione jest od rzeczywistego czasu pracy i wykorzystanych zasobów**. Wymaga ono odpowiednich postanowień umownych dotyczących raportowania prac, limitów budżetowych oraz zasad kontroli kosztów. Z kolei „*continuous integration*” oraz „*continuous delivery*” opisują praktyki techniczne polegające na częstym integrowaniu kodu oraz gotowości do jego regularnego wdrażania. Ich obecność w kontrakcie bywa powiązana z wymaganiami jakościowymi, automatyzacją testów oraz odpowiedzialnością za stabilność środowisk.

Istotne znaczenie kontraktowe mają także pojęcia związane z komunikacją i współpracą stron, takie jak „ceremonie *scrumowe*” (np. *sprint review*, *sprint retrospective*), które w umowach mogą pełnić funkcję formalnych punktów kontrolnych projektu. *Sprint review* umożliwia ocenę dostarczonych przyrostów i bieżącą akceptację efektów prac, natomiast *retrospective* służy doskonaleniu sposobu współpracy i procesów projektowych. Uzupełnieniem tych pojęć jest „*self-organizing team*”, czyli zespół samoorganizujący się, którego autonomia w zakresie sposobu realizacji prac wymaga odpowiedniego rozłożenia odpowiedzialności pomiędzy zamawiającego a wykonawcę.

Wszystkie wskazane pojęcia potwierdzają, że metodyki zwinne wprowadzają do umów IT język oraz mechanizmy odmienne od klasycznych modeli kaskadowych.

Ważne

Kontrakt w podejściu *agile* nie może ograniczać się wyłącznie do opisu rezultatu końcowego, lecz musi regulować proces współpracy, podejmowania decyzji, zarządzania zmianą oraz akceptacji przyrostów.

Brak precyzyjnego ujęcia tych elementów w umowie może prowadzić do niepewności interpretacyjnej oraz zwiększonego ryzyka sporów, mimo że sama metodyka *agile* zakłada elastyczność i adaptacyjność realizacji projektu.

Dobór metodyki realizacji projektu IT ma więc zatem bezpośredni wpływ na treść i konstrukcję kontraktu. Podczas gdy umowy *waterfallowe* sprzyjają precyzyjnemu opisowi świadczeń i odpowiedzialności stron, podejście *agile* wymaga większego nacisku na elastyczność, mechanizmy współpracy oraz podział ryzyk związanych z ewolucją wymagań. Świadome dostosowanie postanowień umownych do wybranej metodyki stanowi jeden z kluczowych warunków prawidłowej realizacji projektu IT oraz ograniczenia sporów na tle jego wykonania.

Podstawa prawna:

- Data Act,
- KC,
- PrAut,
- RODO.

Mateusz Kozieł

Rozdział II. Umowy wdrożeniowe (projektowe)

1. Uwagi ogólne

W przypadku klasycznego przebiegu projektu informatycznego, naturalnym oraz intuicyjnym modelem wydaje się sekwencja, w której rozpoczyna się on od wdrożenia oprogramowania, następnie przechodzi w etap utrzymania efektów wdrożenia, a na końcu następuje faza rozwoju oprogramowania w celu dostosowania go do dalszego cyklu „życia” programu komputerowego.

Taka oparta na swoistym trójpodziale struktura umożliwi uporządkowanie projektu informatycznego w systematyczny sposób, a z uwagi na wieloletnie ugruntowanie w praktyce projektowej, zapewnia osobom biorącym udział w projekcie (niezależnie od ich roli jako osoby technicznej, właściciela biznesowego, prawnika itp.) poczucie kontroli nad każdym etapem prac.

W praktyce jednak znaczna liczba przedsięwzięć IT zawiera mniejsze lub większe odstępstwa od tego klasycznego porządku, wynikające chociażby ze specyfiki danego typu oprogramowania lub rynku na jakim ono funkcjonuje, zmiennych potrzeb natury biznesowej czy ewoluujących założeń projektowych. Pamiętać bowiem trzeba, że umowy regulujące projekty informatyczne to „(...) złożone i specjalistyczne kontrakty, które wymykają się prostym klasyfikacjom. Znajdziemy w nich elementy umowy o dzieło, świadczenia usług, a także – często skomplikowane i niejednorodne – postanowienia umów z zakresu prawa autorskiego: umów licencyjnych i umów przeniesienia majątkowych praw autorskich. Stopień skomplikowania umów IT wynika nie tylko z zakresu świadczeń wykonawcy, lecz także różnorodności przedmiotów i uprawnień, które wzajemnie się przenikają”¹.

¹ P. Wasilewski, Licencjonowanie i prawa autorskie w umowach IT – aspekty praktyczne, LEX/el. 2021.

Nie bez znaczenia dla wskazanych zmian tendencji pozostaje również szerokie stosowanie w ostatnim czasie tzw. zwinnych metod prowadzenia projektów. W szybko zmieniającym się świecie projektów informatycznych metodyki zwinne *agile* zyskały szeroką popularność przede wszystkim dzięki dwóm zasadniczym cechom: elastyczności (łatwej adaptacji do zmian) i podejściu opartemu na współpracy. To cechujące metodykę *agile* bieżąca współpraca – zarówno wewnątrz zespołów deweloperskich, jak również między zespołami programistycznymi i interesariuszami – oraz iteracyjny, inkrementacyjny² rozwój oprogramowania z jednoczesnym zapewnieniem realnej wartości biznesowej, niemal na każdym etapie projektu powodują, że granica między poszczególnymi etapami projektu informatycznego coraz częściej się zaciera lub jest na tyle płynna, że w sposób naturalny przechodzi on z jednego jego etapu w drugi, bez jakichkolwiek formalności, w ramach jednej umowy wiążącej strony.

Nie oznacza to jednak, że powinniśmy całkowicie odstępować od analizy klasycznego przebiegu projektu IT, składającego się z trzech głównych faz: wdrożenia, utrzymania oraz rozwoju. Wymienione etapy mają swoje specyficzne cele, zadania oraz wymagania, które wpływają na przebieg całego projektu i każdemu z tych etapów należy się na początek nieco dokładniej przyjrzeć.

2. Definicja i cel umów wdrożeniowych

2.1. W poszukiwaniu legalnej definicji

W polskim oraz unijnym porządku prawnym nie sposób doszukać się legalnej definicji wdrożenia czy też umowy wdrożeniowej w *stricto* informatycznym kontekście i rozumieniu tych pojęć, choć same w sobie nie pozostają one obce tekstom obowiązujących aktów prawnych³.

Jednocześnie proces wdrożenia oprogramowania oraz jego poszczególne etapy są w powszechnie obowiązujących aktach prawnych coraz częściej dostrzegane i normowane wprost językiem ustawodawcy, zwłaszcza w zakresie regulacji o charakterze szczegółowym, stawiających określone wymagania dotyczące np. względów bezpieczeństwa czy też obronności⁴.

² Inkrementacja oprogramowania czy też metodyka inkrementacyjnego programowania to podejście do opracowywania oprogramowania, w którym jest ono budowane i dostarczane stopniowo, w małych częściach, zwanych przyrostami (czy też właśnie inkrementami). Każdy kolejny przyrost w procesie tworzenia programu dodaje nową wartość dodaną do wcześniejszych efektów prac, skutkując ostatecznie zakończeniem projektu.

³ Zob. art. 2 pkt 19 OchrInfU zawierający ustawową definicję rozwiązania informatycznego, którym jest urządzenie lub zespół urządzeń, oprogramowanie, narzędzie lub usługa informatyczna umożliwiająca przetwarzanie informacji niejawnych w postaci elektronicznej, eksploatowana lub planowana do wdrożenia wyłącznie we wskazanych w OchrInfU jednostkach organizacyjnych. Pojęciem wdrożenia operują również przepisy RozwInformR, wydanego na podstawie art. 51 ust. 6 OchrInfU.

⁴ W tym zakresie warto przede wszystkim zwrócić uwagę na przepisy InformPodPublU, a zwłaszcza art. 21 InformPodPublU wskazujący na zasady przeprowadzenia badania poprawności wdrożenia rozwiązań dot. struktur dokumentów elektronicznych, formatów danych oraz protokołów komunikacyjnych i szyfrujących stosowanych w oprogramowaniu interfejsowym przy wykorzystaniu testów akceptacyjnych. Same testy akceptacyjne art. 3 pkt 12 InformPodPublU definiuje jako udokumentowane wartości danych wejściowych wprowa-

Nie posiadając jednak jakiegokolwiek bazy pojęciowej w aktach normatywnych, warto poszukać zrozumienia pojęcia wdrożenia w praktyce technicznej oraz biznesowej.

2.2. Próba definicji poprzez dekompozycję pakietu świadczeń (około)wdrożeńowych

Z perspektywy technicznej oraz biznesowej „wdrożenie” to początkowy etap projektu, który koncentruje się na dostarczeniu realnie funkcjonującego produktu IT, poprzez działania dążące do jego implementacji, a następnie uruchomienia. Faza ta opiera się w przeważającej mierze na zrozumieniu potrzeb użytkownika końcowego i dokładnej koordynacji każdego elementu prac. **Obejmuje w szczególności projektowanie, wybór technologii i metodyki prowadzenia projektu oraz realizację rozwiązania i integrację z innymi systemami.** To właśnie na etapie wdrożenia odbywają się prace programistyczne, konfiguracyjne oraz testowe, mające na celu wyeliminowanie błędów i zapewnienie, że rozwiązanie zadziała zgodnie z oczekiwaniami. Testowanie i przygotowanie do uruchomienia systemu kończą etap wdrożenia, a jego sukces tworzy podstawy pod dalsze fazy projektu.

Próbując jak najlepiej zdefiniować i zrozumieć pojęcie „wdrożenie”, warto rozłożyć na czynniki pierwsze pakiet standardowych czynności mogących wchodzić w jego zakres. Z technicznego punktu widzenia, zawierać on będzie wiele świadczeń odpowiadających na różnorodne potrzeby projektowe, takich jak:

- 1) analiza potrzeb użytkownika;
- 2) wymiarowanie zasobów i infrastruktury;
- 3) projektowanie rozwiązania;
- 4) opracowanie planu i harmonogramu projektu;
- 5) prace programistyczne polegające na tworzeniu kodu oprogramowania, w tym kodu oprogramowania dedykowanego;
- 6) prace programistyczne polegające na konfiguracji już istniejącego oprogramowania i/lub dostosowaniu go do potrzeb użytkownika;
- 7) prace programistyczne polegające na integracji już istniejącego oprogramowania z innym oprogramowaniem;
- 8) testowanie i odbiór (akceptacja) oprogramowania;
- 9) *deployment* oprogramowania, czyli przeniesienie oprogramowania do jakiegokolwiek środowiska programistycznego, w tym środowiska produkcyjnego;
- 10) start produkcyjny, czyli rozpoczęcie korzystania z oprogramowania (tzw. *go-live*);
- 11) wstępna faza gwarancyjna/utrzymeniowa, czyli reakcja na błędy i problemy powstające tuż po starcie produkcyjnym, jeszcze w ramach krótkiego etapu powdrożeniowego, wchodzącego jednak w zakres i harmonogram całego wdrożenia.

Z powyższej (zaznaczmy: wyłącznie skrótowej i przykładowej) listy możliwych świadczeń wchodzących w „pakiet” wdrożeniowy, wyłania się obraz wysoce skomplikowanego

dzanych do systemu teleinformatycznego i powiązanych z nimi wartości oczekiwanych danych wyjściowych, opisujące zestawy poprawnych odpowiedzi systemu teleinformatycznego na podawane dane wejściowe, pozwalające na sprawdzenie poprawności wdrożenia oprogramowania interfejsowego.

[Przejdź do księgarni →](#)

ksiegarnia.beck.pl