

Gabriela Idzikowska*, Zofia Owczarek**

METODY KONTROLI PROCESU PROJEKTOWANIA
I DOKUMENTACJI PROJEKTOWEJ SYSTEMÓW INFORMATYCZNYCH

1. Cechy dokumentacji projektowej

Szczególne znaczenie dokumentacji w całości prac związanych z projektowaniem i wdrażaniem systemu informatycznego wynika z powszechnego stosowania dokumentów jako nośników informacji. Jest ona niezbędna do odzwierciedlenia poszczególnych faz procesu projektowania oraz przy eksploatacji systemów i ich konserwacji.

W praktyce przyjmuje się podział procesu projektowania na następujące fazy¹:

1. prace projektowe wstępne
 - prace przygotowawcze (przeddecyzyjne),
 - analiza istniejącego systemu (przy metodzie diagnostycznej projektowania systemów),
 - projektowanie ogólne (rozszerzone założenia techniczno-ekonomiczne);
2. prace projektowe szczegółowe
 - projektowanie techniczne,
 - programowanie,

* Dr, adiunkt w Katedrze Informatyki.

** Dr, adiunkt w Katedrze Informatyki.

¹ Na ten temat można znaleźć bardziej szczegółowe informacje w literaturze dotyczącej projektowania systemów informatycznych, np. R. Jakubowski, Projektowanie dokumentacji systemu przetwarzania danych w warunkach ich automatyzacji, WNT, Warszawa 1978; E. Niedzielska [red.], Projektowanie systemów informatycznych, PWE, Warszawa 1977; H. Zygier, Metodyka projektowania systemów informatycznych, OBRI, Warszawa 1974.

- opracowywanie dokumentacji eksploatacyjnej;
- 3. wdrażanie i eksploatacja systemu
- wstępna eksploatacja i weryfikacja systemu,
- eksploatacja użytkowa.

Każda z faz procesu projektowania powinna być udokumentowana, a dokumentacja zatwierdzona przez kierownictwo użytkownika.

W historii projektowania i eksploataowania systemów informatycznych nie dopracowano się obowiązujących, uregulowanych prawnie standardów dokumentowania systemów. Istniejące "standardy" są przyjęte zwyczajowo i różnią się w zależności od ośrodków obliczeniowych, zespołów projektowych, a nawet przyjętych metodyk projektowania. Należy podkreślić, że istnienie ogólnie obowiązujących standardów, a przynajmniej standardów w ramach danego przedsiębiorstwa, ułatwiłoby komunikowanie się różnych grup użytkowników między sobą, prace kontrolne oraz prace projektowo-adaptacyjne.

Z dokumentacji korzystają różne grupy osób, a mianowicie: użytkownicy końcowi, operatorzy systemów, operator maszyny, kierownictwo użytkownika, bibliotekarz zbiorów. Każdej z tych grup osób jest potrzebny inny zakres informacji o systemie. Kompletną informacją posługuje się jedynie operator systemu.

Cechy dokumentacji systemu informatycznego są w zasadzie takie same, jak każdej dokumentacji, a więc odnoszą się do niej także pewne standardy dokumentów rachunkowych, takie jak: kompletność, zgodność ze standardami przyjętymi w danej jednostce organizacyjnej, dokumentowanie zdarzeń, sprawdzalność, prawidłowość (integralność).

Dokumentacja musi być sporządzona w formie komunikatywnej, zrozumiałej i jasnej. Większą jej czytelność uzyskuje się poprzez zastosowanie strukturalnych metod projektowania. Wpływa to na łatwość utrzymywania w stanie aktualnym dokumentacji systemu oraz jej kontrolowania. Dokumentacja powinna być zawsze aktualna; wszelkie zmiany dokonywane w systemie należy dokładnie i szybko dokumentować. Powinna być ona także rozdzielna; poszczególne jej części muszą być udostępnione wszystkim zainteresowanym osobom przed rozpoczęciem próbnej eksploatacji. Podział dokumentacji ma na celu dostarczenie poszczególnym użytkującym grupom niezbędnych informacji o systemie, a jednocześnie zapobieżenie przeciekowi informacji do nie upoważnionych osób.

Istnieją narzędzia wspomagania projektowania systemów informa-

tycznych, które jednocześnie tworzą dokumentację systemu. Uzyskiwana tą drogą dokumentacja odpowiada wszystkim wcześniej opisanym standardom, a więc np. posiada cechy kompletności, czytelności, rozdzielności. W takim pakiecie wspomagającym istnieją programy, które analizują dane o systemie, dostarczone przez zespół projektowy, m. in. z punktu widzenia wzajemnych zależności. Wszelkie błędy są sygnalizowane i muszą być poprawione. Tak więc ostateczna dokumentacja posiada również cechę integralności. Cała dokumentacja jest zapisana w pamięci zewnętrznej komputera. Jeśli jest ona bieżąco aktualizowana, w przypadku dokonywania zmian, to w każdym momencie można uzyskać wydruk jej aktualnej wersji.

2. Kontrola procesu projektowania systemu informatycznego i jego dokumentacji

Procedury kontroli obejmują jakość i integralność procesu analizy systemu, projektowania i wdrażania [2, 4]. Mają one na celu zapobieganie błędom w projektowanym systemie. Wszystkie popełniane błędy można podzielić na dwie grupy:

- błędy projektowania, czyli te, które powstają dlatego, że oprogramowanie nie spełnia wymagań użytkownika (niezrozumienie założeń systemu przez projektantów),
- błędy kodowania, czyli te, które powstają na skutek nieprawidłowego stosowania klauzul języka programowania (niedokładność w programowaniu lub rozmyślne włączanie nie autoryzowanych instrukcji).

Bezpośredni nadzór nad pracami projektowymi kontroler² ma wówczas, gdy jest członkiem zespołu projektowego. Jako członek zespołu ma ponadto możliwość dokonania rozszerzeń programów zastosowań o te moduły, które ułatwią mu w późniejszym czasie przeprowadzenie

² Pod pojęciem "kontrolera" można rozumieć zarówno osobę z zewnątrz (np. biegły księgowy), jak i pracownika kontroli wewnętrznej w jednostce gospodarczej. Do tej pory w Polsce kontrole procesu projektowania i eksploatacji systemów informatycznych nie były raczej prowadzone z udziałem niezależnego kontrolera. Natomiast na Zachodzie istnieją już w tej dziedzinie konkretne doświadczenia. Artykuł ten postuluje dokonywanie kontroli systemu informatycznego w celu podniesienia jego wiarygodności.

kontroli. Jednakże przy takim rozwiązaniu istnieje niebezpieczeństwo, że kontroler straci obiektywność konieczną do niezależnej oceny pracy zespołu projektowego. Rola kontrolera powinna więc być ograniczona do niezależnej kontroli działań związanych z opracowywaniem systemu, nie powinien on być jednak uczestnikiem tego procesu. Niezależne przeglądy dokumentacji dokonywane przez kontrolerów w trakcie procesu projektowania systemu umożliwiają w większym stopniu osiągnięcie celów kontroli; preferuje się przeglądy dokonywane w trakcie, a nie po zakończeniu procesu projektowania.

O ile uczestnictwo kontrolera w zespole projektowym jest dyskusyjne, o tyle bezdyskusyjny jest udział użytkowników (przedstawicieli zainteresowanych komórek oraz kontroli wewnętrznej) w pracach nad systemem. Daje to kontrolerowi większą gwarancję, że w projekcie systemu zostaną uwzględnione wszystkie żądania użytkowników.

Podstawę nadzoru nad pracami projektowymi stanowią sprawozdania ze spotkań zespołu projektującego, zawierające listę uczestników, datę spotkania oraz ustalenia dotyczące opracowywanego systemu.

Po zakończeniu prac projektowych osoba kontrolująca powinna sprawdzić i ocenić:

- udział użytkowników w procesie projektowania systemu,
- dane wynikowe, uzyskane podczas próbnej realizacji programów,
- sposób i wyniki testowania systemu,
- dokumentację, potwierdzającą przyjęcie projektu systemu przez kierownictwo jednostki gospodarczej.

Elementy projektu powinny być przedyskutowane z użytkownikami systemu w celu zorientowania się, czy rozumieją oni jego istotę oraz sposób wykorzystania.

Kontroler powinien przejrzeć dokładnie całą dostępną dokumentację użytkownika odnoszącą się do procesu testowania, aby upewnić się, że wszystkie procedury programowe mające wpływ na dane rzeczywiste zostały przetestowane [2, 1]. Kontrole procesu testowania obejmują:

- cele testowania,
- przeglądanie danych testowych,
- wyniki testu.

Jeśli kontrola ujawni braki w dokumentacji, niezgodność z założeniami lub inne błędy, osoba badająca powinna dokonać testowania

we własnym zakresie przed ostatecznym zatwierdzeniem systemu przez kierownictwo. Do dokładniejszego badania programów może użyć jednej z metod opisanych w dalszej części artykułu.

3. Metody kontroli dokumentacji po zmianach

Problemy [2], które mogą powstać w wyniku zmian w programach, są podobne do tych, które występują w procesie projektowania systemu. Po pierwsze, mogą być nieumyślnie wprowadzone błędy, wynikające z niezrozumienia założeń do programów lub z nieuważnego programowania. Po drugie, mogą być rozmyślnie wprowadzone do programów nie autoryzowane instrukcje. Te problemy pojawiają się nie tylko w odniesieniu do programów użytkowych, ale także do oprogramowania systemowego.

Aby mieć pewność, że do programów zostały wprowadzone jedynie autoryzowane zmiany, wszelkie modyfikacje powinny być udokumentowane i zatwierdzone, przy jednoczesnym przestrzeganiu zasady rozdzielania przynajmniej następujących funkcji: programowania, obsługi komputera i bibliotekarza nośników magnetycznych. Brak rozdzielania tych funkcji (a więc ich spełnianie przez jedną osobę) mogłoby ułatwić wprowadzanie nie autoryzowanych zmian do programów.

Kontroler, który ma zbadać zmiany w zmodyfikowanych programach powinien rozpocząć od dokonania kontroli ogólnej polegającej na:

- przejrzaniu schematów organizacyjnych i opisów zadań wykonywanych na poszczególnych stanowiskach (sprawdzenie rozdzielania funkcji),
- sprawdzeniu, czy występują zabezpieczenia odnoszące się do kontroli dostępu do sprzętu komputerowego, zbiorów programów i dokumentacji programów,
- przejrzaniu zapisów w dzienniku utrzymywanym przez bibliotekarza nośników magnetycznych.

Zmodyfikowane (bądź nowe) programy, zwłaszcza te, które wzbudziły zastrzeżenia osoby badającej podczas dokonywania ogólnej kontroli, powinny być sprawdzone szczegółowo przy użyciu jednej z następujących metod:

- przeglądania wersji źródłowej programu,
- porównywania programu z wersją atestowaną,

- śledzenia programu,
- powtórnego przetwarzania i symulacji równoległej.

3.1. Metoda przeglądania wersji źródłowej programu [4]

Przeglądanie polega na czytaniu wydruku wersji źródłowej programu w celu określenia, czy istnieją nie autoryzowane instrukcje oraz wygenerowania hipotezy o potencjalnych błędach w logice. Przeglądanie może dotyczyć części lub całości programu. Dzięki takiemu przeglądaniu można zidentyfikować instrukcje:

- błędne - wynikające z nieuważnego programowania lub nieprzestrzegania założeń do programu,
- nie autoryzowane - wprowadzone nielegalnie,
- nieskuteczne - nie osiągające celów programu,
- nieefektywne - umieszczone w niewłaściwym porządku, co powoduje niepotrzebne wielokrotne ich wykonywanie.

Aby przetestować hipotezy dotyczące błędów w logice kontroler wykorzystuje dane testowe. Błędy znalezione w logice programu muszą być poprawione.

Przeglądanie wersji źródłowej programu realizuje się w następujących krokach:

- przeglądanie standardów programowania w ośrodku obliczeniowym (np. obowiązek włączania komentarzy do programu),
- zapoznanie się z założeniami do programu,
- uzyskanie wersji źródłowej (osoba badająca powinna upewnić się, że otrzymała wydruk aktualnie wykorzystywanej wersji programu),
- przeglądanie kompilatora języka,
- przeglądanie wersji źródłowej programu,
- formułowanie wniosków.

Nie ma obecnie ani teorii, ani doświadczeń wskazujących najlepszy sposób przeglądania programu źródłowego. Wiele pytań pozostaje bez odpowiedzi:

- czy pewne metody przeglądania identyfikują więcej błędów niż inne,
- czy pewne metody przeglądania są szybsze niż inne,
- czy metoda przeglądania jest zależna od tego, jak program został napisany (np. strukturalnie),
- czy metoda jest zależna od celu przeglądania,

- czy efektywność i skuteczność przeglądania zależą od psychologicznych cech osoby badającej.

Pomocą dla osoby kontrolującej w zastosowaniu tej metody mogą być pakiety programowe, które [2, 4]:

- automatycznie tworzą schemat blokowy programu; program taki interpretuje wersję źródłową i generuje schemat blokowy;

- automatycznie tworzą tablice decyzyjne; program generuje tablicę decyzyjną, reprezentującą logikę opisaną w wersji źródłowej programu;

- zawierają procedury przeszukujące, które badają, czy w programie występują określone nazwy zmiennych lub inne kombinacje znaków oraz drukują te nazwy, a także miejsca i sposoby ich wykorzystania w programie ("procedura drukowania odsyłaczy");

- zawierają procedury odwzorowania, które można uaktywnić podczas normalnego przetwarzania; umożliwiają one identyfikację tych części programu użytkowego, które nie były wykonywane (badanie efektywności programu).

Główną zaletą opisywanej metody jest to, że dostarcza ona możliwości dokładnego zapoznania się z programem, co jest trudne do osiągnięcia przy użyciu innych metod. Jej wadą jest wysoki koszt wynikający z dużej czasochłonności.

3.2. Metoda porównywania programu z wersją atestowaną [4]

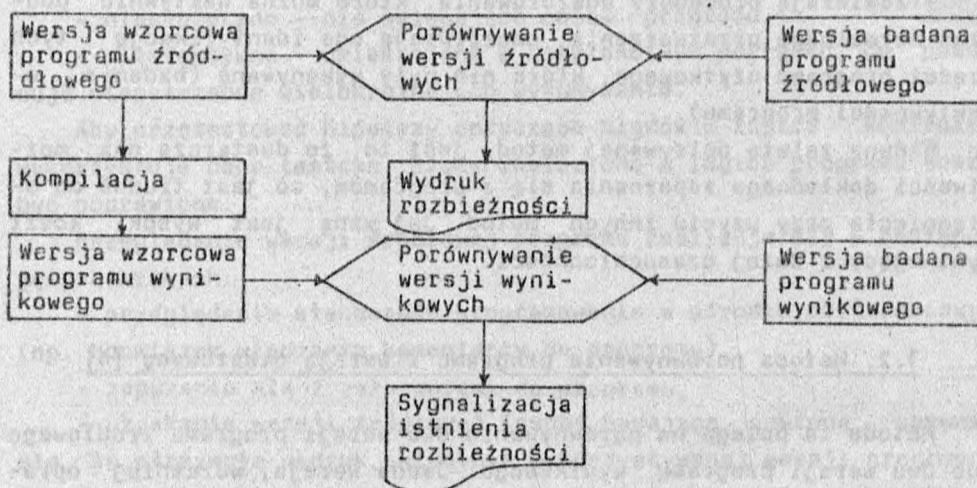
Metoda ta polega na porównywaniu dwu wersji programu źródłowego lub dwu wersji programu wynikowego. Jedna wersja, wcześniej opracowana, posiadająca znane kontrolerowi atrybuty, jest wersją atestowaną. Przez porównanie można określić, czy inna wersja posiada takie same atrybuty, a tym samym można ustalić istnienie rozbieżności między obiema wersjami.

Można dokonywać dwóch typów porównań:

- porównania wersji źródłowych,
- porównania wersji wynikowych.

Czynności porównania są zwykle wykonywane przy użyciu specjalnego software'u. Ów software, w wyniku porównywania wersji źródłowych, dostarcza wydruku różnic między tymi wersjami. Na podstawie wydruku następuje badanie natury i przyczyny wystąpienia różnic. Na-

to miast przy porównaniu wersji wynikowych program udziela jedynie odpowiedzi na pytanie, czy występują różnice. Jeżeli różnice nie występują, oznacza to, że kontrolowana wersja programu jest właściwa. W przypadku wystąpienia różnic należy, przede wszystkim, upewnić się, czy nie wynikają one z faktu użycia różnych kompilatorów do uzyskania porównywanych wersji wynikowych (producenci oprogramowania systemowego często modyfikują kompilatory i dostarczają użytkownikom nowych wersji). Jeżeli różnice nie wynikają z użycia różnych kompilatorów, należy zastosować inne metody, które umożliwią ustalenie przyczyn tych różnic (np. porównanie wersji źródłowych). Porównywanie przebiega przez fazy przedstawione na rys. 1.

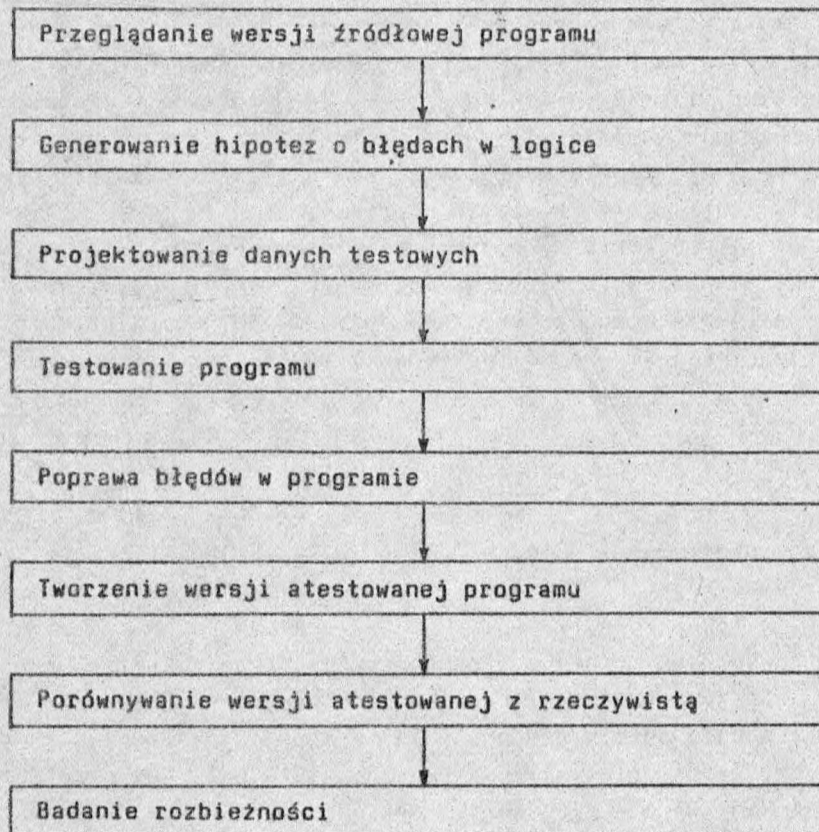


Rys. 1. Całościowe podejście do wykorzystania metody porównywania programów

Źródło: [4, s. 467]

Porównanie wersji źródłowych lub wynikowych programów jest najbardziej efektywne, gdy jest stosowane w powiązaniu z innymi metodami. Powiązanie metod przedstawia rys. 2.

Metoda porównywania programów pozwala w łatwy sposób stwierdzić, czy zostały dokonane zmiany w programach. Eksploatacja oprogramowania dokonującego porównań nie jest kosztowna.



Rys. 2. Zintegrowane użycie różnych metod badania programów
Źródło: [4, s. 449]

3.3 Metoda śledzenia programu [2, 3]

Śledzenie programu (program tracing) jest metodą, która umożliwia kontrolerowi uzyskanie dokładnej wiedzy o logice programu użytkowego, jak również sprawdzenie zgodności programu z założeniami kontroli. Aby wykorzystać tę metodę przetwarza się program użytkowy z transakcjami rzeczywistymi lub hipotetycznymi, aktywizując przy tym procedurę śledzenia wbudowaną w oprogramowanie systemowe. Procedura ta powoduje, że komputer drukuje w porządku sekwencyjnym li-

stę wszystkich kroków programu zastosowań (numery linii i nazwy instrukcji) wykonanych w czasie przebiegu programu łącznie z danymi, które stanowiły wynik działania poszczególnych instrukcji. Kontroler może więc obserwować dokładną sekwencję zdarzeń, które miały miejsce w czasie realizacji programu. Następnie porównuje on wersję źródłową programu z otrzymaną listą, aby potwierdzić swoje oczekiwania dotyczące logiki działania programu.

Metoda ta umożliwia wykrycie nie autoryzowanych instrukcji programowych, niepoprawnych ścieżek logicznych lub nie używanych sekwencji instrukcji wewnątrz programu użytkowego. Po ujawnieniu takich przypadków jest konieczne zbadanie ich przyczyn i skutków.

3.4. Metody powtórnego przetwarzania i symulacji równoległej [2]

Dwie dodatkowe metody, które wykorzystuje się do wykrywania nie autoryzowanych zmian w programach, są ponowne przetwarzanie i symulacja równoległa.

Aby zastosować metodę powtórnego przetwarzania kontroler musi uzyskać atestowaną wersję programu źródłowego, a następnie przechować ją w celu późniejszego użycia. W dowolnych odstępach czasu stosuje się ją do przetwarzania danych, które były już przetworzone przez aktualną wersję programu, a następnie porównuje dane wynikowe uzyskane w obu przebiegach. Jeśli wystąpiły rozbieżności w danych wynikowych, należy wyjaśnić ich przyczyny.

Symulacja równoległa działa w podobny sposób z tą różnicą, że zamiast używania wersji atestowanej programu, kontroler posługuje się własnym programem, który jest wykorzystywany do ponownego przetwarzania danych przetworzonych wcześniej przez program badany.

Literatura

- [1] B o d n a r G. H., Accounting Information Systems, Allyn and Bacon, Boston 1983.
- [2] C u s h i n g B. E., Accounting Information Systems and Business Organizations Reading, Addison-Wesley Publ. Comp., Mass. 1982.

- [3] Hoffman L. J., Poufność w systemach informatycznych, WNT, Warszawa 1982.
- [4] Weber R., EDP Auditing, McGraw-Hill, N. Y. 1982.

Gabriela Idzikowska, Zofia Owczarek

METHODS OF CONTROL OF THE PROCESS OF DESIGNING
AND DESIGN DOCUMENTATION OF INFORMATION SYSTEMS

The article deals with problems concerning the control of the process of designing information systems and of the design documentation connected with it. Such controls aim at increasing the reliability of information systems. There have been described, among others, the following methods:

- source code review,
- comparison of a program with an attested version,
- program tracing,
- reprocessing,
- parallel simulation.