

# Maja Leszczyńska

---

## Proces utrzymania zintegrowanych systemów informatycznych

---

Ekonomiczne Problemy Usług nr 101, 121-136

---

2012

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej [bazhum.muzhp.pl](http://bazhum.muzhp.pl), gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.

MAJA LESZCZYŃSKA

Uniwersytet Ekonomiczny we Wrocławiu

## PROCES UTRZYMANIA ZINTEGROWANYCH SYSTEMÓW INFORMATYCZNYCH

### Wprowadzenie

Zintegrowane systemy informatyczne (ZSI) zawierają zestaw standardowych funkcji pozwalających na kompleksowe wsparcie podstawowych procesów gospodarczych zachodzących w przedsiębiorstwach. Funkcjonują one w przestrzeni rynkowej w dwóch postaciach. Przede wszystkim jako produkty oferowane na rynku, a z drugiej strony – jako rozwiązania już wdrożone w konkretnych przedsiębiorstwach (zwane instalacjami systemu). Muszą ciągle nadążać za zmianami zachodzącymi w przestrzeni prawno-gospodarczej, technologicznej oraz za specyficznymi potrzebami aktualnych i potencjalnych odbiorców. Z tego punktu widzenia zasadne oraz ciekawe wydaje się więc podjęcie analizy procesu ich utrzymania przez pryzmat dorobku teoretycznego oraz praktycznego, co jest celem niniejszego artykułu.

### 1. Proces utrzymania w świetle dorobku literatury zagranicznej

Literatura obcojęzyczna wiąże proces utrzymania z pojęciem ewolucji. Ewolucja oprogramowania (ang. *software evolution*) określana jest jako proces zmian zachodzących w oprogramowaniu w czasie jego życia i jako taki nie-

unikniony i naturalny. Pogląd ten dominuje w pracach amerykańskiego naukowca M.M. Lehmana, który w wyniku swoich wieloletnich badań sformułował osiem praw dotyczących ewolucji systemów informatycznych<sup>1</sup>. Należy zauważyć, że prawa Lehmana powstały na bazie obserwacji kilku dużych systemów informatycznych (np. IBM OS/360-70), arbitralnie wybranych przez ich twórcę, prowadzonej od lat sześćdziesiątych do osiemdziesiątych ubiegłego wieku. Do ich stworzenia nie wykorzystano metod analizy statystycznej ani formalnych metod dowodzenia, stąd też w chwili powstania nie były w pełni akceptowane jako prawa. Z czasem wyniki prac M.M. Lehmana zostały ponownie zweryfikowane i potwierdzone w serii projektów FEAST (ang. *Feedback, Evolution and Software Technology*) i nazwa „prawa” została utrzymana, a twierdzenia M.M. Lehmana weszły na stałe do kanonu nauk związanych z inżynierią oprogramowania. Najważniejsze z nich mówią o tym, że<sup>2</sup>:

- a) program stosowany w rzeczywistym środowisku musi być stale do niego adaptowany, w przeciwnym przypadku będzie stopniowo coraz mniej użyteczny – prawo ciągłej zmiany (1974 rok);
- b) wraz z ewolucją oprogramowania jego struktura staje się coraz bardziej złożona, chyba że w wyniku dodatkowych prac (i nakładów) następuje jej uproszczenie – prawo wzrastającej złożoności (1974 rok);
- c) funkcjonalność oprogramowania musi rosnąć, aby satysfakcja odbiorców systemu pozostała na tym samym poziomie – prawo ciągłego wzrostu (1991 rok);
- d) procesy ewolucyjne oprogramowania są wielopoziomowe, mają naturę iteracyjną i wymagają informacji z wielu punktów widzenia. Wykorzystanie informacji zwrotnej pozwala osiągnąć istotną poprawę procesu ewolucji – prawo przyrostowego rozwoju (1996 rok).

W kontekście prac M.M. Lehmana utrzymanie (ang. *software maintenance*) zawiera się w ewolucji. Jest procesem zaplanowanym, mającym na celu kontrolę procesów ewolucyjnych i czynników, które je implikują. Proces utrzymania w literaturze obcojęzycznej, podobnie jak i w krajowej, umiejscawiany jest po fazach analizy, projektowania, tworzenia i wdrożenia, czyli w momencie, kiedy system przechodzi w fazę eksploatacji w danej organizacji.

---

<sup>1</sup> M.M. Lehman, *The Future of Software – Managing Evolution*, inv.contr., vol. 15, no. 1, IEEE Softw., Jan-Feb 1998.

<sup>2</sup> *Ibidem*.

Ponadto podobnie, jak w literaturze krajowej wielu autorów obcojęzycznych ogranicza się jedynie do umiejscowienia utrzymania w cyklu życia, koncentrując swoje rozważania na jego wcześniejszych fazach. Sytuacja ta ulega zdecydowanej zmianie w ostatnim dziesięcioleciu, kiedy to zaczynają się pojawiać pozycje koncentrujące swoje rozważania na utrzymaniu systemu informatycznego.

Opracowywane są też międzynarodowe standardy takie, jak IEEE Standard for Software Maintenance No. 1219 czy ISO/IEC 14764 (Software Engineering – Software Life Cycle Processes – Maintenance), które jednoznacznie definiują proces utrzymania jako „proces obejmujący: modyfikację oprogramowania po jego dostarczeniu i wdrożeniu w celu korygowania błędów, podniesienia efektywności lub innych atrybutów, adaptowania produktu do zmieniających się warunków otoczenia, w tym środowiska operacyjnego”<sup>3</sup>, czyli zarówno rozwój, jak i naprawę oprogramowania w zależności od potrzeb. Ponadto w pozycji *Software Maintenance Management, Evolution and Continuous Improvement*, mówi się już nie tylko o perspektywie użytkowników oprogramowania, lecz także o perspektywie jego twórców<sup>4</sup>.

Literatura obcojęzyczna obfituje w liczne definicje pojęcia utrzymania. Wśród przykładowych można wymienić:

- a) „utrzymanie obejmuje sprzęt, oprogramowanie, dokumentację oraz procedury mające na celu eliminację ewentualnych błędów systemu, odpowiedź na nowe wymagania oraz podniesienie efektywności wspomnianych procesów biznesowych”<sup>5</sup>;
- b) „utrzymanie to ogół działań wymaganych do wsparcia oprogramowania po najniższych kosztach; niektóre działania są inicjowane na etapie tworzenia oprogramowania, ale większość ma miejsce już po jego dostarczeniu do odbiorcy”<sup>6</sup>;

---

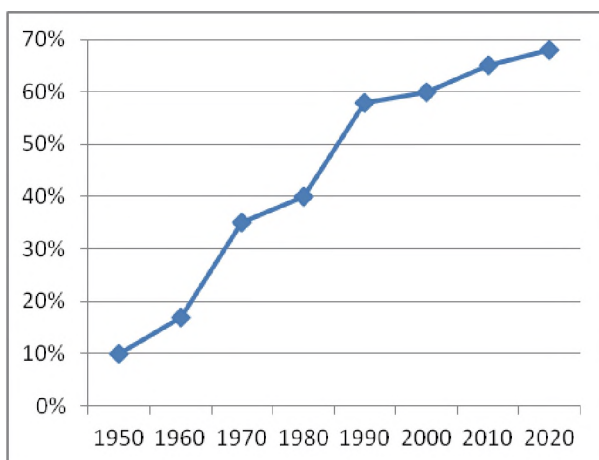
<sup>3</sup> Institute of Electrical and Electronics Engineers. *IEEE Standard for Software Maintenance, Standard 1219*, Institute of Electrical and Electronics Engineers, New York 1998.

<sup>4</sup> A. April, A. Abran, *Software Maintenance Management, Evolution and Continuous Improvement*, John Wiley & Sons, Hoboken 2008.

<sup>5</sup> K.C. Loudon, J.P. Loudon, *Management Information Systems. Managing The Digital Firm*, Prentice Hall, Upper Saddle River 2006.

<sup>6</sup> A. Abran, J.W. Moore, *Guide to the Software Engineering Body of Knowledge (SWE-BOK)*. 2005 Version. IEEE Computer Society: Los Alamos, California, 2005.

- c) „utrzymanie to proces wprowadzenia do systemu niezbędnych zmian wynikających głównie ze sprzężenia zwrotnego między otoczeniem a organizacją”<sup>7</sup>;
- d) „utrzymanie to modyfikacje kodu i powiązanej dokumentacji w odpowiedzi na problemy lub potrzeby rozwojowe, celem jest modyfikowanie istniejącego oprogramowania bez naruszania jego integralności”<sup>8</sup>;
- e) „utrzymanie to zmiany, które są wprowadzane w oprogramowaniu po jego uruchomieniu u odbiorcy”<sup>9</sup>.



Rys. 1. Udział utrzymania w stosunku do pozostałych działań związanych z oprogramowaniem

Źródło: J. Capers, *Estimating Software Costs*, McGraw-Hill 2007.

Autorzy anglojęzyczni podkreślają ponadto znaczenie utrzymania z uwagi na koszty tego procesu, które znacząco przekraczają koszty związane z pierwotnym stworzeniem systemu. W tym kontekście można rozpatrywać utrzymanie jako najdłuższą i w świetle badań najkosztowniejszą fazę cyklu życia syte-

<sup>7</sup> P. Beynon-Davies, *Information systems. An Introduction to Informatics in Organisations*. PALGRAVE. Houndmills–New York 2002.

<sup>8</sup> *International Standards Organization. Information Technology. Software Life Cycle Processes*, ISO/IEC Standard 12207. International Organization for Standardization: Geneva, Switzerland, 1995.

<sup>9</sup> J. Martin, C. McClure, *Software Maintenance. The Problem and Its Solutions*, Prentice Hall, Englewood Cliffs 1983.

mu. Ocenia się, że koszty związane z utrzymaniem to około 50–90% wszystkich kosztów ponoszonych na przestrzeni całego cyklu życia oprogramowania, a na każdego dolara wydanego na stworzenie systemu przypadają dwa wydane na jego utrzymanie<sup>10</sup>. Ponadto udział działań związanych z utrzymaniem oprogramowania w stosunku do wszystkich przedsięwzięć związanych z oprogramowaniem stale rośnie z około 10% w latach pięćdziesiątych, do 60% obecnie i około 70% w ciągu najbliższych 10 lat (patrz rysunek 1). W tym kontekście P. Beynon-Davies zauważa, że „działalność związana z utrzymaniem SI powinna być traktowana priorytetowo głównie ze względu na jej koszty, które znacząco przekraczają koszty tworzenia i wdrażania systemu”<sup>11</sup>. Wskazuje na konieczność traktowania jej jako procesu i podjęcia działań mających na celu zarządzanie nim. Wśród dróg realizacji tego postulatu wymienia:

- a) tworzenie komórek odpowiedzialnych za utrzymanie SI poprzez modyfikowanie, naprawianie i uaktualnianie zastosowanych rozwiązań;
- b) projektowanie systemu z myślą o jego przyszłym rozwoju, czyli stosowanie elastycznych rozwiązań, które ułatwią jego dostosowywanie do zachodzących zmian;
- c) zarządzanie konfiguracją mające na celu identyfikowanie koniecznych zmian SI, kontrolowanie poprawności ich wprowadzania i działania, informowanie o zmianach pozostałych użytkowników;
- d) aktualizowanie starzejących się rozwiązań poprzez planowanie i wprowadzanie zmian w zakresie zastosowanego sprzętu, oprogramowania oraz technologii teleinformatycznych.

Kontynuując rozważania P. Beynon-Davies, należy zauważyć, że ZSI pomimo tego, że są rozwiązaniami ustabilizowanymi pod względem programowym i funkcjonalnym na skutek nieustanych zmian otoczenia oraz samych podmiotów, w których zostały wdrożone, podlegają procesom ciągłych modyfikacji. W tym kontekście zmiana staje się naturalną częścią cyklu ich życia i nie da się jej uniknąć. Proces mający na celu kontrolę, planowanie i zorganizowanie działań podejmowanych w odpowiedzi na potrzeby wprowadzania zmian w systemie informatycznym (niezależnie od ich źródeł) jest w pracy definiowany jako proces utrzymania. O rozpoczęciu procesu utrzymania należy mówić

---

<sup>10</sup> A. April, A. Abran, *op.cit.*, B.W. Boehm, *Industrial Software Metrics Top 10 List*, IEEE Software 1987.

<sup>11</sup> P. Beynon-Davies, *op.cit.*

w kontekście określonej perspektywy. Z perspektywy przedsiębiorstwa, które użytkuje dany system, utrzymanie systemu informatycznego rozpoczyna się w momencie zakończenia procesu wdrożenia (na ogół określanego formalnie podpisaniem protokołu zakończenia wdrożenia) i rozpoczęcia jego eksploatacji. Z perspektywy dostawcy oprogramowania, wiodącej dla rozważań, o utrzymaniu systemu można mówić od momentu zakończenia jego pierwszego wdrożenia. Eksploatacja systemu przez użytkowników obfituje bowiem w liczne wydarzenia, w wyniku których odkrywane są wcześniej niewykryte błędy, pojawiają się nowe wymagania czy zmienia się środowisko funkcjonowania systemu.

## **2. Proces utrzymania w świetle dorobku literatury krajowej**

Z kolei w krajowej literaturze przedmiotu pojęcie utrzymania jest stosunkowo nowe. W odniesieniu do systemów informatycznych stosowane jest dopiero od około dziesięciu lat. W tym kontekście proces utrzymania jest identyfikowany jako jedna z faz cyklu życia systemu informatycznego. Dominująca dla prowadzonych rozważań jest perspektywa użytkowników systemów informatycznych, w związku z tym wskazuje się, że utrzymanie następuje po wdrożeniu i uruchomieniu systemu informatycznego w danej organizacji. Zaznaczyć należy, że autorzy krajowi w stosunku do działań charakterystyczny dla tej fazy cyklu życia systemu informatycznego nie zawsze stosują nazwę „utrzymanie”. W tym kontekście często mówi się o: doskonaleniu, doskonaleniu i rozwoju, adaptacji i modyfikacji, eksploatacji, eksploatacji i modyfikacji lub pielęgnacji i wygładzaniu. Niemniej jednak w literaturze krajowej jednoznacznie wskazuje się na powiązanie działań z tej fazy cyklu życia systemu informatycznego z jego deprecjacją. Deprecjacja rozumiana jest jako zmniejszenie lub utrata wartości systemu informatycznego w trakcie eksploatacji. Jako przyczyny deprecjacji wskazuje się między innymi takie procesy, jak: zużycie sprzętu komputerowego, aktualizacje systemów operacyjnych, zmiany w przepisach prawa, wzrost świadomości potrzeb własnych użytkowników, rozwój technologii informacyjnych. Ponadto literatura krajowa klasyfikuje deprecjację na materialną, ekonomiczną i funkcjonalną. Deprecjacja materialna jest związana z technicznym zużyciem sprzętu komputerowego, ekonomiczna – z moralnym zużyciem sprzętu i oprogramowania wynikającym z tempa rozwoju technologii informatycznych, z kolei deprecjacja funkcjonalna – ze zmianami potrzeb informacyjnych

użytkownika. Potrzeby te mogą ulegać zmianom na skutek przyczyn obiektywnych, które są niezależne od użytkownika (np. zmiany przepisów prawnych) lub wymuszone przez rynek i konkurencję (zmiana sposobu komunikacji z kooperantami, np. faktury czy zamówienia elektroniczne), oraz subiektywnych (np. uświadomienie przez użytkownika konieczności poszerzenia zakresu merytorycznego systemu czy też zwiększania stopnia informatycznego wspomagania procesów decyzyjnych). Odpowiedzią na deprecjację są działania o charakterze modernizacyjnym i rozwojowym. Działania modernizacyjne mają na celu wykrywanie wad i usterek systemu oraz ich usuwanie, tak aby standard systemu został na dotychczasowym poziomie bez wprowadzania zmian ilościowych czy jakościowych. Z kolei działania rozwojowe polegają na wprowadzeniu rozwiązań strukturalnych i funkcjonalnych niestosowanych w przeszłości. W ten sposób dochodzi do powstawania systemu o wyższych parametrach użytkowych.

Wymienione działania we współczesnym rozumieniu można jednoznacznie odnieść do procesu utrzymania, który w literaturze krajowej definiowany w następujący sposób:

- a) „utrzymanie polega na ciągłym monitorowaniu i ocenie poprawności działania systemu, na tym etapie ujawniają się nieusunięte wcześniej defekty, może być wymagane poszerzenie funkcjonalności lub cech użytkowych systemu, może też wystąpić potrzeba dopasowania systemu do zmieniającego się środowiska i realizowanych procesów”<sup>12</sup>;
- b) „utrzymanie to czynności mające na celu zachowanie lub podniesienie wydajności systemu, wprowadzenie zmian usprawniających pracę użytkowników czy włączenie w zakres objęty systemem nowych obszarów funkcjonalnych”<sup>13</sup>;
- c) „utrzymanie to celowe czynności serwisowe, modernizacyjne lub rozwojowe”<sup>14</sup>;
- d) „utrzymanie to ciąg działań umożliwiających prawidłowe użytkowanie i obsługę systemu”<sup>15</sup>.

---

<sup>12</sup> J. Kisielnicki, H. Sroka, *Metody projektowania i wdrażania systemów*, Placet, Warszawa 2001 i 2005.

<sup>13</sup> P. Lech, *Zintegrowane systemy zarządzania ERP/ERP II. Wykorzystanie w biznesie, wdrażanie*, Difin, Warszawa 2003.

<sup>14</sup> *Informatyka ekonomiczna*, red. S. Wrycza, PWE, Warszawa 2010.

<sup>15</sup> *Informatyka gospodarcza*, red. J. Zawila-Niedźwiecki, K. Rostek, A. Gąsiorkiewicz, C.H. Beck, Warszawa 2010.



Analizując literaturę krajową, zauważyć należy, że autorzy ograniczają się jedynie do wskazania miejsca procesu utrzymania w cyklu życia systemów informatycznych oraz krótkiego zdefiniowania samego pojęcia. Proces utrzymania w przeciwieństwie do początkowych faz cyklu życia systemów informatycznych (analiza, projektowanie, wdrażanie) nie jest szczegółowo charakteryzowany.

### **3. Praktyczne aspekty procesu utrzymania**

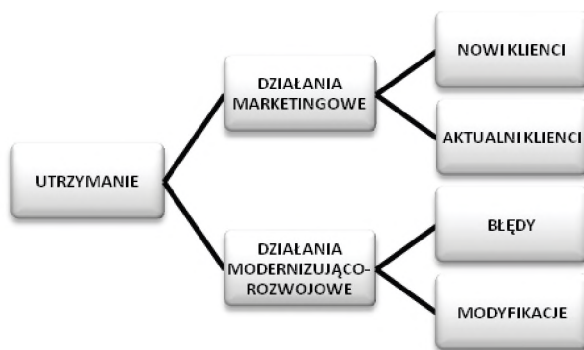
Za wzrostem znaczenia utrzymania zintegrowanych systemów informatycznych w wymiarze nie tylko teoretycznym, ale również praktycznym przemawiają współcześnie argumenty, które można zidentyfikować zarówno po stronie odbiorców, jak i dostawców tych systemów. Przedsiębiorstwa, które zdecydowały się na zakup i wdrożenie systemu zintegrowanego, na ogół ponoszą bardzo wysokie początkowe nakłady inwestycyjne związane z zakupem oprogramowania oraz niezbędnych usług powiązanych, takich jak: analiza przedwdrożeniowa, konfiguracja zmiennych parametrów systemu, szkolenia, asysty po uruchomieniu, serwis oprogramowania. Implikuje to chęć jak najdłuższego użytkowania danego systemu. Przemawia za tym również fakt, że wdrożenie systemu często wymusza reorganizację dotychczasowego środowiska pracy oraz zmianę zakresu obowiązków użytkowników systemu, czyli mówiąc inaczej, wprowadzenie daleko idących, na ogół kosztownych zmian organizacyjnych. Ponadto rzadko kiedy mamy do czynienia z sytuacją spełnienia wszystkich potrzeb użytkowników i nadziei pokładanych w systemie na etapie jego wdrożenia. Dopiero podczas eksploatacji pojawiają się nowe potrzeby czy wykrywane są niezidentyfikowane na wcześniejszych etapach błędy. Do tego dochodzi duża dynamika zmian otoczenia, w którym funkcjonuje dana organizacja, i to zarówno tego bliższego (konkurencyjnego, np. nowe potrzeby klientów, nowe formy współpracy z dostawcami), jak i dalszego (np. zmiany prawne), które na ogół nie uzasadniają w wystarczający sposób konieczności wdrożenia kolejnego systemu, a jedynie modyfikację już użytkowanego. Wszystko to przemawia za istotnością procesu utrzymania z perspektywy jego użytkowników.

Po stronie dostawcy oprogramowania można również znaleźć istotne argumenty wskazujące na duże znaczenie procesu utrzymania w bieżącej dzia-

łałości. Przede wszystkim współcześnie podkreśla się, że trzeba ponieść znacznie więcej nakładów na utrzymanie istniejącego oprogramowania niż na jego wytworzenie. Ponadto stosowane technologie umożliwiają znaczne wydłużenie cyklu życia SI poprzez zapewnienie dużego potencjału rozwojowego tworzonych rozwiązań oraz reużywalność komponentów programistycznych. Wskazane uwarunkowania technologiczne umożliwiają wielokrotne wdrażanie danego rozwiązania, co znacząco komplikuje proces jego utrzymania. Każdorazowa zmiana wprowadzona do systemu (rozwojowa lub eliminująca błędy) musi być bowiem rozpatrywana z punktu widzenia konsekwencji, jakie przyniesie dla konkretnych instalacji (w tym różnych parametrów konfiguracyjnych) i środowisk funkcjonowania. Ponadto zmiany systemu są inicjowane zarówno przez istniejących użytkowników, jak i oczekiwania rynku tak, aby system nie tylko sprawnie funkcjonował jako rozwiązanie już wdrożone, ale również jako konkurencyjny produkt rynkowy. Należy również zauważyć, że proces utrzymania wiąże się z określonymi korzyściami finansowymi w postaci nowych sprzedaży, opłat licencyjnych, umów serwisowych. Zwłaszcza te ostatnie są poszukiwanymi przez producentów źródłami dochodów, bowiem zapewniają stały i z góry określony przychód, który w czasach koniunktury może finansować rozwój systemu, a w czasach dekonunktury zapewniać finansowanie kosztów stałych prowadzonej przez producenta działalności.

Dla producenta ZSI faza utrzymania trwa nieprzerwanie od momentu zdobycia pierwszego klienta i wdrożenia u niego systemu do momentu wycofania go z rynku. W praktyce działalności dostawców ZSI obejmuje działania (patrz rysunek 2):

- a) marketingowe na rzecz pozyskania nowych klientów;
- b) z zakresu marketingu posprzedażowego na rzecz już pozyskanych klientów;
- c) modernizująco-rozwojowe polegające na ciągłym dostosowywaniu systemu do zmieniającego się otoczenia, potrzeb klientów oraz eliminowaniu błędów oprogramowania realizowane w ramach działalności wdrożeniowej i serwisowej.



Rys. 2. Składowe procesu utrzymania

Źródło: opracowanie własne.

W ramach działań marketingowych należy zwrócić szczególną uwagę na te działania, które mają na celu pozyskanie nowego klienta. Pozyskanie przyszłego użytkownika systemu wiąże się z prowadzeniem działań o charakterze handlowym, do których należy zaliczyć:

- a) telemarketing i inne działania (np. udział w konferencjach branżowych) pozwalające na penetrację rynku pod kątem wyłonienia potencjalnych użytkowników systemu informatycznego;
- b) spotkania z potencjalnymi użytkownikami, na których wstępnie rozpoznaje się potrzeby klienta, prezentuje system oraz firmę wdrożeniową;
- c) przygotowanie oferty na zakup oprogramowania, analizę przedwdrożeniową oraz samo wdrożenie (często razem z infrastrukturą sieciowo-sprzętową);
- d) negocjacje umowy wdrożeniowej wraz z warunkami finansowymi.

Często w praktyce równocześnie z umową wdrożeniową negocjuje się zapisy umowy serwisowej zapewniającej współpracę między dostawcą oprogramowania a klientem po zakończeniu procesu wdrożenia.

Z kolei głównym celem działań obsługi (marketingu) posprzedażowego, z uwagi na referencyjność branży ZSI (zapewnianie nowych sprzedaży na podstawie referencji od dotychczasowych użytkowników), jest przede wszystkim utrzymanie długofalowych, dobrych relacji z klientami. Inne zadania realizowane w ramach marketingu posprzedażowego to: informowanie o udostępnionych przez producenta poprawkach do oprogramowania lub nowych jego wersjach, badanie poziomu zadowolenia klientów, wysyłka folderów, ofert, cenników,

organizowanie targów, seminariów czy konferencji. Z wymienionych działań na szczególną uwagę zasługuje aktywność w zakresie organizowania konferencji. Zasadniczo w praktyce biznesowej mamy do czynienia z dwoma ich rodzajami, tj.:

- a) konferencjami dla potencjalnych użytkowników – mającymi na celu promowanie sprzedawanego produktu – systemu informatycznego, adresowanymi do wszystkich firm myślących o zmianie oprogramowania (są one częścią działalności przedwdrozeniowej);
- b) konferencje dla obecnych użytkowników systemu informatycznego, w tym ogólne i branżowe – mające na celu promowanie nowych funkcjonalności, wymianę wiedzy i doświadczeń z innymi użytkownikami oraz pracownikami, a także nawiązanie między pracownikami a klientami relacji interpersonalnych (są one częścią działalności powdrożeniowej).

Działania modernizujące i rozwojowe mają na celu zapewnienie, aby system przystawał do realiów prawno-gospodarczych oraz osiągnięć techniki i nauki. Mają za zadanie zapewnić odpowiednią wydajności systemu oraz poprawność jego działania. Przez działania modernizacyjne należy rozumieć działania mające na celu eliminowanie błędów systemu. Z kolei działania rozwojowe odnoszone będą do działań przyczyniających się do optymalizacji istniejących funkcjonalności lub ich rozbudowy i będą w skrócie nazywane modyfikacjami. Działania modernizacyjno-rozwojowe w praktyce obejmują:

- a) modyfikacje systemu podczas kolejnych wdrożeń zgodnie z potrzebami nowych klientów;
- b) modyfikacje systemu w ramach zgłoszeń serwisowych zgodnie z potrzebami aktualnych klientów;
- c) modyfikacje systemu zgodnie z potrzebami rynku na podstawie zgłoszeń o brakach, sugestii rozwojowych, analizy stanu prawnego czy rozwiązań konkurencyjnych;
- d) poprawianie błędów.

Jak widać, działania modernizacyjno-rozwojowe prowadzone w ramach procesu utrzymania ZSI wiążą się z jednej strony z zapewnieniem poprawności działania systemu, z drugiej – z pozyskiwaniem od klientów informacji o brakach w funkcjonalności systemu, które należy uzupełnić, i sugestii rozwojowych, które należy uwzględnić w nowych wersjach tak, aby system był adekwatny do potrzeb rynku. Producent może powołać zespół czuwający nad rozwojem produktu, który weryfikuje zgłoszenia klientów, na bieżąco śledzi zmia-

ny realiów formalnoprawnych i ustala, które z nich będą wprowadzane do systemu. Modyfikowanie systemu, zwłaszcza jeżeli skutkuje ono wydaniem nowej wersji ZSI, należy traktować jako odrębny projekt. Pojawienie się na rynku nowej wersji SI pociąga za sobą konieczność podjęcia przez producenta decyzji dotyczącej okresu wspierania starszych wersji oprogramowania (są to na ogół dwie wersje wstecz). Musi być ona ściśle powiązana z polityką udostępniania klientom nowych wersji.

Szeroko rozumiane modyfikowanie systemu zgodnie z potrzebami nowych, aktualnych klientów oraz potrzebami rynku może mieć dwojaki charakter: modyfikacji specyficznych i modyfikacji standardowych. Modyfikacje specyficzne są wykonywane w celu zrealizowania potrzeb konkretnego klienta i nie są dostępne dla pozostałych firm użytkujących ZSI. Natomiast modyfikacje standardowe są dostępne dla wszystkich klientów producenta ZSI i realizowane głównie przez niego.

Obok wprowadzania modyfikacji ważnym elementem działań związanych z utrzymaniem prowadzonych przez producenta ZSI jest zbieranie i weryfikowanie zgłoszeń dotyczących błędów. Wymaga to sprzężenia zwrotnego między producentem a użytkownikiem systemu i na ogół odbywa się za pośrednictwem konsultantów z działu wdrożeń, a przede wszystkim działu serwisu oraz partnerów wdrożeniowych.

Przez pryzmat możliwości oddziaływania na ostateczny kształt ZSI lub/i jego konkretnych instalacji utrzymanie ZSI ma odmienne znaczenie dla producenta oprogramowania oraz firm wdrożeniowych. W każdej z tych dwóch perspektyw inny element jest sednem procesu utrzymania ZSI. Dla producenta to przede wszystkim działania mające na celu ciągle dostosowywanie systemu do realiów prawno-gospodarczych oraz nowości technologicznych tak, aby system ZSI był jako produkt w ujęciu marketingowym konkurencyjny na rynku.

Konkurencyjność ta odnosi się zarówno do aktualnych użytkowników, którzy powinni mieć możliwość pracy na systemie nadążającym za zmianami zachodzącymi w otoczeniu makroekonomicznym, jak również do podmiotów dopiero poszukujących systemu informatycznego. Ma to bardzo istotne znaczenie, ponieważ branża ZSI jest, jak już wspomniano, silnie referencyjna, co w praktyce oznacza, że w dużej mierze zadowolony z systemu użytkownik generuje jego dalszą sprzedaż. Dla producenta jest to jedna z najważniejszych przesłanek utrzymywania systemu. Podobne przesłanki ma również firma wdrożeniowa. Ona również stara się poprzez utrzymywanie systemu zaspokoić

potrzeby aktualnych klientów z myślą o referencjach dla kolejnych sprzedaży. Niemniej jednak punkt ciężkości utrzymania SI w przypadku firmy wdrożeniowej jest położony w nieco innym miejscu niż w przypadku producenta. Firma wdrożeniowa nie jest właścicielem kodu systemu ZSI, który dystrybuuje na rynku, w związku z tym nie jest w stanie dostosowywać go do realiów makroekonomicznych czy najnowszych trendów technologicznych. Jest to, jak już wspomniano, zadanie producenta. Firma wdrożeniowa może się natomiast skoncentrować na modyfikacjach systemu w mniejszym zakresie, wynikających ze specyfiki funkcjonowania danego podmiotu. Zmiany tego rodzaju oczywiście mogą być z czasem wprowadzone przez producenta do kodu oprogramowania, jeśli ich zgłoszenia nie są przypadkami odosobnionymi. Niemniej jednak są one odłożone w czasie. W związku z tym działania tego rodzaju z natury rzeczy są podejmowane przez firmy partnerskie, które mogą odpowiedzieć na nie niemal na bieżąco. Są realizowane w postaci płatnych usług, jako modyfikacje specyficzne. Ponadto tego rodzaju niestandardowe rozwiązanie, które nie zostały wprowadzone do kodu systemu ZSI przez producenta, a są jedynie domeną danej firmy wdrożeniowej, mogą stać się cennym elementem zdobywania przewagi konkurencyjnej zwłaszcza w stosunku do podmiotów z określonej grupy (branży). Podsumowując dotychczasowe rozważania można, więc powiedzieć, że o ile producent jako właściciel kodu realizuje przede wszystkim potrzeby wynikające ze zmian makroekonomicznych, o tyle firma wdrożeniowa zapewnia realizację potrzeb wynikających ze specyfiki mikroekonomicznej danego podmiotu. Obu uczestnikom rynku ZSI przyświeca jedna idea – zadowolenie aktualnych użytkowników przede wszystkim w celu zdobycia referencji dla kolejnych sprzedaży systemu.

Z kolei klient – finalny użytkownik systemu – jest zainteresowany tym, aby system spełniał jego potrzeby, i to zarówno w wymiarze makro-, jak i mikroekonomicznym. Kolejne sprzedaże systemu, którego jest użytkownikiem, są dla jego interesów ekonomicznych obojętne. Mogą być istotne z punktu widzenia pewnych bieżących korzyści we współpracy z firmą, której udziela referencji, jednak w skali globalnej, jak już wspomniano, nie mają większego znaczenia. Ponadto z punktu widzenia klienta, zwłaszcza gdy nabywa system za pośrednictwem firmy wdrożeniowej, istotna jest świadomość, że swobodę we wprowadzaniu zmian, zarówno rozwojowych, jak i modernizacyjnych, do oprogramowania ma jedynie jego producent (rozumiany jako właściciel kodu), natomiast nie zawsze zdecyduje się on na ich wprowadzenie, ponieważ koszt takiej opera-

cji dla jednego użytkownika może się okazać zbyt duży lub po prostu zamiana ta nie jest zgodna z przyjętym planem rozwoju funkcjonalności ZSI. Wtedy istnieje jeszcze możliwość, że będzie to modyfikacja specyficzna, której wprowadzenie jest opłacalne dla firmy wdrożeniowej. Należy mieć w tym przypadku na uwadze jeszcze fakt, że wprowadzenie danej modyfikacji musi być zgodne z umową zawartą między producentem a firmą wdrożeniową, bowiem brak takiej zgodności może pozbawić klienta na przykład gwarancji producenta na posiadane oprogramowania.

### **Uwagi końcowe**

Podsumowując rozważania zawarte w artykule i biorąc pod uwagę zarówno analizę dorobku literaturowego, jak i praktyczne aspekty utrzymania ZSI, można je zdefiniować w następujący sposób – z perspektywy producenta utrzymanie ZSI to działania polegające na ciągłym dostosowywaniu systemu do realiów prawno-gospodarczych, najnowszych osiągnięć technologicznych, a przede wszystkim potrzeb aktualnych i potencjalnych użytkowników połączone z bieżącym eliminowaniem błędów oprogramowania. Dla firmy wdrożeniowej utrzymanie SI to działania polegające na ciągłym dostosowywaniu poszczególnych instalacji systemu do zmiennych w czasie potrzeb jego użytkowników w zakresie i formie dopuszczonej umową zawartą z producentem oprogramowania z uwzględnieniem potrzeb przyszłych, potencjalnych użytkowników połączone z bieżącą pomocą w zakresie eliminowania błędów oprogramowania. Z kolei z perspektywy klienta utrzymanie to działania polegające na dostosowywaniu specyficznie skonfigurowanej pod jego potrzeby wersji systemu do realiów prawno-gospodarczych, najnowszych osiągnięć technologicznych, a przede wszystkim nowo pojawiających się potrzeb przy współpracy oraz na zasadach i w granicach określonych w umowach z firmą wdrożeniową i producentem systemu połączone z bieżącym eliminowaniem błędów oprogramowania.

## Literatura

- Abran A., Moore J.W., *Guide to the Software Engineering Body of Knowledge (SWE-BOK)*. 2005 Version. IEEE Computer Society: Los Alamos, California, 2005.
- April A., Abran A., *Software Maintenance Management, Evolution and Continuous Improvement*, John Wiley & Sons, Hoboken 2008.
- Beynon-Davies P., *Information systems. An Introduction to Informatics in Organisations*. PALGRAVE. Houndmills–New York 2002.
- Boehm B.W., *Industrial Software Metrics Top 10 List*, IEEE Software 1987.
- Capers J., *Estimating Software Costs*, McGraw-Hill 2007.
- Informatyka ekonomiczna*, red. S. Wrycza, PWE, Warszawa 2010.
- Informatyka gospodarcza*, red. J. Zawila-Niedźwiecki, K. Rostek, A. Gąsioriewicz, C.H. Beck, Warszawa 2010.
- Institute of Electrical and Electronics Engineers. IEEE Standard for Software Maintenance, Standard 1219*, Institute of Electrical and Electronics Engineers, New York 1998.
- International Standards Organization. Information Technology. Software Life Cycle Processes*, ISO/IEC Standard 12207. International Organization for Standardization: Geneva, Switzerland, 1995.
- Kisielnicki J., Sroka H., *Metody projektowania i wdrażania systemów*, Placet, Warszawa 2001 i 2005.
- Lech P., *Zintegrowane systemy zarządzania ERP/ERP II. Wykorzystanie w biznesie, wdrażanie*, Difin, Warszawa 2003.
- Lehman M.M., *The Future of Software – Managing Evolution*, inv.contr., vol. 15, no. 1, IEEE Softw., Jan-Feb 1998.
- Loudon K.C., Loudon J.P., *Management Information Systems. Managing The Digital Firm*, Prentice Hall, Upper Saddle River 2006.
- Martin J., C. McClure, *Software Maintenance. The Problem and Its Solutions*, Prentice Hall, Englewood Cliffs 1983.



**INTEGRATED INFORMATION SYSTEMS MAINTENANCE PROCESS  
IN THEORY AND PRACTICE**

**Summary**

The article reviews and analyzes integrated information systems maintenance process in theory and practice. The author shows how the approach to the maintenance of IS is changing in recent years in the foreign and domestic literature. In the end author characterizes the importance of maintaining process in practice of IT vendors and their customers.

*Translated by Maja Leszczyńska*