Andrzej Sobczak[1]

Modele postępowania z systemami legacy w administracji publicznej

1. Wstęp

Intensywny rozwój informatyzacji polskiej administracji publicznej na szczeblu centralnym rozpoczął się w połowie lat 90. ubiegłego wieku[2]. To wówczas m.in. zintensyfikowano prace nad systemem POLTAX, który jest wykorzystywany do chwili obecnej przez służby skarbowe. Pojawiały się systemy wspierające działanie ZUS (w szczególności KSI ZUS), pomocy społecznej (na początku system Pomost, a następnie system Syriusz). Kolejne przyspieszenie w zakresie informatyzacji administracji nastąpiło w latach 2004–2006, kiedy w ramach pomocy przedakcesyjnej dostępne stały się środki unijne, tj. środki przeznaczone dla krajów starających się o członkostwo w Unii Europejskiej (w tym czasie powstało m.in. rozwiązanie e-Deklaracje oraz portal e-PUAP). Lata 2007–2013 to następny okres programowania, w ramach którego Polsce zostało przyznane kolejne wsparcie finansowe – jego część przeznaczono na realizację projektów informatycznych w sektorze publicznym[3].

Stosunkowo długi okres informatyzacji administracji wskazuje, że należy spodziewać się, iż istotna część rozwiązań informatycznych używanych obecnie w jednostkach publicznych może zostać zaliczona do systemów legacy. W języku polskim nie ma tłumaczenia zwrotu „system legacy", który w pełni oddawałby znaczenie oryginału. Najczęściej tłumaczy się go jako system odziedziczony lub system zastany. Przyjęcie takiej terminologii powoduje od razu skojarzenie, że jest to system mający za sobą długi okres eksploatacji. Jest to jednak pewne uproszczenie. W inżynierii oprogramowania za system legacy uznaje się system, który

---

[1] Szkoła Główna Handlowa, Zakład Zarządzania Informatyką, Instytut Informatyki i Gospodarki Cyfrowej.

[2] W. Michalski, *Rozwój informatyzacji sektora administracji publicznej w Polsce*, „Telekomunikacja i techniki informacyjne" 2007, nr 3–4, s. 60–67.

[3] Por.: *Informatyzacja państwa w latach 2004–2015*, Instytut Łączności – Państwowy Instytut Badawczy, Warszawa, grudzień 2016.

obarczony jest (bardzo) dużym długiem technologicznym (por. kolejny podrozdział), co powoduje, że jego używanie stanowi istotne ryzyko dla funkcjonowania operacyjnego organizacji lub jest ograniczeniem dla wprowadzania przez nią nowych usług. Jednocześnie taki system zwykle wspiera szereg istotnych procesów/obszarów działania tej organizacji, więc nie można go w prosty sposób (tj. bez podjęcia szeregu złożonych działań o charakterze organizacyjnym i informatycznym) zastąpić innym systemem/systemami[4].

Celem niniejszego artykułu jest zidentyfikowanie kluczowych wyznaczników systemów legacy oraz określenie możliwych modeli postępowania z takimi rozwiązaniami – ze szczególnym uwzględnieniem uwarunkowań występujących w administracji publicznej.

W związku z tak sformułowanymi celami artykułu przyjęto następującą jego strukturę. W punkcie drugim omówiono pojęcie długu technologicznego w kontekście systemów legacy. Punkt trzeci zawiera omówienie wyznaczników systemów legacy. W punkcie czwartym określono możliwe modele postępowania z systemami legacy – uwzględniające specyfikę i uwarunkowania działania administracji publicznej (np. konieczność przestrzegania ustawy o zamówieniach publicznych). Całość artykułu kończy się podsumowaniem, w którym określono kierunki dalszych badań.

W artykule wykorzystano wyniki prac badawczych prowadzonych w ramach przedsięwzięcia SPARTA (Strategie Przedsiębiorstw w świecie Automatyzacji i RoboTyzAcji biznesu), realizowanego od roku 2017 w Zakładzie Zarządzania Informatyką Instytutu Informatyki i Gospodarki Cyfrowej SGH. Jeden z wątków tego przedsięwzięcia dotyczył systemów legacy. W praktyce okazuje się bowiem, że takie systemy stanowią w wielu wypadkach poważne ograniczenie cyfrowej transformacji organizacji i zastosowania na szeroką skalę automatyzacji procesów biznesowych. W ramach tego przedsięwzięcia w okresie maj–czerwiec 2017 r. zorganizowano na SGH dwa seminaria, podczas których przedstawiciele kilkunastu organizacji z bardzo różnych branż (m.in. telekomunikacja, banki, ubezpieczyciele, motoryzacja, sektor zdrowia) omówili problemy związane z systemami legacy. Dodatkowo, w czerwcu 2017 r., razem z ogólnopolskim czasopismem branżowym „Computerworld", zrealizowano w polskich organizacjach poglądowe badania ankietowe dotyczące systemów legacy (wzięły w nich udział

[4] Por.: R. Khadka, B.V. Batlajery, A.M. Saeidi, S. Jansen, J. Hage, *How do professionals perceive legacy systems and software modernization?*, „Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)", Hyderabad, Indie, 31 maja – 7 czerwca 2014, s. 36–47.

jednostki z: administracji publicznej, sektora utilities, telekomunikacji, banków, firm ubezpieczeniowych, przemysłu, sektora zdrowia)[5].

2. Pojęcie długu technologicznego w kontekście systemów legacy

Dla szczegółowego omówienia zagadnień związanych z systemami legacy istotne jest jednoznaczne zdefiniowanie terminu „dług technologiczny"[6]. Pojęcie to może być rozumiane szeroko – tj. jako każde podejście zastosowane (świadomie lub nie) przy budowie systemu informatycznego, które będzie utrudniać jego przyszły rozwój/dalsze zmiany. Dług technologiczny powstaje zatem wówczas, gdy stosowane są złe praktyki kodowania/dokumentowania kodu (w szczególności oznacza to jakikolwiek brak dokumentacji lub jej szczątkową postać), ale także wówczas, kiedy następuje zaniechanie przechodzenia na najnowsze wersje narzędzi/bibliotek programistycznych oraz trwanie przy starych wersjach systemów operacyjnych, baz danych, serwerów aplikacyjnych itp.

Szczególnym rodzajem długu technologicznego jest dług związany z używaniem bardzo starannych urządzeń sprzętowych (np. serwerów, macierzy itp.). Według klasyka inżynierii oprogramowania – E. Yourdona – źródłem długu technologicznego może być także brak projektu systemu, w rozumieniu braku całościowej koncepcji jego rozwoju, powodujący ustawiczne przerabianie oprogramowania w miarę pojawiania się nowych wymagań.

Przyjęcie jako obowiązującej przedstawionej definicji długu technologicznego powoduje, że również stosunkowo nowe systemy informatyczne (tj. mające np. 4–5 lat – co w przypadku wdrożeń systemów w administracji nie jest okresem szczególnie długim), które wspierają kluczowe procesy biznesowe, mogą zostać już zaliczone do kategorii systemów legacy. Mogą być one bowiem obarczone bardzo dużym długiem technologicznym – np. nie będzie istnieć ich dokumentacja, sposób kodowania może być daleki od kanonu zasad dobrego programowania, oprogramowanie może wykorzystywać stary stos technologiczny, a na dodatek na skutek konieczności uwzględnienia chaotycznych zmian

[5] Ze względu na zbyt małą próbkę uczestników badania zgormadzonych wyników nie można uznać za reprezentatywne. Mogą one jednie pomóc nakreślić ramy problemów, jakie występują w przypadku systemów legacy.

[6] P. Kruchten, R. Nord, I. Ozkaya, *Technical Debt: From Metaphor to Theory and Practice*, „IEEE Software", November–December 2012, vol. 29, iss. 6, s. 8–12.

w prawie ma wbudowane bardzo dużo często już zbyteczne funkcjonalności. Jednocześnie bardzo często systemy te wspierają kluczowe procesy urzędu i nie mogą być w prosty sposób zastąpione innymi rozwiązaniami.

3. Wyznaczniki systemów legacy

Jak wynika z analizy przedstawionej w poprzednim rozdziale określenie systemu mianem legacy niekonieczne musi być związane z długim okresem jego eksploatacji. Na podstawie analizy literaturowej[7] oraz rozmów eksperckich przeprowadzonych podczas warsztatów organizowanych na SGH zidentyfikowano listę 14 kryteriów, których zastosowanie pozwala w kompleksowy sposób poddać systemy ocenie pod kątem ich zaklasyfikowania jako „legacy". Można przyjąć, że jeżeli w odniesieniu do analizowanego systemu co najmniej połowa kryteriów będzie spełniona – jest to system legacy.

1. System został stworzony kilka/kilkanaście lat temu (i od tego czasu nie przeprowadzono jego istotnej przebudowy).
   Jest to jedno z najczęściej stosowanych (i najprostszych do użycia) kryteriów oceny systemów pod kątem ich klasyfikacji jako legacy. Można z dużym prawdopodobieństwem przyjąć, że systemy mające 10 i więcej lat są zaliczane do systemów odziedziczonych. Tylko w bardzo szczególnych przypadkach (przy bardzo konsekwentnym podejściu do rozwoju i utrzymania systemu) wiek systemu nie wpływa na tę klasyfikację.
2. Dokumentacja systemu jest cząstkowa/nieaktualna, a wiedza o jego działaniu jest nieskodyfikowana w postaci materialnej.
   Ze względu na ograniczenia czasowe i presję na jak najtańsze tworzenie systemu ograniczono aktualizację jego dokumentacji lub całkowicie zarzucono aktualizację wprowadzanych do niego zmian. Większość wiedzy nt. zasad działania systemu przechowywana jest w „głowach pracowników" – bez utrwalenia w usystematyzowanej i ustrukturalizowanej postaci.

[7] Por.: B. Batlajery, *Revisiting legacy systems and legacy modernization from the industrial perspective*, University of Utrecht, August 2013; J. Bisbal, D. Lawless, B. Wu, J. Grimson, *Legacy Information Systems: Issues and Directions*, „IEEE Software", September–October 1999, vol. 16, iss. 5, s. 2–10; M. Long, *Conquering Large Legacy Software*, Leanpub 2014, https://leanpub.com/largelegacysoftware (data odczytu 27.06.2018).

3. System powiązany jest z całym szeregiem zewnętrznych aplikacji o charakterze pomocniczym, przy czym powiązania te realizowane są w bardzo różny sposób.
   Ze względu na trudności w rozwoju systemu legacy bardzo często określone funkcjonalności realizowane są w zewnętrznych aplikacjach pomocniczych. W celu zapewnienia wymiany danych pomiędzy nimi a systemem legacy stosowane są bardzo różne mechanizmy integracji (przy czym najczęściej nie za pomocą API, lecz z wykorzystaniem bezpośredniego dostępu do bazy danych, czy też za pośrednictwem wymiany plików).
4. Kierunki i podejście do rozwoju systemu w czasie jego eksploatacji ulegało istotnym zmianom.
   W ciągu okresu eksploatacji systemu jego zakres funkcjonalny ulegał istotnym zmianom. Efektem tego jest wbudowanie w system funkcjonalności, które obecnie są zbędne, ale ze względu na szereg ograniczeń (zarówno o charakterze technologicznym, jak i organizacyjnym) ich eliminacja jest niemożliwa.
5. Architektura systemu nie odpowiada aktualnie obowiązującym standardom.
   System powstał kilka–kilkanaście lat temu – zastosowany przy jego budowie paradygmat architektoniczny nie jest zgodny z uznanymi obecnie wzorcami projektowymi. Ze względu na bardzo szeroki zakres wymaganych zmian dopasowanie architektury systemu do obecnych zasad budowy rozwiązań informatycznych nie jest możliwe.
6. Producent systemu nie zapewnia jego dalszego wsparcia.
   Ze względu na szereg uwarunkowań (zarówno o charakterze ekonomicznym, jak i technologicznym) producent systemu lub infrastruktury zaprzestał dostarczania jego/jej dalszego wsparcia. Problem taki dotyczyć może zarówno warstwy programowej systemu, jak i warstwy sprzętowej.
7. System wykazuje liczne podatności w zakresie bezpieczeństwa.
   Wraz ze zwiększaniem się wieku systemu i zaprzestaniem wsparcia ze strony producenta (w szczególności odejścia od publikacji tzw. łat bezpieczeństwa) bardzo istotnie wzrasta ryzyko udanego ataku na system ze strony hakerów, ze względu na ewentualne luki bezpieczeństwa.
8. Wersje bibliotek/języków programowania, za pomocą których został on stworzony nie są już wspierane/rozwijane (lub system nie został przepisany na ich nowe wersje).
   System został stworzony przy wykorzystaniu biblioteki programistycznej lub w języku programowania, który obecnie nie jest już używany lub jest używany i rozwijany, ale z jakichś powodów nie dokonano aktualizacji systemu legacy do jego najnowszej wersji.

9. System budowano z wykorzystaniem wielu różnych styli/sposobów kodowania.
   Ze względu na fakt, że system był rozbudowany przez długi czas, bardzo często przez różne osoby – zastosowano w nim różne style/sposoby tworzenia kodu źródłowego (np. różne konwencje nazewnicze), przy czym były one używane w sposób chaotyczny, bez żadnej systematyzacji w tym zakresie.
10. Część linii kodu systemu przestała być już używana, ale nie są one usuwane, bo wiedza na temat ich znaczenia jest ograniczona.
    Ponieważ system modyfikowano w sposób chaotyczny i bez dokumentacji, a część zmian przestała być zasadna, może okazać się, że istotna część linii kodu źródłowego systemu nie jest już potrzebna. Niestety ze względu na ograniczoną wiedzę dotyczącą sposobów działania systemu (związaną z brakiem aktualnej dokumentacji) nie ma podstaw do bezpiecznego (z perspektywy zapewnienia ciągłości działania systemu) usunięcia tych linii.
11. System bardzo trudno poddaje się automatyzacji procesu wytwórczego.
    Technologie zastosowane do budowy systemu praktycznie uniemożliwiają zastosowanie narzędzi do automatyzacji określonych kroków procesu wytwórczego (np. automatyzacji testów).
12. Istnieją trudności z pozyskaniem kadry znającej zastosowaną do budowy systemu technologię/język programowania/framework programistyczny.
    System zbudowano w technologiach programistycznych, które obecnie nie są już popularne. Przekłada się to na trudności w pozyskaniu znających je specjalistów.
13. Czas wprowadzania zmian (TTM) w systemie jest niezwykle długi.
    Ze względu na brak dostępu do specjalistów znających technologie użyte do budowy systemu, utrudnienia w automatyzacji procesu wytwórczego i braki w dokumentacji czas wprowadzania zmian jest niezwykle długi (dużo dłuższy aniżeli w przypadku współczesnych systemów).
14. System nie jest dopasowany do aktualnych potrzeb urzędu.
    Aktualne funkcjonalności systemu są zgodne z przepisami prawa (bo urząd działa w ramach przepisów prawa i na ich podstawie), ale okazuje się, że pewne procesy mogłyby być realizowane szybciej, gdyby wprowadzono odpowiednie modyfikacje systemowe – jednakże ze względu na szereg ograniczeń (zarówno o charakterze technologicznym, organizacyjnym, jak i finansowym) jest to mocno ograniczone lub wręcz niemożliwe.

Jak wynika z przedstawionego zestawienia lista czynników, które mają wpływ na uznanie, że dany system jest klasyfikowany jako legacy jest szeroka. Należy mieć tego świadomość, dokonując przeglądu systemów używanych w ramach urzędu. Może bowiem okazać się, że system wdrożony zaledwie 3–4 lata temu,

ale niemający aktualnej dokumentacji, napisany w sposób chaotyczny od strony programistycznej, przy budowie którego zastosowano nieaktualne na ten moment jego eksploatacji technologie, musi być uznany za „legacy".

4. Możliwe modele postępowania z systemami legacy

W przypadku, gdy w urzędzie funkcjonuje jeden (lub nawet więcej) systemów legacy możliwe są różne modele postępowania[8]. Poniżej przedstawiono ich charakterystykę.

1. **Ignorowanie (niezauważanie) faktu, że występują systemy legacy.**
   Z doświadczeń własnych autora wynika, że tzw. pracownicy merytoryczni urzędów (zwłaszcza szczebla operacyjnego) nie są zainteresowani problemami związanymi z eksploatacją i rozwojem systemów legacy. Okazuje się wręcz, że są z tego oprogramowania zadowoleni – bo znają jego funkcjonalności i ograniczenia, potrafią ominąć drobne problemy eksploatacyjne, mają już wieloletnie przyzwyczajenia związane z jego używaniem. Zdecydowanie większe ryzyka związane z eksploatacją systemów legacy znajdują się po stronie osoby odpowiedzialnej za obszar IT w urzędzie. Ze względu jednak na fakt, że praca w administracji charakteryzuje się bardzo dużą kadencyjnością (czego świadomość mają również decydenci z obszaru IT) – często ignorują oni te ryzyka – licząc, że nie wystąpią one podczas okresu ich zatrudnienia.
2. **Świadome pozostawienie systemu legacy i zapewnienie zasobów do jego konserwacji.**
   W tym modelu postępowania przeprowadzana jest pogłębiona ocena techniczna systemu legacy. Na jej podstawie przygotowywana jest analiza ryzyka związana z jego dalszą eksploatacją – w szczególności identyfikowane są czynniki ryzyka, których prawdopodobieństwo materializacji jest średnie lub duże, a wpływ na funkcjonowanie urzędu istotny. Prace te są podstawą do realizacji szeregu działań mitygujących te ryzyka – np. odpowiednie wzmocnienie kadrowe urzędu (o pracowników znających technologie stosowane w systemie legacy).

[8] Por.: R.C. Seacord, S. Comella-Dorda, G. Lewis, P. Place, D. Plakosh, *Legacy System Modernization Strategies*, Carnegie Mellon University, The Software Engineering Institute, Technical Report, CMU/SEI-2001-TR-025, July 2001.

3. **Re-inżynieria systemu legacy.**
   W wielu przypadkach możliwe jest dokonanie tzw. refaktoryzacji kodu systemu legacy[9]. Celem tych prac nie jest uzyskanie nowych funkcji systemu, ale uporządkowanie już istniejącego kodu – dzięki czemu staje się on łatwiejszy w utrzymaniu oraz szybciej wprowadza się w nim zmiany. Refaktoryzacja jest obarczona stosunkowo niskim ryzykiem niepowodzenia, można ją realizować etapowo. Jednocześnie istnieją trzy istotne bariery zastosowania jej w urzędach. Istotnym problemem jest fakt, że jej koszty są stosunkowo wysokie, a nie widać bezpośredniej wartości biznesowej z realizowanych prac – często jest to bariera mentalna trudna do przełamania w sektorze publicznym. Po drugie do przeprowadzenia refaktoryzacji trzeba posiadać lub pozyskać odpowiednie kompetencje merytoryczne – a w obecnej sytuacji rynkowej może być to trudne (w szczególności w kontekście zarobków oferowanych w sektorze publicznym). Wreszcie największą barierą jest fakt, że urzędy nie mają praw autorskich do kodów źródłowych eksploatowanego u siebie oprogramowania. Oznacza to, że nie mogą samodzielnie wprowadzać w nich jakichkolwiek zmian.
4. **Dekompozycja systemu legacy na moduły i przeniesienie części z nich do nowego systemu (stary częściowo pozostaje).**
   Krokiem dalej (w stosunku do modelu 3) jest przeprojektowanie architektury systemu legacy i dokonanie zmian nie tylko na poziomie kodu źródłowego, ale wręcz w jego strukturze[10]. Podejście takie jest zdecydowanie bardziej kosztowne i wiąże się z większym ryzykiem, ale na pewno pozwala na osiągnięcie większych korzyści. Na pewno ułatwiony jest bowiem dalszy rozwój funkcjonalny – dotyczy to zwłaszcza tych części systemu legacy, które zostały przeniesione do nowego rozwiązania. Jednocześnie jest to podejście bezpieczniejsze od wymiany systemu na nowy. Trudności, które związane są z zastosowaniem tego modelu są analogiczne do tych, które występują w przypadku modelu 2.
5. **Przygotowanie dla systemu legacy interfejsów wymiany danych.**
   W niektórych przypadkach logika biznesowa, która jest zaimplementowana w systemie legacy jest niezwykle złożona, a koszty jej odtworzenia w nowym

---

[9] M.A. Laguna, Y. Crespo, *A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring*, „Science of Computer Programming" 2013, vol. 78, iss. 8, s. 1010–1034.

[10] F. Fleurey, E. Breton, B. Baudry, A. Nicolas, J.M. Jézéquel, *Model-Driven Engineering for Software Migration in a Large Industrial Context* [w:] G. Engels, B. Opdyke, D.C. Schmidt, F. Weil (red.), *Model Driven Engineering Languages and Systems. MODELS 2007*, seria: Lecture Notes in Computer Science, vol. 4735, Springer, Berlin – Heidelberg, s. 482–497.

rozwiązaniu są bardzo wysokie i długotrwałe. Dlatego pewnym wyjściem z tej sytuacji jest przygotowanie interfejsów pozwalających na dołączenie do starego systemu nowych rozwiązań. Wówczas nowe funkcjonalności są już implementowane w systemach zewnętrznych, a system legacy jest pozostawiany bez zmian. Podejście takie nie eliminuje wszystkich czynników ryzyka, ale pozwala na stosunkowo szybkie dostarczanie nowych funkcji (istotnych z perspektywy wsparcia procesów biznesowych urzędu) oraz zabezpiecza przed przypadkowym wygenerowaniem błędu w systemie legacy (związanym z niewłaściwą jego modyfikacją).

6. **Uruchomienie dużego przedsięwzięcia wymiany systemu legacy i zastąpienie go nowym system (najczęściej budowanym od podstaw).**
   Do tej pory realizacja programu transformacyjnego była często stosowaną metodą radzenia sobie z systemami legacy w sektorze publicznym. Działania takie związane są z bardzo dużymi kosztami, ale administracja publiczna korzystała w tym zakresie ze środków unijnych. Należy się jednak spodziewać, że ten model postępowania będzie stosowany przez urzędy coraz rzadziej. Wpływają na to dwa czynniki. Po pierwsze w najbliższych latach nastąpi zmniejszenie dostępności środków unijnych (co ma związek z zakończeniem perspektywy finansowej na lata 2014–2020). Po drugie okazuje się, że pomimo bardzo dużych budżetów projektowych występują istotne trudności w realizacji tych przedsięwzięć (np. podczas realizacji Programu ePodatki czy projektów P1 i P2 na potrzeby służby zdrowia).
7. **Zastosowanie rozwiązania RPA (*Robotic Process Automation*) i pozostawienie systemów legacy.**
   W przypadku organizacji mających problemy z systemami legacy coraz częściej stosowane są rozwiązania klasy RPA (*Robotic Process Automation* – Zrobotyzowana Automatyzacja Procesów). Systemy tej kategorii działają na poziomie interfejsu graficznego działających rozwiązań i emulują (za pomocą tzw. robotów programowych) zachowania użytkownika systemu. Wówczas operacje wymiany danych pomiędzy systemami legacy realizowane do tej pory przez człowieka zastępuje oprogramowanie RPA. Ten model postępowania z systemami legacy ma istotną zaletę – nie wymaga ingerencji w systemy legacy (w przypadku konieczności zapewnienia wymiany danych pomiędzy np. nowym i starym systemem). Dodatkowo zdecydowanie pozwala ograniczyć stopień zaangażowania zasobów ludzkich w proste prace związane z manualnym przepisywaniem danych między systemami. Wadą jest konieczność zakupu licencji na RPA oraz zbudowanie i utrzymywanie robotów programowych. Dodatkowo narzędzia te nie eliminują

ryzyk związanych z eksploatacją systemów legacy, a jedynie je „ukrywają" (w zakresie integracji między systemami). Ostatnim aspektem, który należy uwzględnić jest fakt, że narzędzia te są stosunkowo nowe i administracja publiczna nie miała okazji się jeszcze z nimi zapoznać w praktyce.

8. **Outsourcing systemów legacy.**
   Jednym z najczęściej stosowanych podejść do radzenia sobie z systemami legacy przez polską administrację publiczną jest przeniesienie ryzyka związanego ze zmianami i utrzymaniem systemu legacy na firmę zewnętrzną. Dzięki temu – przynajmniej pozornie – urząd ma zagwarantowaną obsługę sytuacji awaryjnych i dalszy rozwój systemów. Wiąże się to jednak często z dużymi kosztami ponoszonymi przez sektor publiczny. Dodatkowo w przypadku materializacji ryzyk związanych z obsługą systemów legacy finalną odpowiedzialność podnosi urząd. Wreszcie często występujący brak pogłębianej kontroli nad działaniami firmy outsourcingowej (poza formalnymi odbiorami) może prowadzić do narastania problemu (np. poprzez zwiększanie się długu technologicznego).

Jak widać z przedstawionego zestawienia administracja publiczna ma wiele możliwości postępowania z systemami legacy. Niestety, na podstawie obserwacji autora można stwierdzić, że najczęściej wybierane jest jedno z dwóch skrajnych podejść – tj. ignorowanie zagadnienia lub jego outsourcing. W dłuższej perspektywie jest to istotna bariera przy budowie efektywnej e-administracji publicznej.

5. Podsumowanie i kierunki dalszych badań

W wielu urzędach systemy legacy zaczynają stanowić istotną barierę w dostosowywaniu się do ciągle zmieniających się przepisów prawnych (koszty i czas potrzebny na modyfikację bardzo się zwiększają). Dodatkowo rosną oczekiwania ze strony obywateli odnośnie do jakości usług publicznych świadczonych przez urzędy. Aby im sprostać jednostki administracji publicznej powinny móc wspierać swoje procesy biznesowe efektownymi narzędziami informatycznymi – bardzo częstą barierą w tym zakresie są właśnie systemy legacy[11]. Pierwszym krokiem do poprawy sytuacji jest wyeliminowanie starego systemu i uruchomienie

[11] IDC, *Application Modernization: Expanding Business Capabilities and Reducing Tech Complexity*, IDC Technology Spotlight, 2014, http://www.oracle.com/us/corporate/analystreports/idc-application-modernization-2211249.pdf, s. 1–6 (data odczytu: 27.06.2018).

na jego miejsce nowego rozwiązania. W praktyce okazuje się, że w wielu organizacjach ciągle dużym problem jest zagadnienie wyłączania starych systemów. Oznacza to, że po pierwszej fazie (zakończonej sukcesem) włączania nowego systemu stary system i tak będzie musiał być eksploatowany (w praktyce wynika to z faktu, że nie wszystkie potrzebne funkcjonalności udało się przenieść ze starego do nowego rozwiązania). Jest to rozwiązanie bardzo niebezpieczne dla działalności operacyjnej urzędu.

Jeżeli chodzi o przyszłe uwarunkowania związane ze stanem systemów legacy w polskich urzędach można sformułować dwie hipotezy. Po pierwsze nowo tworzone dla administracji publicznej systemy będą (moralnie) starzeć się coraz szybciej, bo często będą budowane w podejściu zwinnym (należy to odróżnić od rygorystycznie przestrzeganego manifestu i zasad zwinności, z czym znaczna część firm informatycznych w Polsce ma istotny problem) i od razu w wielu wypadkach zaciągany będzie dług technologiczny.

Po drugie coraz częściej następować będzie odejście w administracji od dużych programów transformacyjnych IT, a ze względu na zakończenie w nadchodzących latach dostępu do środków unijnych nie będzie już budżetów na takie działania. Oznacza to, że nie będzie środków na kompleksowe zastąpienie starych rozwiązań nowymi systemami.

Autor planuje podjąć kompleksowe badania dotyczące systemów legacy w polskiej administracji publicznej. Dzięki ich realizacji chce określić faktyczną skalę problemów dotyczących tej klasy systemów, zweryfikować dwie nakreślone powyżej hipotezy badawcze oraz wypracować metodyczne podstawy postępowania z systemami legacy w urzędach.

Bibliografia

Batlajery B., *Revisiting legacy systems and legacy modernization from the industrial perspective*, University of Utrecht, August 2013.

Bisbal J., Lawless D., Wu B., Grimson J., *Legacy Information Systems: Issues and Directions*, „IEEE Software" September/October 1999, vol. 16, iss. 5, s. 2–10.

Fleurey F., Breton E., Baudry B., Nicolas A., Jézéquel J.M., *Model-Driven Engineering for Software Migration in a Large Industrial Context* [w:] G. Engels, B. Opdyke, D.C. Schmidt, F. Weil (red.), *Model Driven Engineering Languages and Systems. MODELS 2007*, seria: Lecture Notes in Computer Science, vol. 4735, Springer, Berlin – Heidelberg, s. 482–497.

Informatyzacja państwa w latach 2004–2015, Instytut Łączności – Państwowy Instytut Badawczy, Warszawa, grudzień 2016.

Khadka R., Batlajery B.V., Saeidi A.M., Jansen S., Hage J., *How do professionals perceive legacy systems and software modernization?*, „Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)", Hyderabad, Indie, 31 maja – 7 czerwca 2014, s. 36–47.

Kruchten P., Nord R., Ozkaya I., *Technical Debt: From Metaphor to Theory and Practice*, „IEEE Software", November–December 2012, vol. 29, iss. 6, s. 8–12.

Laguna M.A., Crespo Y., *A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring*, „Science of Computer Programming" 2013, vol. 78, iss. 8, s. 1010–1034.

Michalski W., *Rozwój informatyzacji sektora administracji publicznej w Polsce*, „Telekomunikacja i techniki informacyjne" 2007, nr 3–4, s. 60–67.

Seacord R.C., Comella-Dorda S., Lewis G., Place P., Plakosh D., *Legacy System Modernization Strategies*, Carnegie Mellon University, The Software Engineering Institute, Technical Report, CMU/SEI-2001-TR-025, July 2001.

Źródła sieciowe

IDC, *Application Modernization: Expanding Business Capabilities and Reducing Tech Complexity*, IDC Technology Spotlight, 2014, http://www.oracle.com/us/corporate/analystreports/idc-application-modernization-2211249.pdf, s. 1–6 (data odczytu: 27.06.2018).

Long M., *Conquering Large Legacy Software*, Leanpub 2014, https://leanpub.com/largelegacysoftware (data odczytu: 27.06.2018).

* * *

Models of Dealing with Legacy Systems in Public Administration

Abstract

The paper discusses problems related to the maintenance and development of legacy systems. The starting point for these considerations was to define the concept of technological debt and the legacy system. The key part of the article are models of dealing with legacy systems in public administration.

Keywords: public administration, legacy system, technological debt, code refactoring, IT transformation