Jaki jest sens atestacji systemów komputerowych?

mgr inż. Zdzisław Żurakowski
Instytut Automatyki Systemów Energetycznych
we Wrocławiu

Na tle różnic technologii wytwarzania oprogramowania i doświadczeń związanych z eksploatacją systemów komputerowych w Polsce i na Zachodzie, przedstawiono cele atestacji według podejścia tradycyjnego oraz spojrzenie na atestację w kontekście, zaproponowanej przez Lewisa Mumforda, koncepcji megamaszyny jako narzędzia analizy rozwoju cywilizacji. Zasadniczym celem artykułu jest odpowiedź na pytanie: jeśli w Polsce brak doświadczeń związanych z ujawnianiem się błędów w oprogramowaniu mimo niestosowania atestacji, to czy atestacja jest potrzebna i jaki by miał być jej sens?

Co miesiąc Software Engineering Notes podaje całe strony przypadków, w których złe funkcjonowanie systemów komputerowych czasu rzeczywistego postawiło społeczeństwo lub środowisko w stan zagrożenia [1]. W analizie wypadków związanych z komputerami, opisywanych w Software Engineering Notes oraz opisów wypadków zebranych przez UK Health and Safety Executive (HSE) całkowitą liczbę ofiar śmiertelnych wypadków związanych z komputerami w skali światowej do końca roku 1992 oszacowano na około 2000 [2]. Zwrócono uwagę, że chociaż jest to liczba stosunkowo mała np. w porównaniu z 4000 ofiar śmiertelnych rocznie na drogach Zjednoczonego Królestwa, nie jest to powód do zadowolenia, gdyż wypadki śmiertelne czasem stanowią tylko wierzchołek „góry lodowej", na którą składają się poważne i lekkie zranienia oraz sytuacje bliskie wypadkom. Informacje na temat wypadków ujawniane są niechętnie i można przypuszczać, że nie wszystkie są podawane do wiadomości publicznej, a jeśli są podawane, dostępna dokumentacja wypadku jest często na poziomie zbyt niskim, aby możliwe było dokładne i pewne ustalenie przyczyn. Z analizy wynika, że głównymi przyczynami wypadków śmiertelnych były: przyczyny fizyczne, głównie zakłócenia elektromagnetyczne (4 %), defekty oprogramowania (3 %), błędy we współdziałaniu operatora z systemem komputerowym (92 %). Na Zachodzie informacje o zagrożeniach związanych z błędami w funkcjonowaniu systemów komputerowych czasu rzeczywistego są publikowane od końca lat 60. i zaangażowanie w prace naukowe oraz działalność, związaną z wyeliminowaniem lub zmniejszeniem ryzyka związanego z występowaniem tych błędów przez ocenę i atestację systemów przed ich przekazaniem do eksploatacji, jest bardzo duże. Mimo to z punktu widzenia naukowego atestacja jest wciąż zagadnieniem nierozwiązanym i w dalszym ciągu stanowi poważny problem we wdrażaniu systemów komputerowych do zastosowań związanych z bezpieczeństwem.

W Polsce dotychczas nie stosuje się atestacji systemów komputerowych. W związku z realizacją przez IASE projektu ISAT*), który obejmował m.in. analizę bezpieczeństwa wybranych funkcji komputerowego systemu sterującego pracą stacji najwyższych napięć, stwierdzono [4]:

- co najmniej niską świadomość zagrożeń związanych z ewentualnym błędem funkcjonowania systemu komputerowego;
- niedostosowanie przepisów do zastosowania systemów komputerowych w ww. stacjach;
- stosowanie nieodpowiednich metod oceny systemów komputerowych przed ich przekazaniem do eksploatacji.

Przykładem niedostosowania przepisów jest instrukcja [5], która w punkcie 3.1.5 mówi: *„Po każdej manipulacji odłącznikiem, niezależnie od prawidłowości wskazań układów sterowania, należy stwierdzić poprzez oględziny rzeczywisty stan noży"*. Instrukcja nie jest dostosowana do sekwencji łączeniowych wykonywanych automatycznie przez komputer, podczas których może następować sterowanie wieloma odłącznikami, inicjowane samoczynnie przez komputer. Żadnych innych wytycznych w tym względzie nie ma. Również stosowana procedura przekazywania do eksploatacji systemów komputerowych, na podstawie wymagania ciągłego funkcjonowania systemu bez wystąpienia błędu przez 72 godziny, jest niczym nieuzasadniona. Rozumiana jest ona w ten sposób, że jeśli w ciągu 72 godzin wystąpi błąd, to należy go usunąć i kontynuować pracę, a jeśli błąd powtórzy się, to należy stosować takie postępowanie dopóty, dopóki nie osiągnie się 72 godzin ciągłej pracy bez wystąpienia błędu. Wiadomo, że **błąd może wystąpić nawet po kilku latach bezbłędnej pracy systemu**. Należy podkreślić, że mimo istnienia przedstawionej pokrótce sytuacji, w przypadku systemów programowalnych stosowanych w sektorze energetyki od przeszło 20 lat, w tym w zakresie funkcji odpowiedzialnych, nie notowano nie tylko żadnego wypadku, ale nawet – o ile autorowi wiadomo – żadnej poważniejszej awarii spowodowanej systemem komputerowym. Ta ostatnia uwaga odnosi się do zastosowania komputerów w Polsce w ogóle, nie tylko w sektorze energetyki. I chociaż zastosowań komputerów, zwłaszcza związanych z bezpieczeństwem, jest w Polsce obecnie nieporównanie mniej niż na Zachodzie, to nie ulega wątpliwości, że sytuacja w naszym obszarze cywilizacyjnym jest zaskakująco różna.

Ponieważ, mimo bardzo dużych nakładów na ocenę i atestację systemów programowalnych, sytuacja na Zachodzie jest gorsza niż u nas, pojawiają się pytania: jaki jest sens atestacji i czy warto w Polsce angażować się w to bardzo trudne i kosztowne zagadnienie? Wydaje się, że te pytania zaczynają wykraczać poza Polskę. Tak bowiem należy rozumieć referat [6], zatytułowany *Certyfikacja oprogramowania: czy jest konieczna?* W Stanach Zjednoczonych normalizacja i atestacja w dziedzinie oprogramowania zaczęła się i ma chyba najsilniejsze podstawy.

*) Niniejszy artykuł został opracowany na podstawie raportu „Does Computers Systems Validation and Certification Make Sens?" nr TR ISAT 97/13, opracowanego w ramach projektu Unii Europejskiej EU Joint Research Project „Copernicus" CP'94 1594 Integration of Safety Analysis Techniques for Process Control Systems (ISAT).

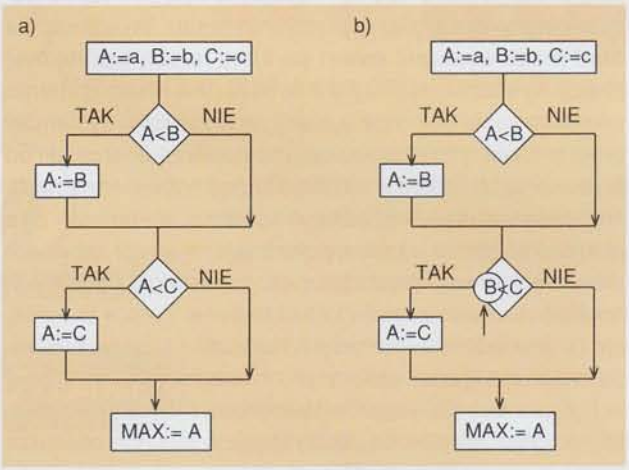
Dlaczego ocena bezpieczeństwa systemu komputerowego stanowi problem?

Pod pojęciem **system bezpieczny** zwykle rozumie się system (sprzęt i oprogramowanie) tak zaprojektowany, że pojawienie się defektu w nim, (np. w oprogramowaniu), nie spowoduje fizycznego zranienia, śmierci, poważniejszych zniszczeń lub zakłóceń w życiu społecznym. Nie chodzi więc o sam system, który jest na ogół bezpieczny, lecz o skutki, jakie może spowodować urządzenie lub obiekt, w który system ten jest wbudowany funkcjonalnie. W tym sensie używany jest również termin **bezpieczne oprogramowanie** [3].

Problemy bezpieczeństwa wiążą się ściśle z teorią niezawodności systemów, z której się wyłoniły. Różnica między niezawodnością a bezpieczeństwem tkwi w rozłożeniu akcentów. O ile niezawodność jest miarą poprawnej pracy systemu w założonych warunkach i w przyjętym przedziale czasu, czyli wymaga położenia akcentu na wyeliminowanie błędów w działaniu w ogóle, o tyle bezpieczeństwo wymaga położenia akcentu na wyeliminowanie tylko tych błędów, które mogą prowadzić do wystąpienia podanych wyżej strat, uznanych za nieakceptowalne.

W przypadku konwencjonalnych, przekaźnikowych układów sterowania zachowanie się systemu może być stosunkowo łatwo określone na drodze analizy układu i sprawdzone podczas badań, czy układ zachowuje się zgodnie z oczekiwaniem.

Zupełnie inna sytuacja jest w przypadku systemów komputerowych, czy ogólniej programowalnych. Funkcjonowanie systemu jest określone przez oprogramowanie, które zawiera co najmniej kilka lub kilkanaście tysięcy linii kodu. Praktyka w krajach zachodnich dowiodła, że nie jest możliwe wytworzenie tak dużego programu bez defektu. Defekty w oprogramowaniu, które mogą powstać we wszystkich etapach tworzenia oprogramowania, mogą ujawnić się – jak dowodzi praktyka – nawet po kilku latach prawidłowej pracy systemu.



Algorytm wyznaczania maksimum z trzech dowolnych liczb rzeczywistych a, b, c; a) – poprawny, b) – z popełnionym błędem (błąd zaznaczono strzałką i kółkiem) [7]

Przedstawiony na rysunku algorytm jest prostym przykładem takiego przypadku, w którym defekt w oprogramowaniu daje niepoprawne wyniki tylko dla określonych danych wejściowych. W podanym przykładzie ma to miejsce wówczas, gdy dane wejściowe spełniają relację $b < c < a$. Łatwo sprawdzić, że w każdym innym przypadku program, mimo defektu, będzie działał poprawnie.

Oprogramowanie ma następujące cechy, różniące je od konwencjonalnych rozwiązań nieprogramowalnych:

- brak postaci fizycznej i nie podleganie zużyciu w czasie użytkowania;
- brak możliwości opisania oprogramowania za pomocą funkcji ciągłej oraz nieistnienie żadnego prawa fizycznego, które pozwoliłoby na badanie składników oprogramowania, np. przez założenie liniowego zachowania się. Defekty występują w sposób przypadkowy – takie pojęcia jak: *naprężenie* czy *współczynnik bezpieczeństwa* w przypadku oprogramowania nie mają zastosowania;
- niezwykłe dla rozwiązań nieprogramowalnych wzmocnienie skutków małych zmian. Zmiana pojedynczego bitu informacji (w programie lub danych) może mieć skutki katastrofalne;
- występowanie dużej liczby stanów dyskretnych bez powtarzalnej struktury, co czyni niemożliwą analizę i testowanie wszystkich możliwych przypadków.

Cechy te są przyczyną istniejących trudności w ocenie bezpieczeństwa systemów programowalnych. Znane stwierdzenie Dijkstry, że można wykazać istnienie błędu, ale nie można wykazać, że go nie ma, stwarza zupełnie nową sytuację w zakresie oceny bezpieczeństwa takich systemów.

Cele atestacji w podejściu tradycyjnym

Interwencja państwa w aktywność człowieka w celu ochrony zdrowia mieszkańców, sięga prehistorii. Ruiny w Dolinie Indus dowodzą, że już w 400 r. p.n.e. wdrożono przepisy budowlane oraz inżynierię sanitarną [8].

W odniesieniu do systemów komputerowych podano [6], że certyfikacja jest niezbędna w przypadku oprogramowania krytycznego ze względu na bezpieczeństwo oraz wagę spełnianych funkcji. Przy definicjach bezpieczeństwa, uwzględniających nie tylko bezpośrednie zagrożenie dla życia i zdrowia, ale i nieakceptowalne ryzyko innych szkód (np. chaosu, do którego może prowadzić błąd w systemach rezerwacji miejsc), te dwie kategorie oprogramowania mogą się w znacznym stopniu lub nawet całkowicie pokrywać [3].

Zwraca się uwagę [6], że niezbędna jest certyfikacja **procesu** tworzenia oprogramowania i certyfikacja **produktu**. Bezpośrednim celem certyfikacji jest danie pewności użytkownikowi, że otrzyma oprogramowanie o odpowiedniej jakości, tzn.:

- spełniające wymagania funkcjonalne,
- łatwe w użytkowaniu,
- bezpieczne,
- o wymaganej niezawodności,
- umożliwiające rozbudowę,
- zapewniające przenośność,
- spełniające wymagania odpowiednich norm itd.

Nie są to cele błahe, jeśli się zważy, że od czasu zidentyfikowania na Zachodzie w latach 60. zjawiska zwanego „kryzysem oprogramowania" [9], we wszystkich publikacjach poświęconych wytwarzaniu oprogramowania podkreśla się niską jakość oprogramowania oraz dużą nieefektywność procesu jego wytwarzania. W 1991 r. podano, że w Wielkiej Brytanii z sum, które przemysł, rząd i handel przeznaczył na oprogramowanie, co najmniej 2 mld funtów rocznie było marnowanych wskutek złej jakości. Według [10], we Francji roczny koszt awarii komputerowych spowodowanych defektami przypadkowymi i zamierzonymi (accidental and intentional faults)

przekraczał 10 mld franków rocznie, tzn. przekraczał dochód całego przemysłu zastosowania komputerów we Francji, włączając w to budowę, dystrybucję i serwis.

Jeśli chodzi o defekty w oprogramowaniu, to z badań przeprowadzonych na Zachodzie w 1986 r. wynika, iż typowo, w dużych systemach, na każdy milion linii kodu przypadało 20 tys. defektów. Normalnie 90 % z nich było wykrywanych podczas testowania, dalsze 200 uzewnętrzniało się w 1. roku eksploatacji, 1800 defektów pozostawało nie wykrytych [1].

Oprócz wymienionych wyżej, zasadniczych celów certyfikacji, w [6] podano następujące dodatkowe korzyści dla firm, które stosują certyfikację:

- zorganizowany (przez wymogi certyfikacji) proces zapewnienia jakości;
- przewaga w konkurencji z firmami, które nie stosują certyfikacji;
- wzrost udziału w handlu wskutek wzrostu zaufania klientów;
- niższe koszty, krótsze cykle czasowe, lepszy produkt;
- postawienie zysku w centrum działalności;
- zwiększenie zaufania;
- wyeliminowanie wielokrotnych kontroli dokonywanych przez klienta;
- stosowanie mechanizmu ciągłego podnoszenia jakości;
- przełamanie barier wydziałowych (certyfikacja musi oceniać jakość końcowego produktu, nie tylko elementów składowych);
- wzrost świadomości w zakresie zagadnień związanych z jakością.

W prezentacji projektu FRESCO [11], w którym uczestniczyły takie organizacje jak: Lloyds Register of Shipping, Railtrack, Civil Aviation Authority, podano niżej wymienione potencjalne korzyści z oceny i certyfikacji systemów komputerowych pełniących funkcje związane z bezpieczeństwem.

1. Dla korzystających z usług zapewnianych przez te systemy:

- zmniejszone składki ubezpieczeniowej,
- obrona przed odpowiedzialnością,
- zmniejszenie nieporozumień kontraktowych,
- zmniejszenie kosztów związanych z atestacją systemu przez użytkownika,
- korzyści związane z przyciąganiem odbiorców do produktów certyfikowanych,
- korzyści związane z ulepszeniem procesu projektowania;

2. Dla społeczeństwa:

- wzrost kultury bezpieczeństwa,
- wzrost świadomości,
- utrzymanie w równowadze rynku.

Jest oczywiste, że również wszystkie korzyści związane z wprowadzeniem norm jakościowych są możliwe do uzyskania tylko wówczas, gdy będzie istniał system certyfikacji oceniający i potwierdzający zgodność oprogramowania z daną normą.

Do celów certyfikacji w podejściu tradycyjnym zaliczyć należy także dążenie do uzyskania korzyści, jakie można zapewnić w przypadku zjawisk przedstawionych w [12], ściśle związanych z systemami programowalnymi. Zwrócono uwagę, że w przyszłości większość komputerów będzie tak niewidoczna, jak dzisiaj większość silników elektrycznych, a *„Programiści nawet teraz piszą coraz więcej „pakietów", których funkcją jest integracja i organizacja zachowania innych „pakietów", których oni nie tworzyli i których autorów nie znają*". Podano również, że duże systemy (np. sterowania pozyskiwaniem energii z reakcji nuklearnych, jak na Three Mile Island) na ogół nie są projektowane w tradycyjnym znaczeniu tego pojęcia. Zaczyna się nawet od projektu w tradycyjnym pojęciu, który może być nawet zrealizowany, lecz wkrótce rozpoczyna się stały proces modyfikacji i rozrostu zarówno funkcji sterowania jak i baz danych, który prowadzi do zasadniczych zmian systemu. Typowym jest, że ten rodzaj „chirurgii" dokonywany jest nie przez programistów, architektów systemu, inżynierów czy kogokolwiek, kto był zaangażowany w projekt od samego początku, lecz przez ludzi, którzy przychodzą i odchodzą do innych zadań. W rezultacie nie ma osób lub zespołów ludzi, którzy rozumieliby te duże systemy, dla których mają wykonać jakąś pracę. Współczesne duże systemy po prostu nie mają swoich autorów, rozwijają się w sposób ewolucyjny. Bardzo interesujący referat [12] kończy następujące stwierdzenie: *„Prof. Minsky dawno temu zauważył, że programiści nie powinni przypisywać sobie zasług za wybitne rzeczy, które ich programy mogą czynić, ani też brać odpowiedzialności za działanie tych programów i jego skutki. Dlatego, że po prostu nie mogą przewidzieć zachowania się organizmów, które wypuszczają w świat. I mówimy tu nie o setkach, tysiącach czy nawet milionach. Jeśli wy, Panie i Panowie siedzący tutaj, nie możecie wziąć i nie weźmiecie odpowiedzialności za to, co robicie i zamierzacie zrobić, to kto i jaka władza waszym zdaniem powinny być obciążone tą odpowiedzialnością?"*

O ile dla obydwu z przytoczonych wyżej zjawisk, charakterystycznych dla rozwoju systemów programowalnych, certyfikacja bez wątpienia umożliwiłaby poprawę jakości oprogramowania, w tym zwiększenie bezpieczeństwa (m.in. przez poprawę procesu projektowania, lepszą dokumentację itp.), to oczywiście drugie z przytoczonych zjawisk, wiąże się z procesami społecznymi, w związku z czym wpływ na poprawę jakości przez certyfikację byłby chyba mniejszy i trudniejszy do uzyskania.

Na podstawie tego, co przedstawiono wyżej można wyciągnąć wniosek, że **na gruncie dotychczasowego rozwoju cywilizacji przemysłowej, a nawet cywilizacji w ogóle oraz na gruncie nauk technicznych** (takich jak inżynieria oprogramowania) cele atestacji są oczywiste, dobrze znane, niepodważalne oraz potwierdzone przez praktykę i **pytanie zawarte w tytule nie ma sensu**.

I tak by należało traktować tę sytuację, gdyby nie wspomniana poprzednio zasadnicza rozbieżność praktyki i doświadczeń w Polsce i w krajach Zachodu, która podważa ten wniosek, przynajmniej w oczach znacznej części kadry kierowniczej i technicznej w Polsce, związanej z tego typu systemami, ze względu na oczywisty w praktyce przemysłowej wzgląd na efekty ekonomiczne i unikanie działań zbędnych.

Atestacja w kontekście mumfordowskiej megamaszyny

Głównym bodźcem do spojrzenia na celowość atestacji w szerszym kontekście społecznym i cywilizacyjnym była konieczność przedstawienia istniejącego stanu praktyki w zakresie oceny, atestacji i certyfikowania oprogramowania w Polsce, związana z realizacją jednego z zadań IASE w projekcie Europejskim ISAT.

Fakt, że w Polsce bez stosowania metod inżynierii oprogramowania, norm, uznanych na Zachodzie dobrych praktyk

w zakresie tworzenia oprogramowania oraz bez atestacji oprogramowania uzyskuje się produkt, który jest bezpieczniejszy i nie stwarza problemów z błędami, jest nie do wyjaśnienia na gruncie nauk technicznych. Dodatkowym bodźcem był fakt, że realizowany projekt, mający na celu rozwój metod analizy bezpieczeństwa stosowanej w procesie projektowania oprogramowania, nie budził w Polsce zainteresowania i nawet w zespole realizującym ten projekt autor dostrzegał wątpliwości, czy realizacja projektu, a tym bardziej wdrożenie w Polsce metod rozwijanych w jego ramach ma sens, jeśli jest dobrze tak jak jest. Jeśli się zważy całą złożoność i kosztowność istniejących procedur oceny oprogramowania jest oczywiste, że decyzja o atestacji i certyfikacji, którą miałyby podjąć władze państwowe (czy dyrekcje firm) nie jest łatwa i mogłaby być bardzo brzemienna w konsekwencje.

Punktem wyjścia do przedstawionego niżej spojrzenia na atestację w szerszym kontekście społecznym i cywilizacyjnym był cytowany wyżej ciekawy i ważny referat [12] oraz podane w bibliografii tego referatu prace Lewisa Mumforda.

Przedstawienie koncepcji mumfordowskiej megamaszyny wykracza poza ramy niniejszego artykułu. Można jedynie zauważyć, że opis i zrozumienie pewnych zjawisk społecznych są trudne przede wszystkim ze względu na brak odpowiednich pojęć. Jak wiadomo, rzeczywistość można opisać na wiele sposobów nie pozostających ze sobą w sprzeczności. Nauka sama nie opisuje wszystkiego. Opis świata, jaki znamy, jest w dużym stopniu metaforyczny. Taką metaforą do pewnego stopnia jest również model megamaszyny przedstawiony przez Lewisa Mumforda i bardzo ogólnie można powiedzieć, że autor używa terminu „maszyna" czy „machina" w znaczeniu takim, w jakim używany jest ten termin w zestawieniu: machina polityczna, machina wojenna itp. [13, 14].

Maszyna ludzka, jak każda maszyna, ma swoją sprawność rozumianą jako stosunek nakładów do uzyskanego efektu. Ofiary związane z funkcjonowaniem cywilizacji obniżają tę sprawność. W tym kontekście rozstrzygnięcie sensu atestacji systemów komputerowych z punktu widzenia „ekonomiki" rozwoju cywilizacji jest zapewne możliwe.

Wspomniana wyżej, zasadniczo różna sytuacja w Polsce w stosunku do krajów zachodnich, jeśli idzie o problemy z ujawnianiem się błędów w oprogramowaniu przy często znacznie mniejszych zasobach ludzkich zaangażowanych w wytwarzanie oprogramowania i krótszych czasach trwania projektów w Polsce, wydaje się wskazywać na nieco różny tryb funkcjonowania cywilizacji. Trudno jest rozdzielić koszty rozwoju cywilizacji na związane z wytwarzaniem oprogramowania i inne, ale zapewne jest możliwe porównanie ogólnych kosztów związanych z funkcjonowaniem tych dwóch trybów funkcjonowania cywilizacji. Jeśli okazałoby się, że koszty u nas są wyższe, oznaczałoby to, że sens atestacji zawiera się m.in. w przyczynianiu się do większej sprawności rozwoju społecznego, czy społecznej machiny wytwarzania.

Dotychczas wszelkie innowacje nie były oceniane w odniesieniu do wszystkich kosztów jakie społeczeństwo ponosi, rozważało się efekty ekonomiczne i ewentualnie wpływ na środowisko. Nie ma maszyn działających bez strat i także w przypadku maszyny ludzkiej chyba nie jest możliwy rozwój bez strat. We współczesnym społeczeństwie straty te są jednak tak niewyobrażalnie wielkie, że wzrost sprawności tej maszyny o pojedyncze procenty byłby czymś bardzo znaczącym.

Konkluzje

1. Klasyczne efekty, związane z atestacją oprogramowania i całych systemów komputerowych są bardzo duże i tak istotne, że ocena i atestacja musi stanowić niezbywalne ogniwo każdego procesu wytwarzania.

2. Do ustalenia są efekty atestacji związane z funkcjonowaniem społecznej maszyny wytwarzania rozważanych systemów, nawet jeśli nie występują straty związane bezpośrednio z ujawnianiem się błędów w ich oprogramowaniu.

3. Niezależnie od ustaleń, czy stosowanie atestacji zwiększa czy zmniejsza sprawność społecznej maszyny, należy mieć na względzie, że ostatecznym celem rozwoju cywilizacji jest rozwój i podmiotowość człowieka, które nie są możliwe bez odpowiedzialnego stosunku człowieka do wytworów swojej działalności.

4. Z konkluzją 3 wiąże się konieczność odpowiedzi na następujące pytanie: czy powinniśmy wytwarzać i wypuszczać w świat systemy, których jakości (w tym bezpieczeństwa) nie potrafimy ocenić i w związku z tym w pełni zagwarantować? Odpowiedź na to pytanie może być czasami złożona, ale jest bardzo ważna.

Bibliografia:

[1] Burns A., Wellings A.: Real-time systems and their programming languages. Addison-Wesley Publishing Company, 1990.

[2] MacKenzie D.: Computer - related accidental death: an empirical exploration. Sience and Public Policy, vol. 21, nr 4, August 1994.

[3] Żurakowski Z.: Systemy komputerowe w zastosowaniach związanych z bezpieczeństwem, Informatyka, nr 3, 1995.

[4] Żurakowski Z., Kwaśnicka H.: Dependability Issues Associated with Application of Computer - Based Systems in EHV Substations, Proc. of the International Symposium on Modern Electric Power Systems (MEPS'96), Wrocław, September 26-27, 1996.

[5] Wspólnota Energetyki i Węgla Brunatnego, Delegatura Zarządu w Katowicach, Instrukcja łączeń ruchowych w sieciach elektroenergetycznych, Katowice, 1989.

[6] Egan L.G., Certification of software: is it necessary? Second IFAC Workshop on Safety and Reliability in Emerging Control Technologies, Dayton Beach, Florida, USA, 1-3 November 1995.

[7] Kuchta D.: Systematyczne testowanie programów - zalety i wady, Informatyka, nr 4 i 5, 1991.

[8] Fullwood R.R., Hall R.E., Probabilistic Risk Assessment in the Nuclear Power Industry. Fundamentals and Applications. Pergamon Press, 1988.

[9] Naur P., Randell B.(eds.): Software Engineering: Report on the Conference Sponsored by the NATO Science Committee. Garmish, Germany, 7-11 October 1968, NATO Scientific Affairs Divission, Brussels, 1969.

[10] Laprie J.C.: Dependability: from Concepts to Limits, SAFECOMP'93. Proceedings of the 12th International Conference on Computer Safety, Reliability and Security. Poznań-Kiekrz, 27-29 October 1993.

[11] Framework for the Evaluation of Safety Critical Objects, Projekt FRESCO (1993-1996), folie z prezentacji.

[12] Weizenbaum J.: Human authority, responsibility and accountability in large-scale real-time systems. Real-Time Data Handling and Process Control, Proceeding of the First European Symposium held in Berlin. 23-25 October 1979, Ed. H. Meyer, North-Holland Publishing Company, 1980.

[13] Mumford L.: Interpretations and Forecasts: 1922-1972. Chapter 24: The First Megamachine. Harcourt Brace Jovanovich Inc., New York.

[14] Mumford L.: The Myth of the Machine. Technics and Human Development, Chapter 9: The Design of the Megamachine. Harcourt, Brace and World Inc., New York.