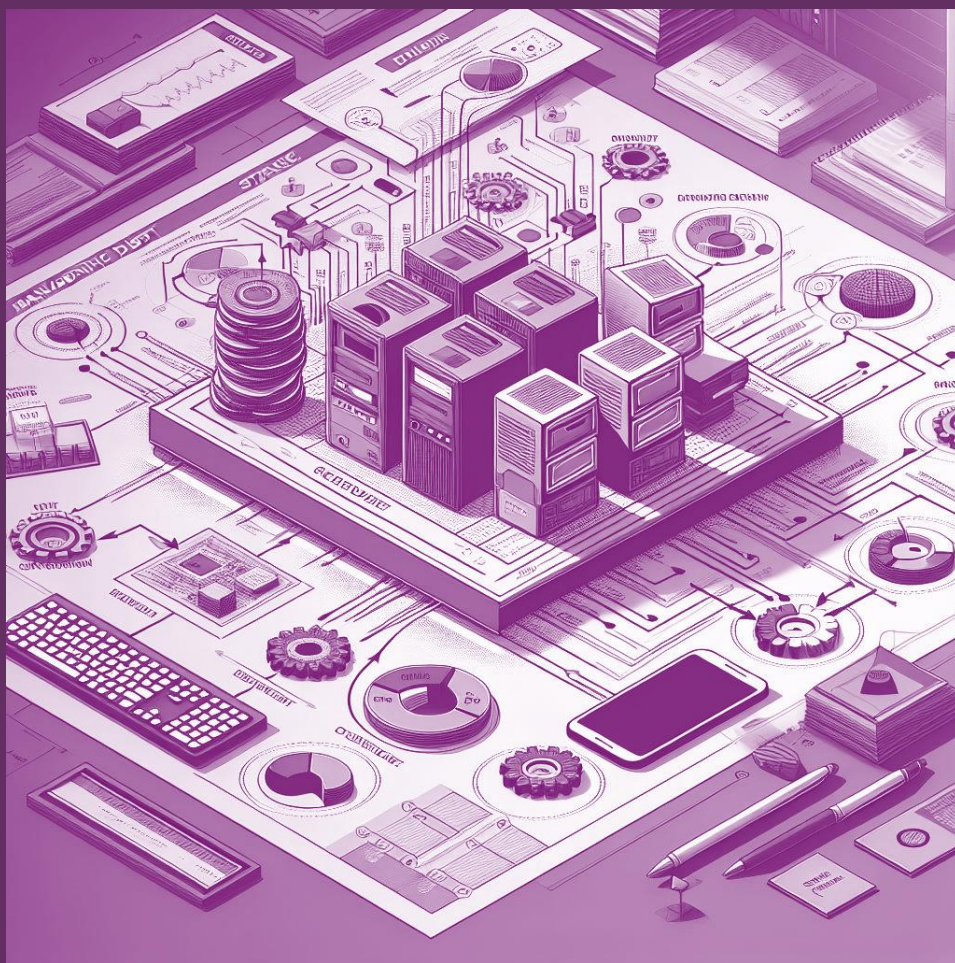




STRATEGIA POMIARU I ZARZĄDZANIA DŁUGIEM TECHNOLOGICZNYM I ARCHITEKTONICZNYM W FIKCYJNEJ FIRMIE UBEZPIECZENIOWEJ XYZ

Materiał wygenerowany w 100% przez AI

Wprowadzenie

Celem dokumentu jest przedstawienie kompleksowej strategii identyfikacji, pomiaru oraz zarządzania długiem technologicznym i architektonicznym w dużej firmie ubezpieczeniowej. Strategia ta uwzględnia kontekst regulacyjny (krajowy i unijny), definiuje pojęcia długu technologicznego i architektonicznego (z uwzględnieniem różnych typów systemów), opisuje metody pomiaru i monitorowania zadłużenia technologicznego, analizuje związane z nim ryzyka oraz przedstawia plan realizacji przeglądu długu i komunikacji wyników wraz z planem działań naprawczych. Dokument bazuje na najlepszych praktykach rynkowych oraz doświadczeniach międzynarodowych firm doradczych.

1. Przegląd regulacji i wytycznych

Znajomość otoczenia regulacyjnego jest kluczowa dla właściwego zarządzania długiem technologicznym w sektorze finansowym. Firmy ubezpieczeniowe podlegają zarówno polskim regulacjom dotyczącym infrastruktury krytycznej i sektora finansowego, wytycznym krajowego nadzoru (KNF), jak i regulacjom unijnym w obszarze odporności cyfrowej i cyberbezpieczeństwa. Poniżej przedstawiono najważniejsze ramy regulacyjne i wytyczne, które należy uwzględnić:

1.1 Polskie regulacje dot. infrastruktury krytycznej i sektora finansowego

- **Infrastruktura krytyczna i cyberbezpieczeństwo:** Ustawa o krajowym systemie cyberbezpieczeństwa z 5 lipca 2018 r. identyfikuje *operatorów usług kluczowych* (OUK) w sektorach istotnych dla funkcjonowania gospodarki, m.in. sektorze finansowym ([Operatorzy usług kluczowych - Ministerstwo Cyfryzacji - Portal Gov.pl](#)). Duża firma ubezpieczeniowa może zostać uznana za OUK, jeśli świadczy usługi o kluczowym znaczeniu (np. ubezpieczenia komunikacyjne czy na życie na szeroką skalę) zależne od systemów informatycznych. Status ten wiąże się z obowiązkami: wdrożeniem systemu zarządzania bezpieczeństwem informacji, regularną analizą ryzyka cyber, utrzymywaniem planów ciągłości działania oraz zgłaszaniem poważnych incydentów bezpieczeństwa właściwym organom. Spełnienie tych wymogów pośrednio wymaga kontroli nad długiem technologicznym – przestarzałe, podatne systemy zwiększają ryzyko incydentów i mogą naruszać wymogi ustawy.
- **Wymogi regulacyjne dla instytucji finansowych:** Poza prawem dotyczącym cyberbezpieczeństwa, firmy ubezpieczeniowe muszą przestrzegać szeregu aktów prawnych regulujących sektor finansowy, np. ustawy o działalności ubezpieczeniowej i reasekuracyjnej, które nakładają wymogi co do zarządzania ryzykiem operacyjnym i ciągłością działania. Infrastruktura teleinformatyczna obsługująca usługi ubezpieczeniowe powinna być traktowana jako element krytyczny – jej awaria mogłaby mieć skutki społeczne (np. opóźnienie wypłat odszkodowań). Dlatego organy nadzoru oczekują od ubezpieczycieli utrzymania należytej *stabilności i aktualności* środowiska IT.

1.2 Rekomendacje KNF i wytyczne dot. IT i ryzyka technologicznego

- **Rekomendacja D KNF:** Komisja Nadzoru Finansowego (KNF) wydała w 2013 r. Rekomendację D dotyczącą zarządzania obszarem technologii informacyjnej i bezpieczeństwa środowiska teleinformatycznego w bankach. Stanowiła ona **fundament zarządzania bezpieczeństwem IT** w sektorze finansowym, ustanawiając wymagania m.in. w zakresie governance IT, rozwoju, utrzymania i bezpieczeństwa systemów ([Rekomendacja D KNF - szansa czy zagrożenie dla EA? | Architektura Korporacyjna](#)). Banki miały czas do końca 2014 r. na wdrożenie zaleceń Rekomendacji D, a KNF wydał *bliźniacze regulacje dla zakładów ubezpieczeń i reasekuracji* oraz innych instytucji finansowych ([Czy dojdzie do uchylecia komunikatu chmurowego i wytycznych IT przez KNF?](#)). Oznacza to, że również ubezpieczyciele powinni posiadać formalne polityki zarządzania IT, procesy zarządzania zmianą, analizę ryzyka IT, plany ciągłości działania itp., analogicznie do sektora bankowego. Rekomendacja D zwracała uwagę m.in. na potrzebę *podejścia architektonicznego* w zarządzaniu IT – po raz pierwszy w dokumencie nadzorczym wspomniano o architekturze IT jako obszarze wymagającym uwagi zarządczej ([Rekomendacja D KNF - szansa czy zagrożenie dla EA? | Architektura Korporacyjna](#)). Wprowadziła też obowiązek

przeprowadzenia analizy luk (gap analysis) i opracowania planu doskonalenia obszaru IT ([Rekomendacja D KNF - szansa czy zagrożenie dla EA? | Architektura Korporacyjna](#)).

- **Inne wytyczne nadzorcze KNF:** KNF wydał również Rekomendację M dotyczącą zarządzania ryzykiem operacyjnym (pierwotnie dla banków), która obejmuje identyfikację i kontrolę ryzyk operacyjnych, w tym ryzyk związanych z technologią. Ponadto obowiązywały tzw. *Wytyczne dotyczące zarządzania obszarem technologii informacyjnej i bezpieczeństwa środowiska teleinformatycznego* (KNF, 2017) oraz tzw. *komunikat chmurowy UKNF* (pierwsza wersja 2017, aktualizacja 2020) dotyczący korzystania z usług chmury obliczeniowej przez instytucje finansowe. Te akty nadzorcze (choć miały charakter „soft law”) precyzowały oczekiwania odnośnie m.in. kontroli zmian w systemach, zarządzania architekturą, utrzymywania aktualnych wersji oprogramowania oraz zasad outsourcowania IT (np. do chmury). **Dług technologiczny** skutkujący np. brakiem wsparcia dla starych systemów, niekontrolowanymi rozwiązaniami użytkowników (EUC) czy zaległościami w aktualizacjach – stoi w sprzeczności z tymi wytycznymi. Zarząd nadzorowanej instytucji ponosi odpowiedzialność za zapewnienie zgodności z rekomendacjami, co wymaga monitorowania stanu IT i planowania działań modernizacyjnych.
- **Ewolucja wymogów – wpływ nowych regulacji:** W ostatnich latach pojawiły się nowe regulacje wpływające na zarządzanie IT w finansach: m.in. PSD2 (wymogi silnego uwierzytelniania), RODO (ochrona danych osobowych) czy właśnie ustawa o krajowym systemie cyberbezpieczeństwa ([Czy dojdzie do uchylecia komunikatu chmurowego i wytycznych IT przez KNF?](#)). Wiele z dotychczasowych krajowych wytycznych (jak Rekomendacja D czy komunikat chmurowy) ma zostać dostosowanych lub uchylonych w związku z wejściem w życie unijnych regulacji DORA i NIS2 ([Czy dojdzie do uchylecia komunikatu chmurowego i wytycznych IT przez KNF?](#)) ([Czy dojdzie do uchylecia komunikatu chmurowego i wytycznych IT przez KNF?](#)). KNF sygnalizuje, że europejskie przepisy ustanowią nowe standardy zarządzania ryzykiem ICT, które zastąpią krajowe soft law. Oznacza to konieczność śledzenia zmian prawa i być może aktualizację wewnętrznych polityk IT firmy ubezpieczeniowej zgodnie z nowymi wymaganiami.

1.3 Unijne regulacje: DORA i NIS2

- **Rozporządzenie DORA (Digital Operational Resilience Act):** Jest to rozporządzenie UE przyjęte w 2022 r., mające na celu wzmocnienie *cyfrowej odporności operacyjnej sektora finansowego*. DORA wprowadza jednolite wymogi w całej UE dotyczące zarządzania ryzykiem ICT w bankach, ubezpieczycielach, firmach inwestycyjnych i innych instytucjach finansowych. Kładzie nacisk na **rygorystyczne zarządzanie ryzykiem ICT**, obowiązkowe testy odporności (np. testy penetracyjne typu TLPT dla największych podmiotów), monitorowanie incydentów i zgłaszanie incydentów poważnych do nadzoru ([Rozporządzenie DORA a Dyrektywa NIS2: Kluczowe różnice w obszarze cyberbezpieczeństwa](#)). DORA obejmuje **wszystkie kluczowe instytucje finansowe** – w tym zakłady ubezpieczeń – zobowiązując je m.in. do posiadania strategii operacyjnej odporności cyfrowej, regularnego przeglądu ryzyka technologicznego, oceny dostawców zewnętrznych IT oraz planów awaryjnych. W praktyce, aby spełnić DORA, firma ubezpieczeniowa musi mieć pełen obraz swojego środowiska IT, w tym **zidentyfikowany i oszacowany dług technologiczny i architektoniczny**, który może wpływać na operacyjną odporność. Np. rozporządzenie wymaga, by instytucje znały *klasy krytyczności swoich systemów* i miały plany ich odtwarzania – systemy obciążone dużym długiem mogą nie spełniać wymogów dostępności czy bezpieczeństwa stawianych przez DORA.
- **Dyrektywa NIS2:** Jest to kolejna unijna regulacja (następca dyrektywy NIS) dotycząca cyberbezpieczeństwa sieci i systemów informacyjnych, przyjęta w 2022 r. NIS2 rozszerza zakres poprzednich przepisów na *szersze spektrum sektorów gospodarki*, uznawanych za kluczowe dla państwa – obejmuje m.in. energetykę, transport, ochronę zdrowia, infrastrukturę cyfrową, a także sektor **bankowy i rynki finansowe** ([Rozporządzenie DORA a Dyrektywa NIS2: Kluczowe różnice w obszarze cyberbezpieczeństwa](#)) ([Rozporządzenie DORA a Dyrektywa NIS2:](#)

[Kluczowe różnice w obszarze cyberbezpieczeństwa](#)). Wymaga od średnich i dużych przedsiębiorstw w tych sektorach wdrożenia odpowiednich środków bezpieczeństwa ICT, zarządzania ryzykiem i raportowania incydentów. Dla firm ubezpieczeniowych NIS2 ma znaczenie o tyle, że **ubezpieczenia** są powiązane z sektorem finansowym; jednak instytucje objęte DORA prawdopodobnie będą wyłączone spod NIS2 (DORA działa jako *lex specialis* dla finansów ([Rozporządzenie DORA a Dyrektywa NIS2: Kluczowe różnice w obszarze cyberbezpieczeństwa](#))). Niemniej jednak, zasady NIS2 – takie jak podejście oparte na analizie ryzyka, obowiązek posiadania procedur zarządzania ciągłością działania i planów reagowania na incydenty – są spójne z wymaganiami DORA. Kluczowe jest zrozumienie, że **DORA koncentruje się na sektorze finansowym, a NIS2 obejmuje szeroko infrastrukturę krytyczną**; wspólnym mianownikiem jest konieczność posiadania przejrzystego obrazu ryzyk technologicznych. Spełnienie NIS2 i DORA będzie wymagać od ubezpieczycieli m.in. inwentaryzacji zasobów IT, oceny luk w zabezpieczeniach, planów modernizacji przestarzałych systemów – czyli *de facto* zarządzania długiem technologicznym.

Podsumowanie wymogów regulacyjnych: Duża firma ubezpieczeniowa musi prowadzić zarządzanie IT w zgodzie z ww. regulacjami. To oznacza **regularne przeglądy stanu architektury i technologii**, identyfikowanie słabych punktów (np. przestarzałych systemów, brakujących aktualizacji, niekontrolowanych narzędzi biznesowych) oraz planowanie ich usprawnienia. Zarządzanie długiem technologicznym powinno być traktowane nie tylko jako kwestia efektywności IT, ale jako element **zgodności z regulacjami i zarządzania ryzykiem operacyjnym**. W dalszych częściach dokumentu opisano, jak zdefiniować taki dług, jak go mierzyć i redukować, by spełnić zarówno wewnętrzne cele biznesowe, jak i zewnętrzne wymagania regulacyjne.

2. Definicja długu technologicznego i architektonicznego

Strategia zarządzania zadłużeniem technologicznym rozpoczyna się od jasnego zdefiniowania pojęć **długu technologicznego** oraz **długu architektonicznego**. W literaturze pojęcia te bywają używane zamiennie z terminem *dług techniczny*, jednak w niniejszym dokumencie dokonujemy rozróżnienia, co pozwoli lepiej kategoryzować problemy w dużej organizacji. Ponadto uwzględniamy różne typy systemów i rozwiązań (od systemów legacy, przez narzędzia tworzone przez użytkowników biznesowych, aż po nowoczesne rozwiązania AI czy chmurowe), które mogą generować dług w różnych komponentach: w kodzie, infrastrukturze lub w procesach. Poniżej przedstawiono definicje i klasyfikacje:

2.1 Dług technologiczny – pojęcie i zakres

Dług technologiczny (często zwany też technicznym) to *hipotetyczne zobowiązanie*, jakie organizacja zaciąga wobec własnej infrastruktury IT, gdy podejmuje świadome lub nieświadome decyzje upraszczające lub odraczające pewne prace techniczne kosztem przyszłej złożoności ([Czym jest dług architektoniczny? | Architektura Korporacyjna](#)). Klasyczna definicja Warda Cunninghama mówi, że powstaje on, gdy „budujemy rozwiązanie informatyczne na skróty, wiedząc, że w przyszłości trzeba będzie wrócić i wykonać tę pracę jeszcze raz” ([Dług technologiczny a robotyzacja procesów biznesowych | Robonomika - Robotyzacja biznesu i jej okolice](#)). Innymi słowy, **cokolwiek czyni system trudnym w modyfikacji lub utrzymaniu w przyszłości, stanowi dług technologiczny** ([Dług technologiczny a robotyzacja procesów biznesowych | Robonomika - Robotyzacja biznesu i jej okolice](#)).

Najczęściej dług technologiczny kojarzony jest z jakością kodu źródłowego – np. nieoptymalnym, zbyt skomplikowanym kodem, brakami w dokumentacji czy testach. Jednak pojęcie to jest szersze. Jak zauważają eksperci, *dług technologiczny to nie tylko pójście na skróty podczas kodowania, to również zaniechanie przechodzenia na najnowsze wersje narzędzi programistycznych oraz niewspieranie najnowszych systemów operacyjnych* ([Dług technologiczny a robotyzacja procesów biznesowych | Robonomika - Robotyzacja biznesu i jej okolice](#)). A zatem pozostawienie systemów na przestarzałych platformach (np. aplikacja działa w starej wersji Java czy .NET, dla której skończyło się wsparcie) także generuje dług. Dług technologiczny może narastać również wskutek **braku refaktoryzacji** – gdy przez

lata dokłada się do systemu nowe łatki i funkcje, nie porządkując struktury, co prowadzi do tzw. *spaghetti code*.

Przykłady długu technologicznego w firmie ubezpieczeniowej mogą obejmować m.in.:

- moduł wyceny składek napisany dawno temu, trudny do modyfikacji z powodu zagmatwanej logiki i braku testów;
- wykorzystywanie **fragile spreadsheets** (delikatnych, podatnych na błąd arkuszy kalkulacyjnych) do kalkulacji aktuarialnych ([Tackling technical debt in the insurance industry](#)) – ich istnienie poza kontrolą IT powoduje, że poprawki i aktualizacje są ryzykowne;
- *quick fixes* zaimplementowane naprędce w systemie szkodowym, które działają, ale zwiększają złożoność (np. prowizoryczne skrypty integracyjne).

Wszystkie powyższe sprawiają, że wprowadzenie zmian w systemach jest kosztowniejsze i wolniejsze – co jest „kosztem odsetek” od długu technicznego, jaki płacimy z czasem. **Dług technologiczny ma charakter lokalny** – często odnosi się do konkretnego systemu, modułu lub aplikacji. Może on być względnie niewielki (np. nieoptymalny fragment kodu) lub bardzo duży (np. cała aplikacja wymaga przepisania od nowa). Warto zauważyć, że pewien poziom długu technicznego jest nieunikniony i akceptowalny – podobnie jak firma może mieć pewne zadłużenie finansowe, o ile jest ono pod kontrolą. Problem pojawia się, gdy dług wymyka się spod kontroli i jego „obsługa” (np. nakłady pracy na utrzymanie starych rozwiązań) zaczyna hamować rozwój firmy.

Systemy typu legacy – czyli starszej generacji, odziedziczone systemy działające nierzadko na przestarzałych technologiach – są głównym źródłem długu technologicznego. W branży ubezpieczeniowej wiele kluczowych procesów opiera się na systemach core (np. polisowych czy szkodowych) tworzonych kilkanaście lub więcej lat temu. Ich modernizacja była odkładana, co oznacza, że z biegiem lat narosła „ukryta zaległość” prac modernizacyjnych do wykonania. Raport McKinsey pokazał, że CIO szacują wartość długu technicznego na **20–40% wartości całego majątku technologicznego firmy** ([Tech debt: Reclaiming tech equity | McKinsey](#)). Oznacza to, że znacząca część środowiska IT wymaga usprawnień. Dług techniczny przejawia się też w **czasie i budżecie** – średnio 10–20% budżetu przeznaczanego na nowe inicjatywy technologiczne jest zużywane na rozwiązywanie problemów związanych z długiem (np. prace naprawcze, obejścia) ([Tech debt: Reclaiming tech equity | McKinsey](#)).

Podsumowując, dług technologiczny to *mierzalna zaległość technologiczna* – np. w postaci koniecznych aktualizacji, refaktoryzacji kodu czy usprawnień wydajności – której nierozwiązanie **obniża zwinność i stabilność IT**. Jego przeciwieństwem jest **utrzymanie wysokiej jakości technicznej** systemów na bieżąco. Kolejna sekcja opisuje dług architektoniczny, który dotyczy szerszego poziomu niż pojedyncze aplikacje.

2.2 Dług architektoniczny – pojęcie i zakres

Dług architektoniczny odnosi się do **strategicznego poziomu IT** i powstaje w wyniku decyzji projektowych dotyczących całej architektury rozwiązań w organizacji (bądź jej istotnej części), które odbiegają od optymalnych założeń lub docelowych standardów. Można go zdefiniować jako *zobowiązanie organizacji wynikające ze sposobu realizacji i integracji określonych komponentów rozwiązania, które nie jest zgodne z przyjętymi pryncypiami architektonicznymi ani docelowym modelem architektury* ([Czym jest dług architektoniczny? | Architektura Korporacyjna](#)). Innymi słowy, gdy firma rozwija swoje systemy w sposób niespójny z planowaną architekturą korporacyjną – np. wprowadza ad-hoc nowe technologie, omija ustalone standardy integracji czy duplikuje funkcjonalności w różnych systemach – zaciąga dług architektoniczny.

Dług architektoniczny jest **bardziej abstrakcyjny i szeroki** niż technologiczny ([Czym jest dług architektoniczny? | Architektura Korporacyjna](#)). O ile dług technologiczny dotyczy typowo jednego

systemu i jest czysto techniczny (kod, narzędzie, konfiguracja), o tyle dług architektoniczny obejmuje **cały ekosystem aplikacji i procesów biznesowych**. Przykłady długu architektonicznego:

- **Silosy danych:** firma posiada wiele rozproszonych baz danych z danymi klientów w różnych systemach, zamiast jednej spójnej hurtowni – utrudnia to uzyskanie pełnego obrazu klienta i wymaga dodatkowych integracji.
- **Niekontrolowane narzędzia biznesowe (Shadow IT):** departamenty biznesowe tworzą własne rozwiązania (np. aplikacje Access, Excel – tzw. EUC) poza oficjalną architekturą. Powstaje równoległy krajobraz systemów, którego IT nie kontroluje.
- **Odejście od standardów integracji:** np. organizacja przyjęła architekturę z ESB (szyną usług) jako standard, ale jedna z jednostek wprowadziła integrację poprzez bezpośrednie połączenia baz danych czy pliki wymiany FTP – tym samym łamiąc spójność architektury ([Dług technologiczny a robotyzacja procesów biznesowych | Robonomika - Robotyzacja biznesu i jej okolice](#)). Taka „łatana” integracja będzie później wymagała kosztownego uporządkowania.
- **Brak zgodności z architekturą docelową:** firma mogła zaplanować docelową architekturę (Target Architecture), np. z określonymi modułami (sprzedaż, obsługa polis, szkody) i kanałami (portal agentów, aplikacja mobilna) oraz zasadami (np. *single source of truth* dla danych). Jeśli jednak rozwój systemów odbywał się bez kierowania się tym planem – np. rozwijano równolegle dwa systemy o pokrywającej się funkcjonalności, albo zaniechano integracji pewnych modułów – to powstała *luka architektoniczna*.

Dług architektoniczny często wynika z historycznych decyzji biznesowych – np. przejęcia innej firmy (i pozostawienie dwóch osobnych platform IT) albo presji czasu, gdy wdrożono szybkie rozwiązanie nie skalujące się na przyszłość. **Konsekwencje długu architektonicznego** mogą być poważniejsze niż technologicznego, bo dotyczą fundamentalnych możliwości organizacji: „*gdy mówimy o decyzjach architektonicznych, dług może być olbrzymi i pociągać konieczność przerabiania kiedyś całych systemów*” ([Dług technologiczny a robotyzacja procesów biznesowych | Robonomika - Robotyzacja biznesu i jej okolice](#)).

Charakterystyczne jest też to, że dług architektoniczny obejmuje zarówno **wymiar IT, jak i procesów biznesowych** ([Czym jest dług architektoniczny? | Architektura Korporacyjna](#)). Przykładowo, brak centralizacji danych klienta to problem IT, ale skutkuje on również nieefektywnością procesów obsługi klienta (trzeba szukać informacji w wielu miejscach). Dlatego dług architektoniczny łączy się z **długiem procesowym** – np. niezainformatyzowane, ręczne procesy obiegu dokumentów mogą być postrzegane jako „zaległość” w transformacji cyfrowej firmy.

W odróżnieniu od długu technologicznego, który bywa widoczny dla inżynierów na poziomie kodu, dług architektoniczny ujawnia się na poziomie **strategicznych analiz IT** – np. w przeglądzie architektury korporacyjnej czy podczas planowania roadmapy IT wychodzi na jaw, że pewne obszary są do gruntownej przebudowy. Badania wskazują, że ten rodzaj długu będzie rosnąć – Gartner prognozuje, że do 2026 roku aż 80% całkowitego długu technicznego przedsiębiorstw stanowić będzie dług architektoniczny ([Technical Debt vs. Architectural Technical Debt](#)). Jest to efektem narastającej złożoności środowisk IT – im więcej systemów i integracji, tym większe ryzyko niespójności.

Podsumowanie: Dług architektoniczny to *odchylenie architektury IT od stanu pożądanego*, skutkujące obniżeniem sprawności i elastyczności całego biznesu. Spłata takiego długu często wymaga **dużych programów transformacyjnych** (np. konsolidacja systemów, re-platforming do chmury, redesign procesów). Zarządzanie długiem architektonicznym musi więc być elementem **strategii IT** i być wspierane przez najwyższe kierownictwo, gdyż ma charakter przekrojowy.

2.3 Rodzaje długu – komponenty techniczne i procesowe

Zarówno dług technologiczny, jak i architektoniczny, mogą przejawiać się w różnych **obszarach (komponentach)**:

- **Dług kodowy (aplikacyjny):** związany z jakością oprogramowania – np. niespójny styl kodu, brak testów, duplikacja kodu, użycie przestarzałych bibliotek. Przejawia się on w *systemach legacy* (stary kod), ale także w nowych rozwiązaniach, jeśli rozwijane są bez dbałości o jakość (np. szybkie prototypy AI/ML bez optymalizacji). Wskaźniki długu kodowego to m.in. wysoka złożoność cyklomatyczna, niski poziom pokrycia testami, wiele *code smells*.
- **Dług infrastrukturalny:** dotyczy infrastruktury IT – sprzętu, oprogramowania systemowego, platform. Objawia się np. posiadaniem serwerów czy baz danych w wersjach nieobjętych wsparciem producenta (End-of-Life), zaległościami w instalowaniu patchy bezpieczeństwa, ręcznym zarządzaniem konfiguracją zamiast automatyzacji (IaC). W środowisku chmurowym może to być brak aktualizacji usług, nieoptymalna architektura wykorzystania chmury (np. utrzymywanie drogich zasobów przez archaiczne wymagania). Dług infrastrukturalny jest typowy dla organizacji, które nie inwestowały regularnie w odnowę technologii – np. data center z serwerami 10-letnimi, stacje robocze z niewspieranym systemem operacyjnym, itp.
- **Dług procesowy:** dotyczy procesów i sposobu pracy – pojawia się, gdy organizacja nie modernizuje swoich procesów razem z technologią. Przykłady: brak automatyzacji powtarzalnych czynności (które mogłyby być wsparte przez RPA lub workflow, a są wykonywane ręcznie), nieefektywna procedura wdrażania zmian (np. brak CI/CD – wdrożenia odbywają się rzadko i manualnie, co spowalnia dostarczanie wartości). Dług procesowy łączy się z architektonicznym – np. konieczność ręcznego przenoszenia danych między dwoma systemami, które nie są zintegrowane, to skutek długu architektury i jednocześnie dodatkowa praca (dług procesowy).

Poniższa tabela podsumowuje różne typy systemów/rozwiązań w firmie ubezpieczeniowej i typowe przejawy długu w każdym z nich:

Typ systemu / rozwiązania	Charakterystyka długu technologicznego/architektonicznego
Systemy legacy (starsze)	Stare technologie (np. mainframe, COBOL), brak wsparcia producenta, monolity utrudniające rozwój. Wysoki dług kodowy (trudny w utrzymaniu kod), dług infrastrukturalny (przestarzałe platformy) i architektoniczny (niepasujące do nowej architektury).
EUC – End-User Computing (narzędzia użytkowników, np. arkusze Excel, bazy Access)	Rozwiązania tworzone poza kontrolą IT. Rozproszone dane w Excelach, powielanie logiki biznesowej w plikach. Generuje dług procesowy (obejście oficjalnych procesów) oraz dług architektoniczny (tzw. shadow IT – równoległa „architektura cienia”). Ryzyko błędów i braku zgodności z regulacjami (dane poza centralnymi systemami).
AI/ML (sztuczna inteligencja)	Modele uczenia maszynowego i algorytmy AI wprowadzone eksperymentalnie mogą stać się źródłem długu, jeśli brak ich dalszego utrzymania. Przykłady: modele wymagające ponownego trenowania (data drift) pozostawione bez opieki, brak dokumentacji modelu, kod prototypowy wokół AI niezaadaptowany do standardów. To dług technologiczny (skrypty, modele do poprawy) i potencjalnie dług architektoniczny (rozwiązania AI działające w silosie, poza głównym obiegiem danych).
Aplikacje mobilne	Szybko rozwijane aplikacje na urządzenia mobilne mogą generować dług, jeśli nie są utrzymywane. Np. starsze wersje aplikacji mobilnej nieprzystosowane do nowych wersji iOS/Android, zaległości w aktualizacji bibliotek SDK. Powoduje to dług technologiczny (kodu mobilnego) oraz ryzyka bezpieczeństwa (jeśli aplikacja nie korzysta z najnowszych zabezpieczeń systemu).

Typ systemu / rozwiązania

Charakterystyka długu technologicznego/architektonicznego

RPA (Robotic Process Automation)

Boty RPA tworzone do automatyzacji manualnych czynności potrafią szybko stać się „długiem”, gdy zmieniają się systemy, z którymi integrują. RPA często działa na poziomie interfejsu użytkownika (UI) – drobna zmiana ekranu może unieruchomić robota. Nagromadzenie setek botów bez centralnego zarządzania to **dług procesowy** (krótkoterminowe łatki zamiast trwałych rozwiązań) oraz **dług technologiczny**, jeśli boty są trudne do modyfikacji. W skrajnych przypadkach RPA staje się nowym legacy.

Chmura (cloud)

Korzystanie z chmury niesie ryzyko długu, jeśli nie ma odpowiedniej strategii. Przykładowo, równoległe używanie wielu chmur (multi-cloud) bez spójnych standardów może utrudnić zarządzanie – tworząc **dług architektoniczny** (brak jednolitego modelu). Również niewykorzystane lub źle skonfigurowane usługi chmurowe (np. instancje obliczeniowe pozostawione bez optymalizacji kosztowej) można traktować jako dług. Dodatkowo, utrzymywanie części systemów on-premises z powodu ograniczeń starych aplikacji tworzy dług infrastrukturalny – firma nie może w pełni skorzystać z benefitów chmury przez zaległości technologiczne.

Powyższe kategorie często na siebie nachodzą. Np. system legacy może jednocześnie powodować dług technologiczny (zły kod), architektoniczny (nie pasuje do nowej architektury) i procesowy (wymaga wielu manualnych obejść). Dlatego w praktyce **zarządzanie długiem wymaga holistycznego spojrzenia**. W kolejnej sekcji opisano, jak można ten dług mierzyć i oceniać, aby nadawać priorytety jego spłacie.

3. Metody pomiaru długu technologicznego i architektonicznego

Aby skutecznie zarządzać długiem technologicznym, trzeba go **zmierzyć lub oszacować** – inaczej trudno komunikować jego skalę i wpływ. Pomiar długu nie jest trywialny, ponieważ obejmuje zarówno twarde metryki techniczne, jak i miękkie aspekty ryzyka czy strategii. Duża firma ubezpieczeniowa powinna zastosować *wieloaspektowe podejście* do oceny długu, łącząc metody ilościowe (np. estymacje kosztów, wskaźniki jakości) z oceną ekspercką i analizą ryzyka. Poniżej przedstawiono rekomendowane metody pomiaru:

3.1 Estymacja kosztu modernizacji i utrzymania

Jedną z praktycznych metod wyceny długu jest podejście **finansowe** – oszacowanie, ile kosztowałoby spłacenie długu (czyli modernizacja) w porównaniu do kosztów jego dalszego utrzymywania. W tej metodzie identyfikujemy kluczowe obszary zadłużenia (np. przestarzały system polisowy) i estymujemy:

- *Koszt modernizacji/naprawy*: np. koszt projektu migracji systemu legacy do nowej platformy, przepisania modułu od zera, aktualizacji środowisk lub integracji rozproszonych narzędzi w jedno rozwiązanie. Wycena może bazować na liczbie roboczogodzin, potrzebnych licencjach, usługach zewnętrznych itp.
- *Koszt utrzymania status quo*: czyli jaką cenę płacimy zostając z długiem. Może on obejmować wyższe koszty utrzymania (np. specjalistów do starych technologii, częstsze awarie), ewentualne kary finansowe (np. za brak zgodności z regulacjami, jeżeli dług powoduje braki w bezpieczeństwie), a także *koszt alternatywny* – utracone korzyści z powodu niższej innowacyjności.

Różnica lub **relacja między tymi kosztami** daje miarę długu. Jeśli koszt modernizacji jest znacznie niższy niż przewidywane koszty dalszego trwania z długiem, oznacza to, że dług jest poważny i opłaca się go spłacić. W praktyce używa się wskaźników takich jak *Technical Debt Ratio (TDR)*, który definiuje

się jako stosunek kosztu naprawy do kosztu wytworzenia danego rozwiązania, wyrażony procentowo ([8 Top Metrics for Measuring Your Technical Debt in 2022](#)). Wzór:

TDR = (koszt refaktoryzacji/naprawy długu ÷ koszt zbudowania funkcjonalności) × 100%.

Narzędzia do analizy jakości kodu (np. SonarQube) potrafią automatycznie oszacować *techniczny koszt napraw* (np. w osobodniach) na podstawie wykrytych problemów i odnieść go do przybliżonego kosztu napisania kodu – otrzymując TDR. **Wartości TDR** rzędu kilku procent są uznawane za zdrowe (np. 5%), natomiast kilkadziesiąt procent to sygnał alarmowy, że system jest mocno zadłużony ([8 Top Metrics for Measuring Your Technical Debt in 2022](#)).

Podobne podejście można zastosować na poziomie architektury: oszacować np. że integracja rozproszonych systemów w jedną platformę kosztowałaby X mln zł, ale brak tej integracji oznacza Y mln zł rocznie kosztów operacyjnych więcej (praca ludzi, utrzymanie wielu systemów). Taka *analiza ROI spłaty długu* pozwala liczbowo przedstawić problem.

Warto podkreślić, że ta metoda pomiaru, choć bazuje na liczbach, zawiera pewne przybliżenia. Dlatego dobrze jest połączyć ją z kolejnymi metodami – analizą ryzyka i oceną jakości – aby otrzymać pełny obraz.

3.2 Analiza ryzyka operacyjnego i strategicznego

Drugą istotną metodą jest spojrzenie na dług przez pryzmat **ryzyka**. Każdy element długu technologicznego zwiększa jakieś ryzyko dla organizacji – np. ryzyko awarii, ryzyko niezgodności z regulacjami czy ryzyko utraty konkurencyjności. Można zatem mierzyć dług oceniając *prawdopodobieństwo i wpływ negatywnych zdarzeń*, do których dług może doprowadzić. Kroki takiej analizy:

- **Identyfikacja ryzyk związanych z danym długiem:** Np. przestarzały system może stwarzać ryzyko częstej niedostępności (ryzyko operacyjne) oraz ryzyko cyberataku (brak poprawek bezpieczeństwa). Arkusz Excel używany do istotnych obliczeń – ryzyko błędu ludzkiego i niezgodności z zasadami raportowania (ryzyko operacyjne i zgodności). Brak spójnego widoku klienta – ryzyko strategiczne utraty możliwości cross-sellingu.
- **Ocena ryzyka (metryka):** Dla każdego ryzyka określamy jego poziom, np. w skali punktowej 1-5 dla prawdopodobieństwa i wpływu. Mnożymy te wartości, uzyskując *ważoną ocenę ryzyka*. Można też ocenić *wartość ryzyka* w sensie finansowym (np. potencjalna strata przy awarii – ile może kosztować przestój).
- **Mapowanie na dług:** Następnie ryzyka agregujemy w kontekście poszczególnych systemów lub obszarów IT. Jeśli np. system A posiada 3 istotne ryzyka wynikające z jego stanu, sumujemy je – uzyskując **ryzyko skumulowane związane z długiem systemu A**. To pozwala rankować systemy: które generują największe ryzyko dla firmy.

Taka analiza ryzyka jest jakościowa, ale przekłada dług na **język zrozumiały dla biznesu i działów ryzyka**. Przykładowo, można stwierdzić: „System X niesie wysokie ryzyko (np. 20 punktów w macierzy ryzyka), co plasuje go w czerwonej strefie na mapie ryzyka operacyjnego”. Dla zarządu to sygnał, że system X to *poważne zagrożenie ciągłości działania* – a powodem jest nagromadzony dług technologiczny (stare technologie, łatki zamiast trwałych poprawek itd.).

W analizie ryzyka warto wziąć pod uwagę zarówno **ryzyka operacyjne (bieżące)**, jak i **ryzyka strategiczne (długofalowe)**. Operacyjne to np. awarie, incydenty bezpieczeństwa, błędy w procesach. Strategiczne – utrata przewagi konkurencyjnej, brak zdolności dostarczenia nowych produktów na rynek na czas (time-to-market). Dług architektoniczny będzie mocniej wpływał na te strategiczne aspekty (np. czy możemy szybko wdrożyć nowy cyfrowy kanał sprzedaży, czy architektura na to pozwala?), zaś technologiczny – na bieżącą stabilność i koszty.

Skala długu w kategoriach ryzyka może być prezentowana np. poprzez sumaryczny **wskaźnik ryzyka IT** dla organizacji lub wskaźniki cząstkowe. Ta metoda nie daje jednej liczby długu, ale profil ryzyka, co jest cennym uzupełnieniem twardych metryk.

3.3 Wskaźniki jakości architektonicznej i kodowej

Kolejnym podejściem jest ocena długu poprzez **mierniki jakości** – zarówno jakości kodu i systemów, jak i jakości architektury. Współczesne narzędzia i dobre praktyki dostarczają wielu KPI w tym zakresie:

- **Metryki kodu (wewnętrzna jakość oprogramowania):** Zespół deweloperski może mierzyć np. złożoność cyklomatyczną funkcji, stopień pokrycia kodu testami jednostkowymi (%), gęstość błędów (liczba błędów na 1000 linii kodu), duplikację kodu (% kodu powielonego), liczba ostrzeżeń statycznego analizatora kodu (code smells, vulnerabilities). Wysokie wartości tych metryk wskazują na dług technologiczny w obszarze kodu. Przykład: **cyklomatyczna złożoność** powyżej ustalonego progu sugeruje konieczność refaktoryzacji ([8 Top Metrics for Measuring Your Technical Debt in 2022](#)); niski udział pokrycia testami (np. poniżej 50%) – dług w obszarze testowania.
- **Metryki architektoniczne:** Tu ocena jest trudniejsza ilościowo, ale można śledzić pewne wskaźniki: np. *liczba punktowych integracji* (ile połączeń system-system poza zalecaną bus?), *stopień zgodności z architekturą referencyjną* (jak wiele odstępstw stwierdzono w audycie architektury), *redundancja funkcjonalności* (czy istnieją duplikaty modułów pełniących podobną rolę). Można także mierzyć **czas realizacji procesów end-to-end**, który często wydłuża się w wyniku długu architektonicznego (np. dane w kilku systemach – więcej kroków).
- **Ocena dojrzałości procesów IT:** Przykładowo model CMMI lub COBIT pozwala ocenić, na jakim poziomie dojrzałości są procesy w IT (zarządzanie zmianą, konfiguracją, wydaniem). Niski poziom (np. brak automatyzacji deploy'ów) to *dług procesowy* – dający się zmierzyć poprzez wskaźniki typu *częstość wydań* (release frequency), *czas od zgłoszenia zmiany do wdrożenia* itp. Im rzadziej i dłużej wdrażamy – tym większy prawdopodobnie dług w procesie wytwarzania oprogramowania.

Ocena jakości może być realizowana narzędziowo. **Statyczna analiza kodu** (np. SonarQube, Checkmarx) generuje raporty, które często zawierają również *wartość długu w dniach roboczych* potrzebnych do osiągnięcia pewnych celów jakości. **Przeglądy architektury** (np. architektoniczne code review dla nowych projektów vs standardy) mogą wykryć odstępstwa, które potem są rejestrowane jako *elementy długu architektonicznego*.

Dobłą praktyką jest ustanowienie **dashboardu metryk jakości** – tablicy, gdzie zespół na bieżąco śledzi kluczowe parametry (np. każda aplikacja ma przypisany „health index” wynikający z błędów i pokrycia testami, a cała architektura – indeks zgodności z target architecture). Pozwala to mierzyć postępy w spłacaniu długu (wskaźniki powinny się poprawiać) lub alarmować, gdy wskaźniki się pogarszają.

3.4 KPI i KRI do monitorowania długu technologicznego

Na podstawie powyższych metod, organizacja powinna zdefiniować **KPI (Key Performance Indicators)** oraz **KRI (Key Risk Indicators)** związane z długiem technologicznym, aby monitorować go w czasie. KPI będą pokazywały *jak efektywnie zarządzamy długiem*, a KRI – *jakie ryzyko z niego wynika*. Kilka przykładowych wskaźników:

- **Technical Debt Ratio (TDR):** wspomniany wyżej wskaźnik (procentowy) określający stosunek kosztu spłaty długu do kosztu wytworzenia. Np. TDR dla systemu A = 15% oznacza, że spłata zaległości technicznych równa się 15% wysiłku poniesionego na stworzenie tego systemu ([8 Top Metrics for Measuring Your Technical Debt in 2022](#)). **KPI:** celem może być utrzymanie TDR poniżej np. 5-10% ([8 Top Metrics for Measuring Your Technical Debt in 2022](#)) dla kluczowych systemów.

- **Udział budżetu IT przeznaczony na redukcję długu:** np. „% budżetu zmiany technologicznej poświęcony na projekty modernizacyjne”. McKinsey wskazuje, że dziś ok. 10-20% budżetów na nowe inicjatywy idzie na walkę z długiem ([Tech debt: Reclaiming tech equity | McKinsey](#)), można więc mierzyć ten udział i dążyć do optymalnego poziomu (nie za wysokiego, bo to hamuje nowe inicjatywy, ani za niskiego – bo dług rośnie).
- **Liczba krytycznych systemów na niewspieranych platformach:** np. liczba systemów produkcyjnych działających na EOL (End of Life) wersjach systemu operacyjnego, bazy danych czy frameworka. To **KRI** – rosnąca liczba takich systemów zwiększa ryzyko poważnej awarii lub braku wsparcia dostawcy.
- **Liczba istotnych incydentów spowodowanych długiem:** można monitorować incydenty (awarie, błędy) i tagować te, których przyczyną źródłową był np. stary błąd w kodzie, brak aktualizacji lub przeciążenie archaicznej architektury. Np. „ilość przerw w dostępności spowodowanych awarią systemu legacy w kwartale”. To KRI obrazujący realny wpływ długu.
- **Średni czas wprowadzenia zmiany (change lead time):** miernik DevOps/Agile, który pokazuje ile czasu mija od pomysłu lub wymagań do wdrożenia w produkcji. Jeśli jest bardzo długi, może to wskazywać, że architektura i proces mają dług (np. ręczne testy, zawiła architektura – wszystko wydłuża prace). Poprawa tego KPI (skrócenie czasu) często wymaga spłaty długu procesowego i architektonicznego.
- **Stopień wykorzystania automatyzacji w procesach operacyjnych:** np. % zgłoszeń obsługiwanych automatycznie vs manualnie, % procesów zmapowanych i mierzalnych. Niski poziom automatyzacji = dług procesowy. To może być KPI transformacji cyfrowej.
- **KRI cyberbezpieczeństwa związane z długiem:** np. liczba znanych luk bezpieczeństwa (CVEs) nierozwiązanych od >30 dni, liczba systemów bez aktualnej łatki krytycznej. Badania pokazują, że 60% incydentów wycieku danych miało przyczynę w znanych lukach, które nie zostały załatane ([Risks of Unpatched Vulnerabilities | ProWriters Insurance](#)) – stąd takie wskaźniki są ważnym KRI.

Można stworzyć tabelę *KPI/KRI długu technologicznego* i ustalić dla nich wartości progowe (thresholds) wyznaczające akceptowalny poziom. Przykładowy fragment takiej tabeli:

Wskaźnik	Opis	Rodzaj Próg alarmowy	
Technical Debt Ratio (TDR)	Procentowy stosunek kosztu naprawy długu do kosztu wytworzenia (wyliczany dla systemów). Niższy = lepiej.	KPI	> 10% (wysoki)
Systemy na EOL	Liczba kluczowych systemów działających na niewspieranych wersjach oprogramowania (OS/DB).	KRI	> 0 (każdy taki system to alarm)
Niezałatane luki bezpieczeństwa	Liczba krytycznych podatności (CVSS 9-10) nieusuniętych powyżej 1 miesiąca.	KRI	> 5
Incydenty spowodowane błędem legacy	Ilość incydentów (P1/P2) w kwartale, których przyczyną był błąd w starym kodzie lub awaria starej infrastruktury.	KRI	> 2 na kwartał
Lead time zmiany	Średni czas (dni) dostarczenia zmiany od zgłoszenia do wdrożenia (metryka DevOps). Długi czas sugeruje zadłużenie procesu.	KPI	> 30 dni

Wskaźnik	Opis	Rodzaj	Próg alarmowy
Pokrycie testami jednostkowymi	Średni % pokrycia kodu testami dla głównych aplikacji. Niższy = potencjalny dług jakościowy.	KPI	< 60%

Tabela 1. Przykładowe wskaźniki KPI/KRI do monitorowania długu technologicznego.

Takie wskaźniki należy regularnie mierzyć (np. miesięcznie, kwartalnie) i raportować do właściwych interesariuszy (IT, ryzyko, zarząd). Pozwalają one na **wczesne wykrycie trendu** – czy dług rośnie, czy spada – i podjęcie działań zanim dojdzie do poważnych problemów. W kolejnej części przeanalizujemy, jakie ryzyka wiążą się z nadmiernym długiem technologicznym.

4. Analiza ryzyk związanych z długiem technologicznym

Dług technologiczny i architektoniczny przekłada się na szereg ryzyk dla firmy ubezpieczeniowej. Zrozumienie tych ryzyk jest kluczowe, aby uzyskać wsparcie organizacji (szczególnie biznesu i zarządu) dla działań naprawczych. Poniżej omówiono główne kategorie ryzyka:

- Ryzyko operacyjne:** To bezpośrednie zagrożenie dla codziennego działania firmy wynikające z zawodności przestarzałych lub złożonych systemów. Dług technologiczny zwiększa prawdopodobieństwo **awarii, przestoju i błędów operacyjnych**. Np. stary system polisowy może ulec awarii w szczycie sprzedaży, wstrzymując obsługę klientów. Również skomplikowane, zadłużone środowisko IT wymaga więcej „gaszenia pożarów” przez zespoły – incydenty powtarzają się, bo podłoże problemu (dług) nie zostało wyeliminowane. Według danych, duże organizacje dostrzegają, że *komplikacje spowodowane starymi, przestarzałymi systemami sprawiają, że integracja nowych produktów i funkcji bywa prohibicyjnie kosztowna, a ukryte wyzwania architektoniczne powodują przekroczenia budżetów projektów i opóźnienia* ([Tech debt: Reclaiming tech equity | McKinsey](#)). To opisuje sytuację, gdy projekty operacyjnie nie dowożą wartości, bo zmagają się z „balastem” starej technologii – co jest de facto materializacją ryzyka operacyjnego (projekty muszą przeznaczać czas na rozwiązywanie problemów technicznych zamiast na rozwój funkcji). Dodatkowo, ryzyko operacyjne obejmuje **błędy danych i procesów** – np. korzystanie z wielu arkuszy Excel może prowadzić do pomyłek w kalkulacjach rezerw lub niewłaściwych decyzji underwritingowych, co szkodzi jakości operacji.
- Ryzyko zgodności (compliance):** Dotyczy niezgodności z przepisami i regulacjami. Dług technologiczny może utrudniać spełnienie wymogów prawa. Przykładowo, nowe regulacje (jak IFRS17 w rachunkowości ubezpieczeniowej czy RODO) często wymagają zmian w systemach – firmy z dużym długiem mają problem, by na czas dostosować swoje systemy do nowych przepisów. Ponadto, ręczne procesy i narzędzia EUC mogą prowadzić do naruszeń zasad ładu informatycznego i ochrony danych (np. dane osobowe klientów przechowywane w nieszyfrowanych arkuszach). **Błędy spowodowane długiem** mogą przerodzić się w incydenty naruszenia compliance – np. opóźnione wypłaty świadczeń (narażenie na kary KNF), utrata danych (naruszenie RODO). Ciężki dług utrudnia również implementację standardów bezpieczeństwa wymaganych np. przez DORA czy KSC. *Procesy z dużym udziałem manualnej pracy są bardziej podatne na błędy. Błędy te – od irytujących po niebezpieczne – mogą utrudnić przestrzeganie standardów i regulacji oraz aktualizację procedur wraz ze zmianą wymagań* ([Tackling technical debt in the insurance industry](#)). Ten cytat obrazuje, że jeżeli firma polega na archaicznych, ręcznych procedurach, może mieć trudność by szybko reagować na nowe wymogi regulacyjne, co jest istotnym ryzykiem compliance.
- Ryzyko reputacyjne:** Wynika z potencjalnego **negatywnego wizerunku firmy** w oczach klientów, partnerów i rynku, gdy problemy techniczne przełożą się na odczuwalne dla interesariuszy zdarzenia. Np. długa niedostępność systemu obsługi szkód uniemożliwiająca zgłoszenia szkód przez klientów może trafić do mediów, poważnie szkodząc reputacji ubezpieczyciela jako niezawodnego. Podobnie incydent cyberbezpieczeństwa (np. wyciek danych klientów)

spowodowany lukami w starym oprogramowaniu – oprócz kar finansowych – przynosi ogromne straty niematerialne: utratę zaufania. W raporcie Hiscox 43% firm, które padły ofiarą poważnego cyberataku, utraciło część klientów ([Evolution of cyber threats over a decade | NordLayer Blog](#)), co pokazuje skalę wpływu reputacyjnego. Ryzyko reputacyjne jest trudniej mierzalne, ale realne: klienci oczekują od instytucji finansowej profesjonalizmu i nowoczesności, a wpadki techniczne (np. archaiczna strona internetowa niewspółpracująca z mobile, częste usterki) mogą ich zrażać. W dobie mediów społecznościowych każdy większy problem technologiczny szybko zyskuje rozgłos.

- **Ryzyko cyberbezpieczeństwa:** Jak wspomniano, przestarzałe technologie są często dziurawe jak sito. **Niezałatane podatności**, brak aktualnych mechanizmów zabezpieczeń, systemy niezgodne z aktualnymi standardami szyfrowania – to zaproszenie dla cyberprzestępców. Według badań, 60% incydentów naruszenia danych miało przyczynę w znanych lukach bezpieczeństwa, które nie zostały załatane ([Risks of Unpatched Vulnerabilities | ProWriters Insurance](#)). To jednoznacznie wskazuje, że zaniechania w utrzymaniu technologii (czyli dług) przekładają się na incydenty cyber. Dla ubezpieczyciela konsekwencje mogą być katastrofalne: wyciek danych osobowych klientów, atak ransomware szyfrujący dane – co wstrzyma operacje firmy, a za odszyfrowanie trzeba płacić okup. Ponadto, dyrektywa NIS2 i DORA nakładają obowiązek silnych zabezpieczeń – niezastosowanie się grozi sankcjami ([DORA, NIS2 - cyberbezpieczeństwo to kluczowy dziś aspekt biznesu](#)). Firma z dużym długiem technologiczno-infrastrukturalnym może nie być w stanie przejść wymaganego **testu warunków skrajnych ICT** czy audytu bezpieczeństwa. Warto zauważyć, że ryzyko cyber odnosi się nie tylko do starych systemów on-premise – także niespójna konfiguracja chmurowa (często efekt szybkiego „technicznego” długu w architekturze chmury) bywa źródłem podatności.
- **Wpływ na time-to-market i rozwój nowych usług (ryzyko strategiczne):** W sektorze ubezpieczeniowym coraz ważniejsza jest **szybkość wprowadzania innowacji** – nowe produkty, nowe kanały (np. direct, mobile), personalizowane oferty oparte na AI. Dług technologiczny to *hamulec innowacji*. Zespoły IT zamiast budować nowe funkcjonalności, spędzają czas na utrzymaniu starych systemów (tzw. „operational load”). McKinsey zauważa, że wiele czasu pracowników IT w zadłużonych firmach schodzi na zarządzanie złożonością zamiast na innowacje ([Tech debt: Reclaiming tech equity | McKinsey](#)). Dług architektoniczny z kolei oznacza, że architektura nie wspiera nowych inicjatyw – np. brak API uniemożliwia łatwe otwarcie usług ubezpieczyciela dla partnerów zewnętrznych (ekosystemu InsurTech). W efekcie, firma może *przegrywać konkurencję* z bardziej zwinnymi graczami, którzy mają nowoczesne systemy. Jak stwierdza jeden z ekspertów: *„organizacje, które będą lepiej radzić sobie ze spłatą długu, wymuszą poprawę u reszty – innowatorzy staną się bardziej rentowni, a pozostali ubezpieczyciele będą musieli ich gonić albo zostaną w tyle”* ([Tackling technical debt in the insurance industry](#)). Jest to klasyczne ryzyko strategiczne: jeżeli nie zainwestujemy w unowocześnienie technologii dziś, jutro możemy stracić udział w rynku. Dług może też blokować konkretne inicjatywy biznesowe – np. chęć wdrożenia nowego produktu ubezpieczeniowego może wymagać zmian w systemie, które są nieproporcjonalnie trudne z powodu długu (np. system prowizyjny nie da się łatwo dostosować do nowego typu polisy). W skrajnych wypadkach projekt wprowadzenia nowego produktu może zostać porzucony lub znacznie opóźniony, co oznacza *utratę przychodów*.

Podsumowanie analizy ryzyka: Dług technologiczny i architektoniczny nie jest abstrakcyjnym problemem IT – realnie przekłada się na **ryzyko finansowe i operacyjne** firmy. Może powodować straty bezpośrednie (koszty incydentów, kary), jak i pośrednie (utratę klientów, okazji rynkowych). Warto dokumentować te ryzyka np. w rejestrze ryzyka operacyjnego firmy oraz przypisywać im właścicieli i plany mitygacji. W ten sposób dług technologiczny zostaje „przetłumaczony” na język zarządzania ryzykiem, co ułatwia jego zrozumienie poza działem IT.

W kolejnej sekcji przedstawiony zostanie plan realizacji pomiaru i analizy długu w praktyce – czyli jak faktycznie podejść do przeglądu środowiska IT i wyciągnięcia powyższych wniosków w dużej organizacji.

5. Plan realizacji pomiaru długu – harmonogram i zasoby

Efektywne zmierzenie i zmapowanie długu technologicznego w całej firmie ubezpieczeniowej to poważne przedsięwzięcie. Niezbędne jest podejście projektowe – z jasno zdefiniowanymi etapami, zadaniami, przypisaniem odpowiedzialności, zaangażowaniem ekspertów oraz wykorzystaniem odpowiednich narzędzi. Poniżej przedstawiono **proponowany harmonogram działań** obejmujący analizę różnych obszarów IT (core systems, EUC, mobile, AI, cloud itd.):

Etap	Zakres / Obszar	Działania	Ramy czasowe
1. Inwentaryzacja IT (pełny spis zasobów)	Całe środowisko: systemy core, aplikacje satelitarne, bazy danych, infrastruktura, narzędzia EUC, rozwiązania RPA, komponenty AI, usługi chmurowe.	- Zgromadzenie aktualnej listy wszystkich systemów i narzędzi IT, wraz z kluczowymi atrybutami: wersje technologii, daty wdrożenia, odpowiedzialni właściciele, zależności między systemami.- Identyfikacja <i>miejsz występowania EUC</i> (np. kluczowe pliki Excel używane w procesach) – poprzez ankiety w działach biznesowych.- Identyfikacja robotów RPA działających w organizacji (lista procesów zrobotyzowanych).- Przegląd wykorzystania chmury: które systemy są w chmurze, a które on-prem, i w jakich technologiach.	1–2 mies. (w zależności od dojrzałości CMDB i dokumentacji; może równolegle z etapem 2)
2. Ocena jakości kodu i aplikacji (dług kodowy)	Kluczowe systemy aplikacyjne (np. system polisowy, system obsługi szkód, portal klienta, aplikacja mobilna, moduły AI).	- Wykorzystanie narzędzi statycznej analizy kodu dla wybranych aplikacji (SonarQube lub podobne): zebranie metryk jakości (błędy, kody zapachowe, pokrycie testami, złożoność, duplikacja, techniczny „czas długu”).- Przegląd manualny krytycznych fragmentów kodu przez architektów/deweloperów seniorów – jakościowy opis zadłużenia (np. „moduł kalkulacji składek – spaghetti code, brak dokumentacji”).- Ocena zaległości w aktualizacjach wersji bibliotek i platform dla każdej aplikacji (czy są na najnowszych frameworkach?).- W przypadku aplikacji mobilnych: audyt kodu pod kątem zgodności z najnowszymi wytycznymi platform (iOS/Android), ewentualne wskaźniki wydajności i stabilności z store.	1 mies. (analiza może być rozłożona na różne systemy sekwencyjnie)
3. Ocena architektury i	Architektura korporacyjna	- Przeprowadzenie przeglądu architektury : porównanie aktualnej	1–2 mies. (wymaga

Etap	Zakres / Obszar	Działania	Ramy czasowe
infrastruktury (dług architektoniczny i infrastrukturalny)	(systemy core i ich integracje), infrastruktura Data Center i chmurowa, bezpieczeństwo.	architektury z modelami docelowymi (jeśli istnieją) lub z dobrymi praktykami. Identyfikacja odstępstw (np. duplikaty funkcji, omijanie ESB, brak centralnej bazy danych klienta).- Mapowanie przepływów danych między systemami – wykrycie silosów oraz punktów podatnych (single points of failure).- Ocena stanu infrastruktury: inwentaryzacja wersji OS, baz, serwerów – zaznaczenie które są EOL lub blisko EOL. Audyt konfiguracji pod kątem zgodności z aktualnymi standardami bezpieczeństwa (np. szyfrowanie, backup).- Przegląd wykorzystania chmury: ocena, czy obecna architektura hybrydowa jest optymalna, czy są zaległości w migracji (np. systemy, które planowano przenieść do chmury, a wciąż działają lokalnie z powodu ograniczeń).- Identyfikacja długu architektonicznego: lista stwierdzonych luk (np. „brak integracji systemu X i Y powoduje konieczność ręcznego przenoszenia danych”). Skorzystanie z wiedzy architektów korporacyjnych i dokumentacji EA (Enterprise Architecture). - Analiza, które kroki w procesach są wykonywane poza systemami (np. w Excelu, e-mailowo, ręcznie). Zidentyfikowanie tych kroków i ocena ich konsekwencji (opóźnienia, błędy).- Przegląd zidentyfikowanych plików Excel/Access (EUC): ocena ich złożoności (liczba makr, formuł), częstotliwości użycia, tego czy zawierają „<i>logikę biznesową krytyczną</i>”. Ocena ryzyka związanego z każdym (np. plik używany do raportowania do KNF – wysoki risk).- Przegląd wdrożonych robotów RPA: które procesy obsługują, jak stabilne są (historia awarii botów), czy mają alternatywę w postaci integracji systemowej. Ocena, czy liczba botów nie wymknęła się spod kontroli („robot sprawł”).- Ocena dojrzałości procesów IT (Change Management, Release Management): czy obecne praktyki nie	warsztatów z architektami i właścicielami systemów)
4. Ocena procesów biznesowych i narzędzi użytkowników (dług procesowy, EUC)	Krytyczne procesy biznesowe (np. zawieranie umów, likwidacja szkód, raportowanie aktuarialne) i ich wspomaganie narzędziami poza głównymi systemami.		1 mies. (wymaga współpracy z kluczowymi użytkownikami biznesowymi, np. warsztaty procesowe)

Etap	Zakres / Obszar	Działania	Ramy czasowe
		spowalniają dostarczania (np. czy każdy release wymaga akceptacji wielu ręcznych kroków).- Wynikiem będzie lista obszarów procesowych do usprawnienia (np. „Proces X zawiera 5 miejsc korzystania z EUC – ryzyko błędu i opóźnień, potrzeba automatyzacji w systemie docelowym”). - Zebranie wyników cząstkowych z etapów 2–4 w jedną bazę „długu”. Każdemu systemowi/obszarowi przypisanie: wskaźniki techniczne (z etapu 2 i 3), główne ryzyka (z etapu 3 i 4), szacowany koszt modernizacji (z wiedzy ekspertów lub danych rynkowych).- Stworzenie priorytetyzacji: rankingu systemów/procesów według istotności problemu. Np. macierz priorytetów biorąca pod uwagę <i>wartość biznesową vs poziom długu/ryzyka</i>. Wykorzystanie KPI/KRI: np. wskazanie które elementy powodują przekroczenie progów KRI.- Oszacowanie (wysokopoziomowe) nakładów na rozwiązanie problemów: np. „system A – konieczna wymiana w ciągu 2 lat, koszt ok. Y mln zł”, „proces B – do automatyzacji w ramach projektu RPA za X zł”.- Ustalenie zależności między długami: np. czy modernizacja systemu X rozwiąże jednocześnie problem Exceli Y i Z (bo staną się zbędne).- Konsultacja tych ustaleń z interesariuszami biznesowymi – upewnienie się, że priorytety technologiczne są zbieżne z priorytetami biznesu (np. jeśli dział sprzedaży bardzo narzeka na pewien system, jego problemy powinny być wysoko na liście).	
5. Agregacja i analiza całościowa (mapa długu i ryzyk)	N/A (podsumowanie poprzednich etapów)	np. wskazanie które elementy powodują przekroczenie progów KRI.- Oszacowanie (wysokopoziomowe) nakładów na rozwiązanie problemów: np. „system A – konieczna wymiana w ciągu 2 lat, koszt ok. Y mln zł”, „proces B – do automatyzacji w ramach projektu RPA za X zł”.- Ustalenie zależności między długami: np. czy modernizacja systemu X rozwiąże jednocześnie problem Exceli Y i Z (bo staną się zbędne).- Konsultacja tych ustaleń z interesariuszami biznesowymi – upewnienie się, że priorytety technologiczne są zbieżne z priorytetami biznesu (np. jeśli dział sprzedaży bardzo narzeka na pewien system, jego problemy powinny być wysoko na liście).	1 mies. (analiza biurowa + spotkania decyzyjne)
6. Raportowanie i roadmapa działań (wyniki i rekomendacje)	N/A (opracowanie wyników)	- Przygotowanie raportu końcowego z przeglądu długu technologicznego: dokumentu zawierającego m.in. opis stanu obecnego (znalezione długi, metryki), mapę ryzyka IT, oraz rekomendacje działań naprawczych z priorytetami.- Stworzenie roadmapy wysokiego poziomu (plan wieloletni)	1 mies. (przygotowanie i uzgodnienia)

Etap	Zakres / Obszar	Działania	Ramy czasowe
		redukcji długu: lista inicjatyw/projektów do uruchomienia, z proponowanymi ramami czasowymi, np. „2024: modernizacja infrastruktury sieciowej, 2025: wymiana systemu finansowego, 2026: konsolidacja baz danych klientów” – oczywiście w porozumieniu ze strategią biznesową.- Przygotowanie materiałów dla różnych odbiorców (szczegóły w punkcie 6): prezentacja dla zarządu, skrócony raport dla działu ryzyka, lista zadań dla IT.- Uruchomienie mechanizmu ciągłego monitorowania: zalecenie, aby wyniki przeglądu zostały wpisane do rejestru ryzyk i backlogu IT, a wskaźniki KPI/KRI były śledzone regularnie.	

Tabela 2. Proponowany harmonogram przeglądu i pomiaru długu technologicznego w organizacji.

Powyższy plan zakłada całościowe jednorazowe przedsięwzięcie (np. projekt 6- miesięczny) w celu *fotografii stanu obecnego*. W praktyce można też rozłożyć to na części – np. najpierw ocenić najbardziej krytyczne systemy core, a następnie sukcesywnie kolejne obszary. Ważne jest jednak, by **nie pominąć żadnego istotnego „źródła długu”** – często najtrudniej uchwytne są rzeczy poza głównym nurtem IT (właśnie EUC, drobne aplikacje). Dlatego potrzebna jest ścisła współpraca IT-Biznes podczas inwentaryzacji.

Zasoby potrzebne: Taki projekt wymaga zaangażowania multidyscyplinarnego zespołu:

- *Eksperti techniczni (inżynierowie oprogramowania, architekti IT):* do analizy kodu, oceny architektury, identyfikacji problemów technicznych. Np. architekt korporacyjny, architekti domenowi (od bezpieczeństwa, od danych), senior developerzy znający legacy.
- *Analitycy biznesowi / procesowi:* do zmapowania procesów i wychwycenia miejsc, gdzie technologia nie domaga (EUC, RPA, ręczne obejścia).
- *Specjaliści od ryzyka / compliance:* aby ocenić implikacje regulacyjne i ryzyka operacyjne zidentyfikowanych problemów oraz zapewnić zgodność z metodologią risk management w firmie.
- *Project manager / koordynator:* ponieważ dużo wątków musi być zgranych, a informacje zebrane i opracowane – ktoś powinien koordynować całość prac.
- *Narzędzia analityczne:* Warto wykorzystać dostępne narzędzia: system CMDB (Configuration Management Database) lub repozytorium architektoniczne (jeśli firma takie posiada) do inwentaryzacji; narzędzia do scanowania kodu (statyczna analiza), narzędzia do monitoringu infrastruktury (raporty o wersjach, patch level); arkusze/ankiety do zbierania informacji o EUC od użytkowników. Być może pomocne będzie też dedykowane narzędzie do oceny długu technicznego – na rynku istnieją np. platformy analityczne pokazujące „mapy zależności i długu” (czasem stosowane w enterprise architecture).
- *Budżet na ewentualnych zewnętrznych doradców:* Często firmy korzystają z pomocy konsultantów zewnętrznych (np. firm doradczych IT) do przeprowadzenia niezależnego assessmentu. Międzynarodowe firmy doradcze dysponują benchmarkami i narzędziami, które

mogą przyspieszyć analizę (np. porównają wskaźniki firmy do średniej rynkowej, wskażą typowe luki). Włączenie zewnętrznych audytorów IT może też zwiększyć wiarygodność wyników w oczach zarządu (jako obiektywnego spojrzenia).

Współpraca z dostawcami zewnętrznymi: W dużej organizacji wiele systemów jest dostarczanych lub utrzymywanych przez zewnętrznych vendorów/kontraktorów. Należy ich włączyć w proces oceny długu:

- Poprosić dostawców o **informacje o roadmapach i cyklu życia** używanych produktów (np. czy używana wersja systemu będzie wspierana, czy planują nową wersję – co może być okazją do aktualizacji).
- Wymagać od dostawców **raportu o stanie**: np. dostawca utrzymujący system może dostarczyć listę znanych problemów technicznych, zaległych update'ów, sugestii usprawnień.
- Uzgodnić ewentualnie wspólne warsztaty architektoniczne: integratorzy systemów często mają wiedzę, gdzie są trudne fragmenty integracji.
- Przy analizie RPA, jeśli korzystamy z usług firmy wdrażającej roboty, warto ich zaangażować do oceny czy obecne rozwiązania RPA są zgodne z najlepszymi praktykami (np. czy zastosowano narzędzia monitorowania botów).
- Dostawcy chmury (np. AWS, Azure) mogą pomóc w **ocenie efektywności wykorzystania chmury** – często mają architektów, którzy potrafią wskazać nieoptymalne elementy (to też forma długu – płacimy za chmurę nie uzyskując pełnej wartości przez złą konfigurację).

Dokumentacja: W trakcie projektu ważne jest dokumentowanie wszystkich znalezisk – powstanie swoisty **rejestr długu technologicznego** (lista punktów z opisem problemu, oceną krytyczności, rekomendacją rozwiązania). Ten rejestr posłuży później do śledzenia realizacji działań korygujących.

Po przeprowadzeniu powyższych etapów, firma będzie miała **całościowy obraz swojego zadłużenia technologicznego i architektonicznego**. Jednak równie istotne, co sam audyt, jest to, co z nim zrobimy dalej. Kolejna sekcja dotyczy **komunikacji wyników** w organizacji oraz zaplanowania i wdrożenia działań naprawczych.

6. Plan komunikacji wyników i działań naprawczych

Zebranie danych o długu technologicznym to dopiero początek – kluczowe jest odpowiednie przekazanie tych informacji decydentom i interesariuszom oraz wdrożenie konkretnych kroków naprawczych. Strategia komunikacji powinna uwzględniać różne poziomy organizacji (zarząd, piony biznesowe, IT, ryzyko) oraz zapewnić zrozumienie priorytetów. Poniżej opisano zalecenia w tym zakresie:

6.1 Raportowanie wyników do odpowiednich interesariuszy

Różne grupy odbiorców wymagają innego sposobu przedstawienia wyników audytu długu:

- **Zarząd (C-level, Rada Dyrektorów):** Należy przygotować zwięzłe, wysokopoziomowe podsumowanie ukazujące *skalę problemu i konsekwencje biznesowe*. Najlepiej przemawiają **kilka kluczowych liczb** (np. „średni poziom długu to 25% wartości IT” lub „istnieje 5 krytycznych obszarów wymagających natychmiastowej interwencji”) oraz **mapa ryzyka** (np. wykres pokazujący, że pewne ryzyka IT są w czerwonym polu). Zarząd nie potrzebuje technicznych detali, tylko informacji: *jak dług wpływa na realizację strategii firmy* i jakie inwestycje są potrzebne, by sytuację poprawić. Warto przedstawić **plan działania wraz z korzyściami** (np. „inwestując X w modernizację, zmniejszymy ryzyko poważnej awarii o Y% i skrócimy time-to-market nowych produktów o Z tygodni”). Forma: prezentacja zarządcza, z wykresami trendów, macierzą ryzyka, rankingiem top problemów. Ważne jest też zaznaczenie zgodności z regulacjami: np. „ten plan adresuje wymogi DORA – zarząd spełni obowiązek nadzoru nad ryzykiem ICT”.

- **Dział zarządzania ryzykiem / zgodności:** Tutaj raport powinien zawierać **szczegółowe KRI** i mapę ryzyka operacyjnego. Zaleca się przygotowanie osobnego rozdziału raportu poświęconego ryzykom, który może zostać włączony do procesu ERM (Enterprise Risk Management) firmy. Przykładowo: jeśli zidentyfikowano ryzyko „awaria systemu X może spowodować 8-godzinny przestój sprzedaży (szacowana utrata X zł)”, powinno to trafić do rejestru ryzyka operacyjnego. Dział ryzyka będzie monitorował, czy to ryzyko się zmniejsza (czyli czy dług jest spłacany). Raport dla nich może być bardziej *tekstowy, opisowy*, z odniesieniem do kategorii ryzyk korporacyjnych. Warto przedstawić planowane działania jako *mechanizmy mitygacji ryzyka z terminami*.
- **Pion IT (zespoły deweloperskie, infrastruktura, architekci):** Tutaj należy zejść na poziom konkretnych technicznych. Dostarczyć **listę zadań** lub backlog techniczny wynikający z audytu: np. „System A – upgrade Oracle DB do wersji nowszej”, „Refaktoryzacja modułu Y”, „Migracja serwerów Z do chmury”. Te elementy powinny zostać wprowadzone do narzędzi zarządzania pracą (Jira, Azure DevOps itp.) jako elementy roadmapy IT. Wewnętrzny raport IT może mieć formę szczegółowej dokumentacji (załączniki z wynikami skanów kodu, lista wykrytych podatności, itp.). Ważne, aby **ustalić priorytety** – np. oznaczyć zadania krytyczne vs zwykłe. Zespoły IT muszą wiedzieć, na czym się skupić w pierwszej kolejności. Można tu użyć kategorii: „must-do” (dla bezpieczeństwa/regulacji), „should-do” (dla efektywności) itd. Ponadto architekci powinni otrzymać *dokument z rekomendacjami architektonicznymi* (np. „wdrożyć platformę API gateway, żeby wyeliminować dług integracyjny w obszarze komunikacji z agentami”).
- **Jednostki biznesowe:** Dla dyrektorów poszczególnych pionów (sprzedaż, ubezpieczenia korporacyjne, likwidacja szkód, finanse itp.) warto przygotować spersonalizowaną informację: *jak dług tech. wpływa na ich obszar i co planujemy z tym zrobić*. Np. dla dyrektora sprzedaży: „System ofertowy ma opóźnienia w generowaniu ofert przez przestarzały moduł – planujemy go wymienić w przyszłym roku, co skróci czas przygotowania oferty o X minut i umożliwi szybkie zmiany produktowe”. Taka komunikacja buduje **poparcie biznesu** dla inwestycji w IT. Może mieć formę krótkiego memo lub warsztatu z kluczowymi managerami, gdzie IT prezentuje „co naprawimy, aby wasza praca była łatwiejsza”. Warto używać języka korzyści biznesowych (np. lepsza obsługa klientów, szybsze wprowadzanie produktów, mniej ręcznej pracy dla waszych zespołów).

Poniższa tabela przedstawia podsumowanie – komu, co i jak raportować:

Grupa docelowa	Zakres raportowania	Forma i częstotliwość
Zarząd (CEO, CFO, COO, CIO)	Ogólny poziom zadłużenia technologicznego; główne ryzyka dla biznesu (operacyjne, strategiczne); potrzebne decyzje inwestycyjne i priorytety.	Kwartalny raport na posiedzeniu zarządu (slajdy z kluczowymi KPI/KRI, trendami). Szczególne omówienie w ramach <i>IT Strategy Update</i> . Wizualizacje: wykresy, mapy ryzyka, ranking top 5 „długów” do spłaty.
Komitet Ryzyka / Dział Ryzyka	Szczegółowe ryzyka wynikające z długu IT, ich ocena (prawdopodobieństwo/skutek); postęp działań ograniczających ryzyko. Zgodność z apetytami na ryzyko.	Raport ryzyka operacyjnego (miesięczny lub kwartalny) – sekcja dot. ryzyka technologicznego. Tabela KRI z wartościami vs threshold. Udział w naradach zespołu ryzyka; raport do KNF (jeśli wymagany, np. ORX).
IT (dział rozwoju i utrzymania)	Konkretne elementy długu do adresowania: backlog techniczny, listy działań (upgrade’y, refactoring, testy). Priorytety i harmonogram prac.	Bieżące narzędzia zarządzania zadaniami (Jira – rejestr długu jako epiki/user story). Cotygodniowe statusy w ramach PMO IT.

Grupa docelowa	Zakres raportowania	Forma i częstotliwość
		Dodatkowo wewn. wiki z pełnym raportem technicznym z audytu.
Piony biznesowe (np. sprzedaż, operacje, finanse)	Wpływ długu na ich procesy i wyniki (np. wskazanie miejsc, gdzie tech hamuje sprzedaż lub generuje dodatkową pracę). Planowane usprawnienia i korzyści dla biznesu.	Spotkania/warsztaty z kluczowymi dyrektorami po audycie (jednorazowo) + kwartalne aktualizacje statusu projektów naprawczych. Materiały: krótkie prezentacje/skróty z naciskiem na korzyści (np. „po modernizacji systemu X czas obsługi klienta skróci się o Y”).

Tabela 3. Strategia raportowania wyników audytu długu do różnych interesariuszy.

W komunikacji należy unikać nadmiernego *żargonu technicznego* (poza raportem dla IT). Chodzi o zbudowanie świadomości i poczucia pilności u decydentów. Pomocne bywa odwołanie się do analogii finansowej (stąd nazwa „dług”): np. przedstawienie, że firma de facto ma „ukryty dług” w IT w wysokości X mln zł, który rośnie z odsetkami – to często działa na wyobraźnię finansistów.

6.2 Wizualizacja i priorytetyzacja działań naprawczych

Po zaakceptowaniu przez kierownictwo istnienia i skali długu, konieczne jest nadanie **priorytetów działaniom naprawczym**. W praktyce prawdopodobnie nie da się od razu spłacić całości długu (ani nie byłoby to wskazane – pamiętajmy, pewien dług jest akceptowalny). Dlatego należy zdecydować, co zrobić najpierw, a co może poczekać. Do tego służą:

- **Macierze priorytetów:** Np. macierz 2x2 z osiami *Wpływ na biznes* (lub ryzyko) vs *koszt/łatwość naprawy*. Pozycje w ćwiartce wysokiego wpływu i relatywnie niskiego kosztu to „quick wins” – te działania należy podjąć natychmiast (np. drobna modernizacja, która znacząco poprawi bezpieczeństwo). Pozycje o wysokim wpływie i wysokim koszcie to *projekty strategiczne* – też priorytetowe, ale wymagające planu (np. wymiana core systemu – duże koszty, ale krytyczne znaczenie). Niski wpływ a wysoki koszt – prawdopodobnie *odroczyć/spłacać stopniowo*.
- **Roadmapa czasowa:** Należy rozplanować inicjatywy w czasie, biorąc pod uwagę zależności. Często pewne prace muszą poprzedzać inne – np. najpierw uporządkować infrastrukturę (migracja do chmury), co potem ułatwi refaktoryzację aplikacji. Można stworzyć wykres Gantta wysokopoziomowy na 2-3 lata z zaznaczonymi strumieniami prac.
- **Przypisanie odpowiedzialności:** Każde działanie naprawcze powinno mieć swojego „właściciela” – osobę lub jednostkę odpowiedzialną za doprowadzenie do końca. Np. „Dyrektor Infrastruktury odpowiada za eliminację wszystkich serwerów Windows 2012 do końca roku”, „Departament Rozwoju A odpowiada za przebudowę modułu polisowego”. To zapewnia, że konkretne osoby poczuwają się do realizacji zadań.
- **Budżet i zasoby:** Priorytety muszą znaleźć odzwierciedlenie w alokacji budżetu. Jeśli modernizacja systemu ma wysoki priorytet, trzeba ją uwzględnić w planie inwestycyjnym i ewentualnie zatrzymać mniej ważne projekty, by zwolnić środki. Często walka z długiem konkuruje z inicjatywami rozwojowymi o budżet – tutaj pomocne jest przedstawienie z punktu 4 (ryzyka), by uzasadnić wydatki defensywne.

Wizualizacja postępów: Dla ułatwienia śledzenia i motywacji zespołów warto wizualizować redukcję długu. Np. **dashboard długu** pokazywany co miesiąc: ile punktów długu spłacono (to może być zestaw KPI jak w tabeli 1 – obserwowanie ich poprawy). Można również stosować **metafory** – np. „technologiczny debt burndown chart” podobny do wykresu spalania zwinnego, gdzie zobrazujemy, jak planujemy zmniejszać „wagę” długu w kolejnych kwartałach.

Priorytetyzacja a ryzyko: Generalnie pierwszeństwo powinny mieć te działania, które **adresują największe ryzyka** (np. krytyczne luki bezpieczeństwa – do załatwienia ASAP, nawet kosztem zamrożenia innych prac). Drugi w kolejności priorytet to te, które przyniosą **największe korzyści biznesowe** (np. umożliwią szybkie wprowadzenie nowego produktu – tu dług architektoniczny jest spłacany, by otworzyć drogę biznesowi). Transparentna argumentacja priorytetów – komunikowana również biznesowi – ułatwi akceptację, że pewne mniej istotne życzenia biznesowe mogą zejść na dalszy plan dopóki nie rozwiążemy palących problemów technicznych.

6.3 Wdrożenie rekomendacji i monitorowanie postępów

Wdrożenie rekomendacji z audytu długu powinno przybrać formę oficjalnego planu (roadmapy IT) zatwierdzonego przez kierownictwo. Najlepiej, jeśli zostanie to włączone w istniejące procesy zarządzania projektami i zmianą:

- Utworzenie (lub aktualizacja) **programu transformacji IT** – np. nadanie formalnej rangi projektom redukcji długu. Duże inicjatywy mogą stać się osobnymi projektami z własnym kierownikiem (np. „Program Legacy Core Replacement 2024–2025”). Mniejsze zadania można grupować w strumienie (np. „inicjatywy DevOps usprawniające procesy wydawnicze”).
- **Zintegrowanie z portfolio projektów:** Komitety sterujące ds. projektów powinny włączyć projekty naprawcze do portfolio i monitorować je tak samo jak projekty biznesowe. Być może potrzebna jest *re-priorytetyzacja portfela* – czasem aby zrobić miejsce na projekty techniczne, trzeba opóźnić jakiś projekt biznesowy mniejszej wagi. Tutaj kluczowe jest poparcie zarządu – mając je, dział IT i ryzyka mogą argumentować za tą zmianą priorytetów.
- **Zmiany procesowe:** Wdrożenie rekomendacji to nie tylko jednorazowe projekty. Część zaleceń może dotyczyć zmiany sposobu pracy na stałe – np. wprowadzenie obowiązkowego code review, aktualizacji bibliotek co pół roku, ograniczenia tworzenia nowych EUC poprzez lepszą obsługę potrzeb użytkowników w oficjalnych systemach. Te zmiany należy ująć w *politykach i procedurach IT*. Np. zaktualizować **politykę architektoniczną** firmy, by zapobiegać ponownemu narastaniu długu (np. „każdy nowy system musi mieć plan cyklu życia, z przewidzianymi aktualizacjami wersji co najmniej raz na 2 lata”).
- **Szkolenia i kultura:** Częścią wdrożenia jest też praca nad kulturą organizacji. Trzeba uświadomić zespołom, że dług technologiczny to wspólna odpowiedzialność. Można przeprowadzić warsztaty dla developerów z pisania czystego kodu, dla użytkowników biznesowych – z alternatyw dla tworzenia własnych makr (zachęcanie, by zgłaszali wymagania do IT zamiast tworzyć „na boku”). Jak radzą eksperci, kluczowe jest *zbudowanie kultury zmiany* i zrozumienia kosztów zaniechań ([Tackling technical debt in the insurance industry](#)) – pracownicy powinni rozumieć, czemu eliminujemy pewne stare nawyki.

Monitorowanie postępów: Po uruchomieniu działań naprawczych, konieczne jest śledzenie ich skuteczności:

- **Regularne przeglądy stanu długu:** np. co kwartał zespół architektoniczny dokonuje przeglądu wskaźników KPI/KRI i ocenia, czy zmierzamy w dobrym kierunku. Jeśli nie, korygujemy plan (może trzeba przyspieszyć jakieś działania lub zwiększyć nakłady).
- **Raportowanie do zarządu i rady nadzorczej:** w ramach np. raportów kwartalnych IT można pokazywać trend: „od ostatniego kwartału obniżyliśmy średni TDR systemów z 15% do 12%, zredukowaliśmy liczbę podatnych systemów z 10 do 4” itp.
- **Mechanizm ciągłego zarządzania długiem:** Ideałem jest wbudowanie monitoringu długu w **proces wytwórczy i utrzymaniowy**. Np. ustalamy, że przed wdrożeniem każdej większej zmiany przeprowadzimy analizę wpływu na dług (czy zmiana spłaca dług, czy może dodaje nowy? – jeżeli dodaje, to czy świadomie?). Można w **Definition of Done** dla projektów wpisać, że nie

zwiększą istniejącego długu powyżej ustalonego poziomu. Takie podejście prewencyjne uchroni przed powtórką sytuacji za kilka lat.

Benchmarking i najlepsze praktyki: Warto co jakiś czas porównać się z rynkiem – np. uczestniczyć w forach branżowych, wymieniać doświadczenia z innymi ubezpieczycielami, korzystać z raportów firm doradczych co do poziomu dojrzałości IT. Być może okaże się, że nasz dług w jakimś obszarze jest i tak mniejszy niż u konkurencji – wtedy możemy świadomie nie inwestować tam w pierwszej kolejności, skupiając się na obszarach, gdzie odstajemy.

Elastyczność planu: Plan działań naprawczych nie może być sztywny – otoczenie się zmienia. Może wejść nowa regulacja, która wymusi przesunięcie akcentów (np. priorytetem stanie się bezpieczeństwo chmury pod kątem NIS2, więc więcej zasobów pójdzie na to). Ważne, by mieć ogólny kierunek (redukcja długu), ale taktycznie dostosowywać kolejność kroków.

Podsumowanie

Realizacja opisanej strategii pozwoli dużej firmie ubezpieczeniowej **przejsć od etapu świadomości długu technologicznego do etapu jego kontrolowanej redukcji**. Poprzez kompleksowy audyt i pomiar, firma zidentyfikuje *gdzie i jak duży* jest problem, dzięki analizie ryzyk zrozumie *dla czego* należy działać, a dzięki zaplanowanym inicjatywom i dobrej komunikacji *jak* ten problem rozwiązać. Ważne jest ciągłe zaangażowanie zarówno IT, jak i biznesu w ten proces – dług technologiczny to problem całej organizacji, a jego spłata przełoży się na **większą odporność, zgodność regulacyjną i zdolność do innowacji**. Jak obrazowo stwierdził były CIO jednego z gigantów technologicznych: „*przebudowując nasze podejście do długu technicznego, zmniejszyliśmy czas inżynierów marnowany na jego obsługę z 75% do 25%, co pozwoliło nam stać się tym, kim dzisiaj jesteśmy*” ([Tech debt: Reclaiming tech equity | McKinsey](#)). Taki efekt – uwolnienie potencjału IT do kreowania wartości – jest celem, do którego dąży niniejsza strategia. Dzięki niej firma ubezpieczeniowa może bezpieczniej i szybciej wprowadzać nowe usługi na rynek, jednocześnie minimalizując ryzyko negatywnych niespodzianek ze strony przestarzałej technologii.