

Krzysztof Gruszczyński, Jacek Małyszko

Uniwersytet Ekonomiczny w Poznaniu, Wydział Informatyki i Gospodarki Elektronicznej, Katedra Informatyki Ekonomicznej

Autor do korespondencji: Krzysztof Gruszczyński, k.gruszczyński@live.com

WPŁYW AWARII SYSTEMÓW INFORMATYCZNYCH NA WYDAJNOŚĆ PROCESÓW BIZNESOWYCH PRZEDSIĘBIORSTWA

Streszczenie: Artykuł podejmuje temat badania wpływu liczby awarii systemu informatycznego na wydajność procesów biznesowych przedsiębiorstwa, które korzysta z awaryjnego oprogramowania. Autorzy stawiają tezę, że wzrost liczby awarii w danym dniu w mierzalny sposób może być powodem notowania wysokich strat przez użytkownika końcowego. Badanie przeprowadzono na podstawie rzeczywistych danych uzyskanych od firmy KRUK S.A., które składają się z dwóch zbiorów: logów systemowych oraz danych operacyjnych. Poprzez szukanie odstępstw w obu zbiorach i analizę przyczyn defektów podjęto próbę identyfikacji dni, w których wydajność procesu wykazywała wysoką zależność od liczby awarii. Wykazano w ten sposób, że możliwe jest zmierzenie poziomu odstępstwa wydajności procesu w konsekwencji wystąpienia licznych awarii oprogramowania.

Słowa kluczowe: niezawodność oprogramowania, koszt awarii, wydajność procesów.

Klasyfikacja JEL: L86, C88.

SOFTWARE FAILURE IMPACT ON A COMPANY'S BUSINESS PROCESS EFFICIENCY

Abstract: The article discusses the subject of measuring the influence of software failure rate on a company's business process efficiency. We put forward the thesis that it is possible to measure the impact of increasing failure rate on a user's financial

condition. The research is based on real data provided by KRUK S.A., which consists of two subsets: system failure logs and operational data. By searching for data outliers and by conducting a causal analysis, we tried to identify the days, which presented a high dependence of process efficiency on the observed failure rate. As a result, we have shown that it is possible to discover a measurable outlier in the process efficiency that is caused by a specific set of failures.

Keywords: software reliability, failure cost, performance, process efficiency.

Wstęp

Awarie są nieodłącznym elementem współczesnych złożonych systemów informatycznych. Mogą być powodowane przez szereg czynników – wewnętrznych (błąd w algorytmie programu) lub zewnętrznych (awaria infrastruktury informatycznej, dostępu do zdalnego interfejsu i wiele innych). Część z nich nie niesie ze sobą większych konsekwencji – umożliwiają kontynuację procesu biznesowego lub obranie alternatywnej, równie efektywnej ścieżki procesu. Zdarzają się jednak również awarie, których skutkiem jest zablokowanie całego procesu biznesowego, co może mieć już istotniejszy wpływ na funkcjonowanie przedsiębiorstwa.

Liczne awarie systemów informatycznych niosą więc negatywne konsekwencje dla ich użytkowników. W przypadku konieczności naprawy defektu koszty takich działań ponosić może producent oprogramowania. Jednak spadek wydajności procesu zwiększa również koszt alternatywny dla użytkownika końcowego. Możliwość zmierzenia tej straty jest w centrum zainteresowania prowadzonego badania. Pozostałe typy kosztów, tj. powstające po stronie producenta oprogramowania, są poza poruszonymi rozważaniami w tym tekście.

Celem artykułu jest weryfikacja, czy sposób, w jaki awarie systemów informatycznych wpływają na wydajność procesów biznesowych realizowanych przez przedsiębiorstwo, jest mierzalny. Jako miarę tej wydajności przyjęto liczbę zakończonych z sukcesem procesów biznesowych w ramach badanego przypadku. Kryterium uznania, że wydajność procesu biznesowego w pewnym stopniu zależy od poziomu awaryjności oprogramowania, jest identyfikacja konkretnych dni wykazujących korelację ujemną między badanymi obserwacjami wydajności procesu biznesowego a liczbą awarii.

Aby przeprowadzić badania, pozyskano rzeczywiste dane gromadzone przez przedsiębiorstwo KRUK S.A. Składają się one z dwóch zbiorów – danych operacyjnych, zawierających wartości pozwalające oszacować wydajność

procesu biznesowego, oraz logów systemowych, umożliwiającą zliczenie i klasyfikację występujących typów awarii. Jeśli w określonym dniu zaobserwowano znaczny spadek wydajności procesu biznesowego przy gwałtownym wzroście liczby awarii, to jest to podstawa, aby przypuszczać, że istnieją zależności między danymi z tych dwóch zbiorów. Eksperycka weryfikacja, potwierdzająca przyczynę spadku wydajności procesu biznesowego w awariach konkretnych typów odzwierciedlonych w logu systemowym, jest finalnym krokiem zaproponowanej procedury. Wymaga ona udziału osób znających budowę systemu informatycznego oraz posiadających wiedzę biznesową w celu przeanalizowania charakterystyki zaobserwowanych awarii (źródła, rodzaju, zasięgu) i ich możliwego wpływu na wydajność procesu.

Artykuł złożony jest z pięciu części. Pierwsza dotyczy analizy literatury dziedzinowej w celu przedstawienia wyników wcześniejszych badań. W drugiej przedstawiono procedurę przygotowania danych do eksperymentu. Trzecia część opisuje procedurę, według której zostały przeprowadzone badania, oraz narzędzia informatyczne, które zostały wykorzystane w trakcie badań. W części czwartej opisano wynik przeprowadzonego eksperymentu dla dwóch różnych procesów biznesowych. Artykuł kończy się omówieniem wniosków z badań i propozycją potencjalnych dalszych kierunków badań.

1. Analiza literatury

Szacowanie wpływu awarii oprogramowania biznesowego na procesy realizowane przez użytkownika stanowi poważne wyzwanie, zwłaszcza gdy odbiorca wadliwej aplikacji natrafia kilkakrotnie w codziennej pracy na wystąpienie defektów o różnej skali szkodliwości. W swej pracy Zhivich i Cunningham [2009] zaprezentowali najbardziej spektakularne awarie systemów w historii, które spowodowały ekstremalnie wysokie koszty w skali światowej w postaci strat rzędu miliardów dolarów. Przedstawiono również przykłady awarii powodujących jeszcze tragiczniejsze skutki, których efekt jest niemierzalny – utratę ludzkiego życia. W literaturze można spotkać wiele publikacji dotyczących kosztów błędów aplikacji dla producenta, który jest za nie odpowiedzialny. Grottke i Graf [2009] opisują model szacowania wewnętrznych (przed wdrożeniem oprogramowania w środowisku produkcyjnym) i zewnętrznych (po przekazaniu systemu użytkownikowi) kosztów dla dostawcy systemu – utworzenia i rozpropagowania odpowiedniej poprawki, zapewnienia wsparcia odbiorcy i innych.

W swoim artykule Brosch, Gitzel i Koziol [2010] wprowadzili metodę szacowania niezawodności wraz z finansowymi konsekwencjami w sytuacji,

gdy awaria systemu jest przyczyną obrania alternatywnej ścieżki procesu biznesowego. Opracowany model bazuje na dwóch charakterystykach wymaganych do estymacji: prawdopodobieństwie wystąpienia awarii (która prowadzi do wykonania awaryjnego scenariusza procesu) i całkowitym koszcie zaistniałego defektu. Metoda jest dwustopniowa: w pierwszej kolejności zamodelowana zostaje warstwa biznesowa z wykorzystaniem notacji BPMN. Na tę warstwę składają się scenariusze procesu zarówno w normalnych, jak i wyjątkowych (awaryjnych) warunkach. Również warstwa IT zostaje zamodelowana, tym razem przy użyciu notacji Palladio Component Model (PCM). Drugi krok polega na określeniu niezawodności i kosztów w kontekście zaprojektowanych warstw. Faza ta wymaga transformacji PCM w proces Markowa z czasem dyskretnym (*Discrete-Time Markov Chain* – DTMC). DTMC przedstawia wszystkie dostępne ścieżki wykonania i ich prawdopodobieństwa. Aby uzyskać informacje dotyczące kosztów, rozważane są dwa czynniki jako mające na nie wysoki wpływ: czasy przestoju oraz czas dodatkowej pracy. Aby wyliczyć ostateczny koszt awarii, wprowadzono następujący wzór:

$$C = \sum_{i=1}^n p_i \cdot (c_i + d_i), \quad (1)$$

gdzie:

- p_i – oznacza prawdopodobieństwo awarii,
- c_i – koszty pracy,
- d_i – czas przestoju.

Gitzel, Krug i Brhel [2010] zaprezentowali inny przykład modelu szacowania kosztów awarii dla użytkownika. Jako mające największy wpływ na użytkownika zostały zdefiniowane dwa główne typy awarii: utrata danych i przerwanie pracy aplikacji. Kroki we wdrażaniu modelu są następujące:

1. Zidentyfikuj determinanty kosztów (w większości przypadków jest to czas przestoju pracowników ściśle powiązany z profilami operacyjnymi).
2. Zamodeluj wystąpienia awarii w powiązaniu z wyzwalającymi je zdarzeniami.
3. Zamodeluj zachowanie użytkowników (wzorce użytkowania aplikacji przez różne grupy w przedsiębiorstwie – profile operacyjne).
4. Oblicz szacowany koszt dla użytkownika. W tym celu wyprowadzono dwa kluczowe wzory:

$$T_i = n_i \cdot t_i \cdot R_i \cdot P, \quad (2)$$

gdzie

- T_i – szacowana strata wyrażona w roboczogodzinach,
- n_i – spodziewana częstotliwość awarii,
- t_i – czas trwania,
- R_i – czas naprawy,
- P – prawdopodobieństwo.

$$C = C^r + \sum_{i=1}^N (T_i \cdot C_i^m), \quad (3)$$

gdzie:

- c^r – czas naprawy,
- C_i^m – koszt pracownika.

Powyższy model poparto przykładem – kalkulacją kosztów dla fikcyjnego przedsiębiorstwa korzystającego z przeglądarki internetowej Firefox. Jest to darmowe oprogramowanie i jego baza danych awarii jest dostępna dla wszystkich użytkowników. Niemniej wykazanie zasadności obranej metody wymagałoby przeprowadzenia bardziej rozbudowanego studium przypadku.

Wymienione modele mogą być użyteczne w sytuacji posiadania kompletu informacji o przebiegu konkretnej instancji procesu. Nie wyjaśniają, w jaki sposób zidentyfikowano konkretną awarię jako zdarzenie zakłócające normalny scenariusz procesu. Wymagają utworzenia modeli procesów i są czasochłonne we wdrażaniu. Aby przeanalizować faktyczny wpływ występujących awarii na finanse przedsiębiorstwa, konieczne jest szersze wykorzystanie rzeczywistych danych operacyjnych, pozyskanych od podmiotów działających na rynku. Na takiej podstawie możliwe będzie śledzenie rzeczywistej wydajności procesów bez konieczności ręcznego modelowania procesów, stanowiącego jeden z głównych kroków istniejących aktualnie metod.

Analiza przyczyn oraz skutków występowania awarii wymaga opracowania odpowiedniego modelu klasyfikacji awarii. Umożliwia on również usprawnienie całego procesu tworzenia oprogramowania – opracowane zostają standardy projektowania aplikacji, wskazówki dla testerów, metody weryfikacji i walidacji systemu, metody wdrażania modeli wzrostów niezawodności [Bowen 1980]. Leszak, Perry i Stoll [2002] omówili metodę klasyfikacji bazującą na danych ze zgłoszeń z prośbą o poprawę defektu w trakcie prac nad optyczną siecią transmisyjną. Metoda ta składa się z opisu fazy odkrycia defektu, jego typu, faktycznej lokalizacji, wyzwalacza. Charakterystyki te są następnie wykorzystane w analizie przyczyn źródłowych (*Root Cause Analysis* – RCA). Bowen [1980]

prezentuje złożony przegląd kilku dobrze znanych modeli klasyfikacji opracowanych przez różne organizacje – TRW-Redondo Beach, Mitre Corporation i inne, jednak ze względu na postęp nowoczesnych środowisk wytwarzania oprogramowania część atrybutów modelu nie znajduje aktualnie zastosowania. Można z nich wyodrębnić trzy wspólne kategorie: dotkliwość, przyczynę i źródło. Klasyfikacja dla oprogramowania wysokiej integralności w pracy Hechta i Wallace [1996] składa się z elementów specyficznych dla defektów, awarii i środowiska. Dla awarii są to odpowiednio tryb, czas wystąpienia, operacja trwająca w momencie wystąpienia, wyzwalacz, czas trwania, zasięg, krytyczność, komponent źródłowy, mechanizm detekcji, mechanizm poprawiający, mechanizm przywrócenia, prewencja i zalecenia. Schemat klasyfikacji opracowany przez HP [Freimut, Denger i Ketterer 2005] składa się z trzech głównych elementów: źródła (czynność prowadząca do wykrycia), typu, trybu (przyczyna defektu). Dwustopniowy model klasyfikacji został zaprezentowany w pracy Jaina i współpracowników [2008]. Podczas pierwszego etapu następuje ekstrakcja atrybutów umożliwiających zdefiniowanie symptomów awarii (czas trwania, wielkość grupy, rodzaj awarii – oprogramowania, sprzętu, sieci). W drugim kroku nacisk jest położony na atrybuty niskiego poziomu, jak lokalizacja i poziom zużycia zasobów, pozwalając zdefiniować przyczynę awarii.

IEEE Computer Society wprowadziło własny standard klasyfikacji anomalii oprogramowania obejmujący kategorie defektów oraz awarii [IEEE 2010]. Stanowi on złożony zbiór atrybutów wraz z przykładowymi wartościami o wysokim poziomie abstrakcji, które można również znaleźć we wcześniej omówionych modelach (typ, tryb, dotkliwość, opis, aktywność wprowadzająca, aktywność umożliwiająca wykrycie i inne). Dodatkowo uwzględniona została perspektywa niższego poziomu z atrybutami, takimi jak data otwarcia zgłoszenia awarii, zamknięcia, zaobserwowania, typ konfiguracji, środowiska i inne. Każdy defekt jest powiązany z awarią, która go uzewnętrzniała, umożliwiając tym samym analizę awarii na możliwie największym poziomie szczegółowości.

2. Przygotowanie danych

W celu przeprowadzenia badań z zakresu zależności wydajności procesów biznesowych od awarii systemów informatycznych pozyskano następujące rodzaje danych: 1) dane operacyjne, będące źródłem wartości umożliwiających oszacowanie wydajności procesu biznesowego oraz 2) logi zdarzeń systemowych, które przechowują informacje o występujących w danym dniu

awariach oprogramowania wspierającego proces biznesowy. Oba zbiory pozyskano z baz danych KRUK S.A., jednego z trzech największych na świecie przedsiębiorstw działających na rynku windykcji należności.

W pierwszej kolejności przeanalizowano dostępne zbiory danych awarii. W przypadku KRUK S.A. dysponowano następującymi zasobami:

1. Baza danych systemu BugTracker, która jest uzupełniana wieloma szczegółowymi informacjami odnośnie do wystąpienia awarii (dane użytkownika, systemu operacyjnego, data wystąpienia, stos wywołania, komponent, klasa i metoda wywołująca wyjątek i wiele innych) w konkretnych systemach zintegrowanych z automatem śledzenia błędów.
2. Baza danych systemu ServiceDesk, w której znajdują się ogólne dane awarii wygenerowanych przez BugTracker oraz inne rodzaje problemów zgłaszane przez użytkowników w ramach bieżącego utrzymania systemów i infrastruktury IT, trafiające następnie do odpowiednich serwisantów (programiści, administratorzy lub helpdesk).
3. Logi w formie tekstowej, które mają różnorodną strukturę (w zależności od aplikacji) i zapisywane są na dysku komputera, na którym wystąpiła awaria.

Dane o jednolitej strukturze znajdują się w bazie danych systemu BugTracker i to one posłużą do przeprowadzenia eksperymentu. Każdy wyjątek w oprogramowaniu zintegrowanym z BugTracker jest zapisywany jako odrębny rekord. Eksperyment przeprowadzono na podstawie danych z okresu 3.01.2011–15.09.2015, na które składało się w sumie około 1,3 miliona rekordów dotyczących 14 systemów. Dane awarii mają charakter ilościowy – poważniejsze problemy na potrzeby eksperymentu zdefiniowano jako zdarzenia generujące większą liczbę wystąpień awarii konkretnego typu w danym dniu. Podejście ilościowe spowodowane jest tym, że na podstawie samego logu w zdecydowanej większości nie można ocenić, jaki faktycznie skutek spowodowała dana awaria. Należy jednak zaznaczyć, że nie w każdej sytuacji takie podejście może być prawidłowe – sprawdzi się ono wyłącznie dla procesów realizowanych przez większą liczbę użytkowników, co gwarantuje zauważalny wzrost liczby awarii w określonym dniu w skali całego systemu. Dla pojedynczych, ale bardzo istotnych z perspektywy przedsiębiorstwa instancji procesów wymagane byłoby indywidualne podejście. Cechowałoby je konieczność określenia skali szkodliwości odkrytego defektu i próby przypisania klasy awarii do konkretnego procesu biznesowego, na którego wydajność wpłynęła. Takie działania wykraczają jednak poza zakres artykułu.

W drugiej kolejności przeanalizowano operacyjne bazy danych, w których zapisywane są informacje dla każdego z 14 systemów. Dane poszczególnych

procesów biznesowych umieszczone są w różnych tabelach, stąd ich ekstrakcja wymagała opracowania odpowiednich skryptów SQL przez ekspertów dziedzinowych (programistów znających konkretną aplikację). Aby poradzić sobie ze złożonością sytuacji, na potrzeby eksperymentu wybrano dwa przykładowe procesy realizowane z dużą częstotliwością przez wielu pracowników firmy. Jako miarę wydajności procesu biznesowego zdefiniowano dla nich liczbę pomyślnie ukończonych działań w danym dniu (w KRUK S.A. może to być przykładowo liczba podpisanych porozumień ratalnych, odbytych wizyt u klienta, udanych połączeń telefonicznych). Dane gromadzone są przez dwa różne systemy w jednej bazie danych. W bazie BugTracker awarie generowane przez te aplikacje mają osobne oznaczenie (identyfikator systemu), co ułatwia dalszą analizę.

3. Procedura przeprowadzenia eksperymentu

Mając odpowiednio przygotowane dane, przeprowadzono eksperyment, którego celem było wykazanie, że posiadając dane liczbowe dotyczące wydajności procesu i liczby w awarii w podziale na konkretne dni, można przeprowadzić analizę dotyczącą ich zależności. Zależność zostanie wstępnie potwierdzona, jeśli wzrost awaryjności systemu informatycznego w danym dniu znajdzie odzwierciedlenie w danych operacyjnych, czyli spowoduje spadek wydajności procesu biznesowego. Całkowite potwierdzenie lub odrzucenie zależności następuje po przeanalizowaniu źródła i cech konkretnych awarii w zbiorze obserwacji. Procedura przeprowadzenia eksperymentu ma następujący przebieg:

1. Zdefiniowanie długości okresów (w dniach), które zostaną wyodrębnione w ramach zbioru. Logi awarii oraz dane operacyjne mogą pochodzić z długiego okresu, podczas którego systemy oraz procesy ewoluowały. Porównywanie wartości oddzielonych od siebie o kilka miesięcy lub lat niesie ryzyko zaburzenia wyników – przykładowo wartość z okresu wcześniejszego, w którym stanowiła odstępstwo, w kolejnym okresie może być normą. Istotny jest dobór na tyle wąskiego okresu, aby zminimalizować ryzyko występowania wysokiej zmienności podczas operowania na takim zbiorze, jednak wymagany jest również odpowiednio szeroki zbiór w celu identyfikacji zakresu standardowych wartości.
2. Wyeliminowanie danych operacyjnych powyżej mediany w ramach każdego okresu. Jeśli wydajność procesu biznesowego przewyższa medianę w danym dniu, to z wysokim prawdopodobieństwem można domniemywać, że proces biznesowy nie był zakłócany przez awarie systemu. Takie

obserwacje nie niosą wartości dodanej dla dalszych badań, dlatego zostały odrzucone. W efekcie analizę przeprowadzono na podstawie obserwacji wydajności procesu biznesowego poniżej mediany.

3. Normalizacja danych liczby awarii i wydajności procesu biznesowego. Aby ułatwić dalszą analizę, wartości określające wydajność procesu biznesowego i liczbę awarii zostały znormalizowane dla każdego z okresów metodą skalowania względem standardowego odchylenia średniokwadratowego [Abdi 2010; Jayalakshmi i Santhakumaran 2011].
4. Identyfikacja odstępstw w liczbie awarii i wydajności procesu biznesowego w podzbiorach danych (okresach). Odstępstwa w wartościach określających wydajność procesu biznesowego i liczbach awarii zostały wskazane z wykorzystaniem kryterium Peirce'a [Ross 2003]. Jeśli w danym okresie wystąpił dzień, w którym zaobserwowano odstępstwo w liczbie awarii i wydajności procesu biznesowego, to taki stan nazwano silnym odstępstwem. Dla odstępstw występujących w danym dniu tylko w ramach jednego zbioru wprowadzono określenie odstępstwa słabego.
5. Wizualizacja znormalizowanych danych na wykresach. Za pomocą wykresów punktowych oraz złożonych (kolumnowych z liniowymi) przedstawiono dane dla każdego okresu w celu weryfikacji zidentyfikowanych silnych odstępstw. Wykresy umożliwiają również analizę słabych odstępstw, dla których zależność wydajności procesu biznesowego od wystąpienia awarii jest wysoce prawdopodobna.
6. Wybór dni, w których zaobserwowano odstępstwa. Wszystkie silne odstępstwa są z założenia kwalifikowane do zbioru wynikowego. Dodatkowo dołączono do nich wszystkie słabe odstępstwa, które kryterium Peirce'a wskazało w wartościach określających wydajność procesu biznesowego i po analizie danych na wykresach zaobserwowano wysoki poziom liczby awarii w danym dniu (jednak niezidentyfikowany jako odstępstwo).
7. Weryfikacja wyników z wykorzystaniem wiedzy eksperckiej. Jest ona finalnym krokiem i ma na celu określenie, czy odstępstwo w wydajności procesu biznesowego w danym dniu mogło być spowodowane przez konkretne awarie, o których informacje znajdują się w logach. Analiza wymaga wiedzy eksperckiej dotyczącej badanego procesu biznesowego oraz szczegółów funkcjonowania oprogramowania, które go wspiera.

W celu zautomatyzowania eksperymentu, opracowano aplikację w języku C# (platforma .NET) przyjmującą na wejściu trzy zbiory danych: liczbę awarii, wartości określające wydajność procesu biznesowego oraz daty obserwacji. Program umożliwia manipulowanie długością okresu oraz odrzucenie

wartości powyżej mediany. Z wykorzystaniem biblioteki zewnętrznej (Sasa. Numerics) identyfikowane są odstępstwa w danych według kryterium Peirce'a, które można zweryfikować na diagramach generowanych przez aplikację.

4. Wynik eksperymentu i jego interpretacja

Eksperyment przeprowadzono na podstawie danych dotyczących dwóch różnych procesów biznesowych oraz odpowiadających im dwóch osobnych zbiorów logów awarii. Pierwszy przypadek opiera się na procesie wysyłania zapytań między pracownikami różnych działów w celu pozyskania informacji lub podjęcia decyzji przez kompetentne osoby. Przypadek drugi dotyczy jednego z najważniejszych procesów zachodzących w firmie KRUK S.A. związanego z kluczowym produktem – ugodą z klientem. Oba procesy wspierane są przez dwie aplikacje zintegrowane z systemem BugTracker, dzięki czemu struktura logów jest jednorodna.

Analiza przypadku 1

Do analizy pierwszego procesu przygotowano dane operacyjne z okresu 1.10.2014–10.10.2015 oraz odpowiadające im logi awarii z systemu BugTracker (o tożsamym okresie dla aplikacji wspierającej badany proces). Wylimitowano ze zbioru dni wolne od pracy (soboty, niedziele oraz święta), w wyniku czego pozostały 254 rekordy.

Dane zostały przekazane jako wartości wejściowe do aplikacji szukających odstępstw, utworzonej na potrzeby eksperymentu. Wylimitowano rekordy z wartościami określającymi wydajność procesu biznesowego przewyższającymi medianę. Zdefiniowano dwie długości okresów, aby ustalić rząd danych umożliwiający jak najlepszą identyfikację odstępstw: było to odpowiednio 30 i 60 dni. Dla dłuższego okresu przedstawiono wyniki w tabeli 1. Wydajność procesu biznesowego oraz liczba awarii mają postać znormalizowaną dla zwiększenia przejrzystości i umożliwienia zwizualizowania obu zbiorów na jednym wykresie. W dniu 29.12.2014 roku zaobserwowano silne odstępstwo w liczbie występujących awarii i wydajności procesu biznesowego.

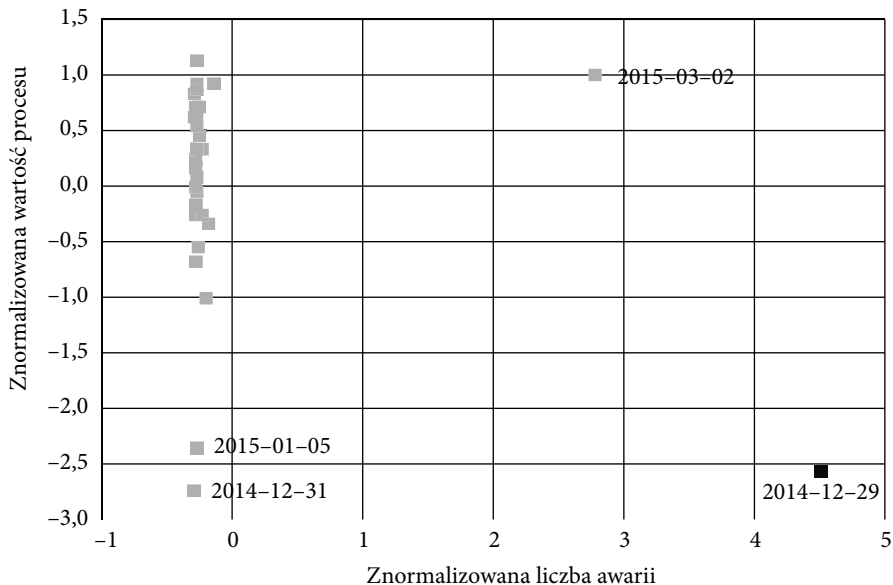
Dla okresu 30 dni aplikacja odkryła 3 dni z odstępstwami w wydajności procesu biznesowego (10.11.2014, 24.12.2014, 14.08.2014). Są to jednak słabe odstępstwa i w konsekwencji wymagają bardziej pogłębionej analizy w celu weryfikacji wpływu awarii na proces. Przy zdefiniowaniu okresu 60-dniowego aplikacja zidentyfikowała 6 odstępstw w wartościach określających wydajność

procesu biznesowego, przy czym jedno z nich jest silne (w dniu 29.12.2014). Wszystkie odstępstwa odkryte w dłuższym okresie (60 dni) zawierają również te z okresu krótszego (30 dni).

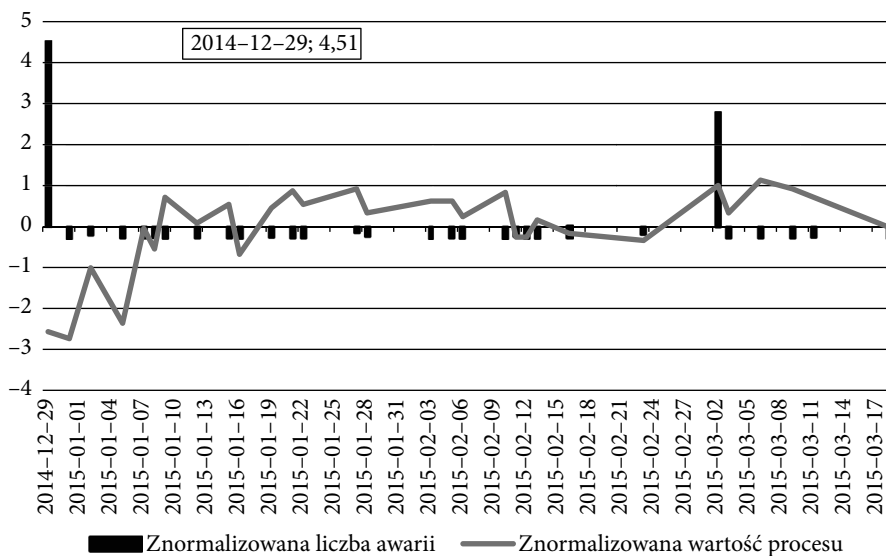
Tabela 1. Wartości wynikowe dla przypadku 1 przy długości okresu 60 dni

Data	Liczba awarii	Wydajność procesu biznesowego	Okres	Odstępstwo dla awarii	Odstępstwo dla procesu
2014-11-10	-0,38	-3,38	0	nie	tak
2014-12-24	-0,37	-3,19	0	nie	tak
2014-12-29	4,51	-2,57	1	tak	tak
2014-12-31	-0,29	-2,74	1	nie	tak
2015-01-05	-0,27	-2,36	1	nie	tak
2015-08-14	-1,21	-3,91	3	nie	tak

Weryfikację odkrytego silnego odstępstwa przeprowadzono na danych wynikowych z okresu 60-dniowego. W tym celu wygenerowano dwa rodzaje wykresów dla okresu, w którego obrębie znajduje się odstępstwo.



Rysunek 1. Wyniki dla okresu 1 badanego przypadku na wykresie punktowym. Wyraźnie widać wartości odstające, odzwierciedlające dni z dużą liczbą awarii oraz z niską wydajnością procesu biznesowego



Rysunek 2. Wyniki dla okresu 1 badanego przypadku na wykresie złożonym. W dniu 29.12.2014 zaobserwowano wysoki wzrost liczby awarii przy spadku wydajności procesu biznesowego

Analizując rysunki 1 oraz 2, w dniu 29.12.2014 roku zauważamy mocny spadek wydajności procesu biznesowego przy silnym wzroście liczby awarii systemu, co pokrywa się ze wskazanym wynikiem przez algorytm Peirce'a. Podobnie mocny spadek wydajności procesu biznesowego nastąpił 31.12.2014 roku – prawdopodobnie z powodu ostatniego dnia roku, i co się łączy z wieloma urlopami pracowników, oraz 5.01.2015 roku – w poniedziałek przed Świętem Trzech Króli, gdy większość osób również była niedostępna w pracy. W ciągu dwóch przytoczonych dni nie zaobserwowano wzrostu liczby awarii.

W dniu zaobserwowanego silnego odstępstwa BugTracker wygenerował 2350 zgłoszeń dotyczących aplikacji wspierającej proces, którego dotyczy eksperyment. Cały log składa się z pięciu rodzajów awarii, których przyczyną jest jedna – problem z połączeniem sieciowym do serwera między aplikacją a zewnętrznym systemem, z którym jest zintegrowana. Problem trwał od godziny 15:47 i w tym dniu proces nie mógł funkcjonować według normalnego scenariusza – działanie oprogramowania było w całości uniemożliwione. Użytkownicy nie mogli realizować procesu biznesowego według normalnej ścieżki polegającej na wymianie przepływu zadań z wykorzystaniem systemu – wymiana danych została w całości zablokowana i wymagała od pracowników znalezienia innych sposobów komunikacji. W tabeli 2 przedstawiono fragment

logu ze wszystkimi rodzajami awarii zaobserwowanymi w ciągu dnia. Rodzaj awarii nadawany jest automatycznie przez system BugTracker na podstawie stosu wywołania aplikacji, który spowodował awarię. Wszystkie awarie tego samego typu w danym dniu posiadają jednakowy identyfikator (rodzaj).

Tabela 2. Typy awarii zidentyfikowane 29.12.2014*

Data pierwszego zgłoszenia	Treść	Typ	Komponent
2014-12-29 15:47	The operation has timed out	System.Net. WebException	Kruk.CaseRequests. Processing
2014-12-29 15:49	The operation has timed out	System.Net. WebException	Kruk.CaseRequests. Processing
2014-12-29 15:55	Unable to connect to the remote server	System.Net. WebException	Kruk.CaseRequests. Processing
2014-12-29 15:55	Unable to connect to the remote server	System.Net. WebException	Kruk.CaseRequests. Processing
2014-12-29 18:13	Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding	System.Data. SqlClient. SqlException	Kruk.CaseRequests. Processing

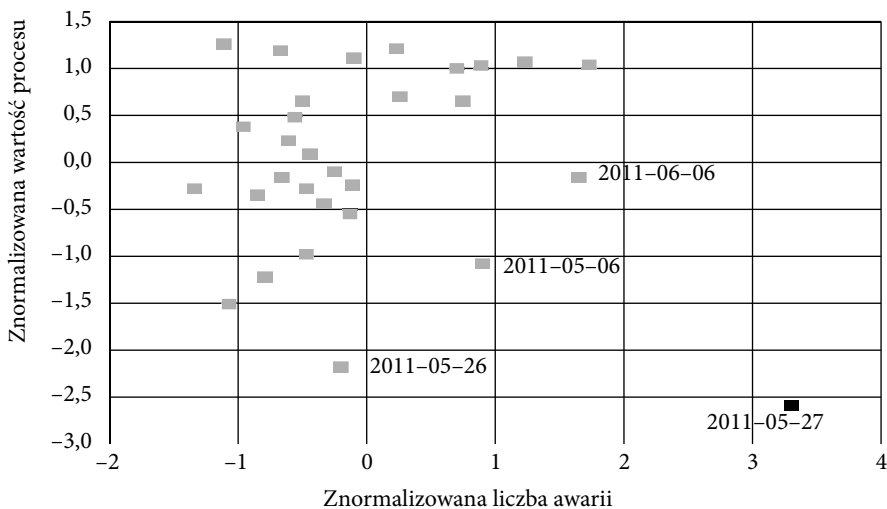
* Wszystkie są spowodowane przez problemy związane z infrastrukturą informatyczną.

Analiza przypadku 2

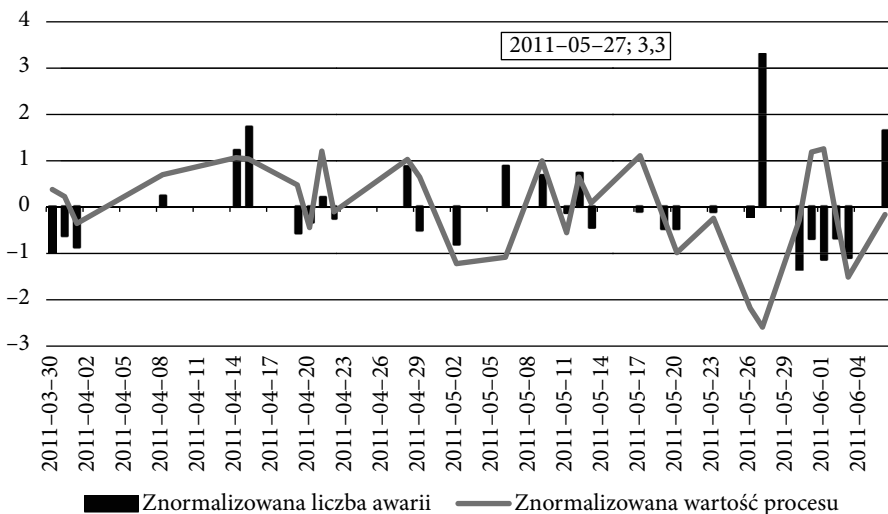
Analizę przypadku 2 oparto na danych operacyjnych i awarii z okresu 3.01.2011–01.10.2015, łącznie 1198 dni. Podobnie jak w poprzednim przypadku odfiltrowano dni wolne od pracy i odrzucono dni z wydajnością procesu biznesowego powyżej mediany dla założonego okresu 60 dni. Aplikacja identyfikująca odstępstwa odkryła 22 dni z odstępstwami w wartościach określających wydajność procesu biznesowego, w tym dwa odstępstwa były silne.

W dniu 24.12.2012 roku zaobserwowano jedno z dwóch silnych odstępstw. Zanotowano gwałtowny spadek wartości procesu, jednak dla liczby awarii zamiast wzrostu także nastąpił mocny spadek, który został zidentyfikowany jako odstępstwo. Była to Wigilia Bożego Narodzenia i większość pracowników udała się na urlop, stąd obie wartości (wydajności procesu biznesowego i liczby awarii) uległy spadkowi. Na tej podstawie obserwacja została wyeliminowana z dalszej analizy.

Szczegółowo przeanalizowano kolejną obserwację – z dnia 27.05.2011. Na podstawie rysunków 3 i 4 widać silny spadek wydajności procesu biznesowego i wzrost liczby awarii.



Rysunek 3. Wyniki dla okresu 1 badanego przypadku na wykresie punktowym z wyróżnionym wyraźnym odstępstwem



Rysunek 4. Wyniki dla okresu 1 badanego przypadku na wykresie złożonym. W dniu 27.05.2011 zaobserwowano wysoki wzrost liczby awarii przy spadku wydajności procesu biznesowego

W kolejnym kroku przeanalizowano najczęstsze typy awarii wygenerowane tego dnia w aplikacji wspierającej proces. Zaobserwowano, że 27.05.2011 roku występowały często problemy z siecią informatyczną (zgłoszenia typu

„timeout”, 407 wystąpień). Są to błędy ogólne, wpływające na funkcjonowanie całego systemu i wszystkich procesów realizowanych w jego zakresie. Drugi defekt, który wywołał wiele awarii, został spowodowany przez programistę (błąd w nazwie tabeli bazy danych, 255 wystąpień). Jest on bezpośrednio związany z modułem wykorzystywanym w analizowanym procesie. W konsekwencji mógł być jednym z głównych powodów spadku wydajności procesu biznesowego w danym dniu. Pracownicy nie mogli realizować wystawiania ugód klientom, co przy zaobserwowanej skali problemu mogło się przełożyć na zauważalny spadek przychodu przedsiębiorstwa.

Wnioski i podsumowanie

Celem artykułu była analiza wpływu awaryjności systemów informatycznych na procesy biznesowe realizowane przez podmiot gospodarczy. Przeprowadzono studium przypadku dla wybranych systemów i rzeczywistych danych (operacyjnych oraz logów awarii) gromadzonych w KRUK S.A.

Stwierdzono, że możliwe jest poszukiwanie i powiązanie przyczyny spadku liczby procesów zakończonych sukcesem w sytuacji nagłego zwiększenia wystąpień awarii systemu informatycznego. Przy użyciu kryterium Peirce’a i analizy wykresów opartych na znormalizowanych danych możliwa była identyfikacja wyraźnych odstępstw dwóch zbiorów danych (awarii i wydajności procesu biznesowego), co było wstępem do dalszej analizy – eksperckiej, pozwalającej na potwierdzenie lub odrzucenie zależności. Całość eksperymentu przeprowadzono z wykorzystaniem aplikacji utworzonej na jego potrzeby.

W trakcie badań napotkano dwie główne trudności i kolejne prace będą stanowiły próby ich rozwiązania – utworzenie reguł filtrowania obserwacji oraz metody wiązania logu awarii z procesem biznesowym. Reguły filtrowania zastosowane dla badania dotyczyły wyłącznie dnia wolnego od pracy oraz możliwości odrzucenia wartości określających wydajność procesu biznesowego powyżej mediany. Jednak eksperyment wykazał, że zdarzają się również dni, w których większość pracowników jest niedostępna, można je w pewien sposób wydedukować (np. dni między świętami). Dodatkowo opcja filtrowania obserwacji wykraczających poza określony poziom umożliwiłaby lepsze dopasowanie wyników. Jeszcze większe wyzwanie stanowi zautomatyzowane wiązanie logu awarii z procesem biznesowym. Pozwoliłoby to na zminimalizowanie udziału ekspertów dziedzinowych w ocenie, czy konkretny defekt mógł zmniejszyć wydajność procesu biznesowego. W tym celu wymagane byłoby opracowanie reguł klasyfikujących dane operacyjne i logi awarii lub projektu

architektury systemu w przypadku tworzenia nowych rozwiązań. Rezultatem byłby model umożliwiający weryfikację działania oprogramowania w celu podjęcia decyzji o jego zmianie lub dalszym rozwoju.

Bibliografia

- Abdi, H., 2010, *Normalizing Data*, in: *Encyclopedia of Research Design*, Sage, s. 935–938.
- Bowen, J.B., 1980, *Standard Error Classification to Support Software Reliability Assessment*, in: *Proceeding American Federation of Information Processing Societies '80*, 19–22 May, s. 697–705.
- Brosch, F., Gitzel, R., Koziolok, H., 2010, *Combining Architecture-based Software Reliability Predictions with Financial Impact Calculations*, *Electronic Notes in Theoretical Computer Science*, s. 3–17.
- Freimut, B., Denger, C., Ketterer, M., 2005, *An Industrial Case Study of Implementing and Validating Defect Classification for Process Improvement and Quality Management*, in: *11th IEEE International Software Metrics Symposium*, <https://dl.acm.org/citation.cfm?id=1092159>.
- Gitzel, R., Krug, S., Brhel, M., 2010, *Towards a Software Failure Cost Impact Model for the Customer*, in: *Proceedings of the 6th International Conference on Predictive Models in Software Engineering*, <https://dl.acm.org/citation.cfm?id=1868354>.
- Grottke, M., Graf, C., 2009, *Modeling and Predicting Software Failure Costs*, in: *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference*, s. 180–189.
- Hecht, H., Wallace, D., 1996, *Error Classification and Analysis for High Integrity Software*, The 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Technology.
- IEEE, 2010, *Standard Classification for Software Anomalies*, STD-1044–2009.
- Jain, S., Prinja, R., Chandra, A., Zhang, Z.-L., 2008, *Failure Classification and Inference in Large-Scale Systems: A Systematic Study of Failures in PlanetLab*, University of Minnesota, Minneapolis.
- Jayalakshmi, T., Santhakumaran, A., 2011, *Statistical Normalization and Back Propagation for Classification*, *International Journal of Computer Theory and Engineering*, s. 89–93.
- Leszak, M., Perry, D.E., Stoll, D., 2002, *Classification and Evaluation of Defects in a Project Retrospective*, *The Journal of Systems and Software*, no. 61, s. 173–187.
- Ross, S.M., 2003, *Peirce's Criterion for the Elimination of Suspect Experimental Data*, *Journal of Engineering Technology*, s. 38–41.
- Zhivich, M., Cunningham, R.K., 2009, *The Real Cost of Software Errors*, *IEEE Security & Privacy Magazine*, s. 87–90.