

# Andrzej Kobyliński

---

## Rola i zasięg standardów w rozwoju oprogramowania

---

Ekonomiczne Problemy Usług nr 104, 329-339

---

2013

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.

ANDRZEJ KOBYLŃSKI

Szkoła Główna Handlowa w Warszawie

## ROLA I ZASIĘG STANDARDÓW W ROZWOJU OPROGRAMOWANIA

### Wprowadzenie

Wytwarzanie oprogramowania nie ma zbyt długiej historii – pierwsze programowane komputery powstały zaledwie niespełna 70 lat temu. Po pierwszym okresie burzliwego, niemniej chaotycznego rozwoju sytuacja dojrzała do tego, by wejść w stadium normalizacji. O ile wszystkie konstruowane w tym pierwotnym okresie komputery wyposażane były w pisany specjalnie dla nich system operacyjny, to stopniowo powszechnie zaakceptowanych zostało kilka systemów operacyjnych, które działały na komputerach pochodzących od wielu producentów. O ile wszystkie wczesne komputery posiadały unikalny, zrozumiały tylko dla nich symboliczny język programowania, to stopniowo popularność zdobyły uniwersalne języki programowania (początkowo Cobol i Fortran), dla których kompilatory stały się nieodzownym elementem oprogramowania systemowego. Ale sam proces wytwarzania oprogramowania nie podlegał tak szybkiej unifikacji. Uważa się, że pierwszą historycznie normą, odnoszącą się do budowy produktów programowych, był standard zapewniania jakości oprogramowania powstały w 1972 roku w Departamencie Obrony Stanów Zjednoczonych<sup>1</sup>, a więc po niespełna 30 latach doświadczeń w konstruowaniu oprogramowania.

W ciągu kolejnych 40 lat sytuacja zmieniła się znacząco. Światło dzienne ujrzało kilkaset (!) standardów odnoszących się wyłącznie do różnych aspektów związanych z procesem wytwarzania oprogramowania: cyklu życia oprogramowa-

---

<sup>1</sup> K. El Emam, J.-N. Drouin, W. Melo (eds.): *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*, IEEE Computer Society, Los Alamitos 1998.

nia, zarządzania przedsięwzięciem informatycznym jako całością i jego poszczególnymi fazami, dokumentowania poszczególnych faz i czynności, zarządzania jakością zarówno produktów, jak i procesów programowych itp. Ale świadomość istnienia i przydatności wykorzystywania różnego rodzaju norm wydaje się być niedostateczna. Stanowi to kontrast między liczbą opracowanych standardów a ich popularnością. Tymczasem stosowanie norm, które stanowią przecież zapis uznanych i powszechnie akceptowanych wzorców praktyk i produktów, mogłoby wpłynąć na zmniejszenie napięć, które często pojawiają się na styku klient-producent oprogramowania, a wynikają z wzajemnego niezrozumienia, a ostatecznie przyczynić się do terminowego i w ramach przewidzianego dostarczania dobrych pod względem jakościowym produktów programowych.

Głównym celem artykułu jest wykazanie, że tak duża liczba standardów, może stanowiąca pozorne utrudnienie w pracy, świadczy o dojrzewaniu tego specyficznego rodzaju produkcji, jakim jest produkcja oprogramowania.

## 1. Typy standardów

Istnieją cztery podstawowe typy standardów, które różnią się ze względu na sposób ich powstania.

1. standardy *de iure* – obowiązujące na mocy prawa;
2. standardy *de facto* – zbiór zwyczajów i technologii stosowanych powszechnie, który nie jest uznany przez żadną z formalnych organizacji standaryzacyjnych;
3. standardy dobrowolne (ang. *voluntary consensus*) – zaprojektowane przez organizacje lub ciała przedstawicielskie zawiązane w celu utworzenia standardu;
4. standardy firmowe – obowiązujące w poszczególnych firmach.

Przykłady:

**Ad 1.** W odniesieniu do produkcji oprogramowania trudno jest podać odpowiedni przykład dotyczący polskich warunków, dlatego posłużymy się przykładami zagranicznymi.

Departament Obrony USA, największy na świecie odbiorca oprogramowania wykonywanego na zamówienie, miał stałe kłopoty ze swymi poddostawcami, związane z ich nierzetelnością w odniesieniu do kosztów, harmonogramu i jakości. Jako remedium miało posłużyć opracowanie pięciostopniowego modelu oceny dojrzałości organizacji wytwarzających oprogramowanie Capability Maturity Model (CMM). Firmy pragnące wziąć udział w przetargach na dostawę oprogramowania muszą legitymować się osiągnięciem co najmniej trzeciego poziomu dojrzałości.

Model ten, w kolejnych wersjach (obecnie obowiązuje Capability Maturity Model Integration 1.3<sup>2</sup>), zyskał olbrzymią popularność.

Przykładami innych standardów obligatoryjnych w pewnych określonych warunkach były:

- Military Standard. Software Development and Documentation. MIL-STD-498 – opisujący sposób konstruowania i dokumentowania oprogramowania we wszystkich departamentach i agencjach podległych Departamentowi Obrony USA<sup>3</sup>,
- Software Development Standard for the German Federal Armed Forces, General Directive 250 – Software Life Cycle Process Model (1992) – tzw. V-Model cyklu życia oprogramowania (Niemcy)<sup>4</sup>.

**Ad 2.** W dziedzinie produkcji oprogramowania wiele standardów ma ten właśnie charakter. Standard tego rodzaju rozpowszechnia się, gdy formalny standard nie istnieje, ma poważne ułomności lub z przyczyn historycznych nie jest rozpowszechniony. Przykładami standardów *de facto*, nieodnoszącymi się wprawdzie do produkcji oprogramowania, ale pochodzącymi z informatyki, były i są na przykład systemy operacyjne IBM OS 360/370, Microsoft Windows, PostScript (uniwersalny język opisu strony opracowany w Adobe Systems, który stał się standardem w zastosowaniach poligraficznych), format DOC (który jest powszechnie wykorzystywany z powodu rozpowszechnienia MS Office). Wymienione produkty zostały opracowane przez wiodących producentów i znalazły szerokie zastosowanie na całym świecie. W miarę wzrostu ich popularności inni wytwórcy dostosowywali swoje produkty tak, by mogły być wykorzystywane w najpopularniejszym w owym czasie środowisku, co zwiększało krąg potencjalnych odbiorców ich wyrobów. Stało się to oczywiście bez żadnego formalnego przymusu – to presja rynku i możliwość zwiększenia sprzedaży spowodowała, że zauważyli oni swój interes w dostosowaniu się do standardu.

**Ad 3.** Jest to największa grupa standardów. Standardy tego rodzaju są opracowywane przez krajowe organizacje standaryzacyjne (np. BSI – British Standard Institute), międzynarodowe organizacje standaryzacyjne (np. ISO International Organization for Standardization) lub przez organizacje zawodowe (np. IEEE – Institute of Electrical and Electronics Engineers). Więcej informacji na temat tych organizacji zawartych zostało w następnym rozdziale. Normy uchwalone przez te organizacje nie mają charakteru obligatoryjnego, poszczególni producenci oprogramowania mogą, ale nie muszą stosować się do zaleceń standardu.

---

<sup>2</sup> CMMI Institute, <http://cmmiinstitute.com> [dostęp 26.01.2013].

<sup>3</sup> Military Standard. Software Development and Documentation. MIL-STD-498, Department of Defense USA, 1994.

<sup>4</sup> S. Zahran: *Software Process Improvement. Practical guidelines for business success*, Addison-Wesley, 1998.

Najlepszym chyba przykładem normy takiego rodzaju jest seria standardów ISO 9000<sup>5</sup>, które dotyczą systemu zarządzania jakością w organizacjach. Organizacja, która chce certyfikować zgodność zaimplementowanego w niej systemu zarządzania jakością z odpowiednią normą, występuje do wybranej przez siebie organizacji certyfikacyjnej o przeprowadzenie auditu i certyfikację systemu.

Normy zaliczane do tej grupy, szczególnie uchwalane przez ISO, są ze swej natury bardzo ogólne, a w szczególności nie zawierają żadnych zaleceń dotyczących technologii. Powodem jest to, że ich opracowanie i uchwalanie trwa bardzo długo – nawet stosując szybką ścieżkę uchwalania trwa to kilka lat. Dobrym przykładem z dziedziny standardów dotyczących inżynierii oprogramowania jest norma ISO/IEC 15504 Information Technology – Process Assessment, nad którą zorganizowane prace rozpoczęły się w 1992/3 roku, a kolejne części tej normy były ostatecznie aprobowane w latach 2004–2012<sup>6</sup>.

**Ad 4.** Prócz trzech wymienionych tu typów standardów można też mówić o czwartej grupie: standardach wewnątrzorganizacyjnych. Są to wewnętrzne standardy obowiązujące w poszczególnych firmach (organizacjach programistycznych), mające formę okólników lub zarządzeń wewnętrznych, normujących różne, niekiedy szczegółowe aspekty działalności firmy. Szczególną rolę pełnią tu przewodniki stylu, standaryzujące sposób kodowania programu.

Za standardy należące do tej grupy można też uznać zalecane przez niektórych czołowych producentów narzędzi programistycznych sposoby zarządzania przedsięwzięciami informatycznymi i wytwarzania oprogramowania (np. Microsoft Solution Framework – MSF lub Rational Unified Process, RUP – IBM).

## 2. Organizacje standaryzacyjne

Istnieją trzy typy organizacji zaangażowanych w opracowywanie formalnych standardów: międzynarodowe, krajowe i zawodowe.

### 2.1. Organizacje międzynarodowe

Najbardziej znaną międzynarodową organizacją działającą na niwie standaryzacyjnej jest bez wątpienia International Organization for Standardization (ISO). Organizacja powstała w 1947 roku w celu inicjowania, wspomagania i promowania standardów międzynarodowych, głównie w celu ułatwienia współpracy i wymiany międzynarodowej. Aktualnie do ISO należy 164 krajowych organizacji normaliza-

---

<sup>5</sup> PN-EN ISO 9000:2006(U) System zarządzania jakością – Podstawy i terminologia; PN-EN ISO 9001:2009 System zarządzania jakością – Wymagania; PN-EN ISO 9004:2010 Zarządzanie mające na celu osiągnięcie trwałego sukcesu organizacji – Podejście przez zarządzanie jakością; PN-EN ISO 19011:2012 (U) Wytyczne dotyczące auditowania systemów zarządzania.

<sup>6</sup> <http://www.iso.org/iso/home/search.htm?qt=15504&sort=rel&type=simple&published=on>.

cyjnych (112 z nich posiada pełne uprawnienia)<sup>7</sup>. Od czasu powstania ISO pod jej auspicjami powstało ponad 19 000 norm dotyczących różnych dziedzin życia i gospodarki. ISO dzieli się na ponad 200 komitetów technicznych<sup>8</sup>, część z nich podzielona jest dodatkowo na podkomitety. Każdy komitet odpowiedzialny jest za rozwój standardów z określonej dziedziny. W dziedzinie norm dotyczących szeroko pojętej technologii informacyjnej ISO współdziała z International Electrotechnical Commission – IEC<sup>9</sup> (działająca w zakresie norm elektrycznych i elektronicznych od 1906 roku) w ramach pierwszego połączonych komitetu technicznego (Joint Technical Committee 1 – JTC1)<sup>10</sup>. JTC1 opracował dotychczas 330 norm. Komitet ten dzieli się z kolei na 19 aktywnych podkomitetów (SC – subcommittees) odpowiedzialnych za poszczególne obszary wymagające standaryzacji (np. techniki bezpieczeństwa, języki programowania, wymiana informacji, kodowanie znaków itp.). Podkomitet 7 (SC7) odpowiada za inżynierię oprogramowania i zakresem jego działania jest standaryzacja procesów, wspomagających narzędzi i technologii w tworzeniu produktów programowych i systemów. Podkomitet JTC1/SC7 jest bardzo aktywny i dotychczas opracował 146 norm z zakresu inżynierii oprogramowania i systemów<sup>11</sup>. Kilka przykładowych norm z tego zakresu zawiera tabela 1.

Tabela 1

Przykładowe normy z zakresu inżynierii oprogramowania przyjęte przez ISO/IEC JTC1 SC7

Numer normy	Nazwa normy
ISO/IEC 12207:2008	Systems and software engineering – Software life cycle processes
ISO/IEC 14764:2006	Software Engineering – Software Life Cycle Processes – Maintenance
ISO/IEC 15504-1:2004	Information technology – Process assessment – Part 1: Concepts and vocabulary
ISO/IEC TR 19759:2005	Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK)
ISO/IEC 90003:2004	Software engineering – Guidelines for the application of ISO 9001:2000 to computer software

Źródło: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_tc\\_browse.htm?commid=45086&published=on](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=45086&published=on) [dostęp 25.01.2013].

<sup>7</sup> [http://www.iso.org/iso/home/about/iso\\_members.htm](http://www.iso.org/iso/home/about/iso_members.htm) [dostęp 25.01.2013].

<sup>8</sup> [http://www.iso.org/iso/home/standards\\_development/list\\_of\\_iso\\_technical\\_committees.htm](http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees.htm).

<sup>9</sup> <http://www.iec.ch/>.

<sup>10</sup> [http://www.iso.org/iso/home/standards\\_development/list\\_of\\_iso\\_technical\\_committees/iso\\_technical\\_committee.htm?commid=45020](http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=45020).

<sup>11</sup> [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_tc\\_browse.htm?commid=45086&published=on](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=45086&published=on).

## 2.2. Krajowe organizacje normalizacyjne

Każdy kraj biorący udział w pracach ISO może być reprezentowany przez jedną organizację normalizacyjną. W Polsce jest nią Polski Komitet Normalizacyjny (PKN)<sup>12</sup>. Organizacje reprezentujące niektóre państwa zostały wymienione w tabeli 2.

Tabela 2

Przykładowe normy z zakresu inżynierii oprogramowania przyjęte przez ISO/IEC JTC1 SC7

Skrót nazwy	Członek ISO	Kraj
AFNOR	Association Française de Normalisation	Francja
JISC	Japanese Industrial Standards Committee	Japonia
DIN	Deutsches Institute für Normung	Niemcy
PKN	Polski Komitet Normalizacyjny	Polska
ANSI	American National Standards Institute	Stany Zjednoczone
BSI	British Standards Institute	Wielka Brytania

Źródło: [http://www.iso.org/iso/home/about/iso\\_members.htm](http://www.iso.org/iso/home/about/iso_members.htm) [dostęp 25.01.2013].

Każda krajowa organizacja normalizacyjna może na terenie kraju uchylać powstanie własnych norm. Od czasu, gdy Polska stała się członkiem Unii Europejskiej, rola Polskiego Komitetu Normalizacyjnego znacznie spadła. PKN zajmuje się przede wszystkim wprowadzaniem norm europejskich, tworzonych przez Europejski Komitet Normalizacyjny (identycznych we wszystkich krajach członkowskich Unii), normy te mają oznaczenie PN-EN (w Niemczech DIN-EN itd.). Z drugiej strony PKN, będący przedstawicielem Polski w ISO, uprawniony jest do występowania i głosowania w każdym Komitecie technicznym ISO podczas uchwalania norm międzynarodowych. Uchwalenie normy wymaga, by opowiedziało się za nią przynajmniej 75% członków.

W połączonej komisji ISO/IEC JTC1 bierze udział 35 organizacji, PKN jest tam wyłącznie obserwatorem (podobnie jak 55 organizacji z innych krajów).

## 2.3. Organizacje zawodowe

Krajowym i międzynarodowym organizacjom normalizacyjnym nieocenioną pomoc w opracowywaniu norm niosą organizacje zawodowe, skupiające profesjonalistów z poszczególnych dziedzin. Wyjątkową aktywność w tym obszarze wyka-

<sup>12</sup> Polski Komitet Normalizacyjny: <http://www.pkn.pl/>.

zuje Institute of Electrical and Electronic Engineers (IEEE)<sup>13</sup> – organizacja skupiająca specjalistów z obszaru elektroniki, elektrotechniki, ale również informatyki i telekomunikacji, i ogólnie nowych technologii. Jest to organizacja globalna, skupiająca 400 000 członków ze 160 krajów. Działalność normalizacyjna zawsze stanowiła istotną część działalności IEEE. Do tej pory ponad 1300 standardów zostało opracowanych przez tę organizację. W swojej historii IEEE opracował 338 norm z zakresu inżynierii oprogramowania i systemów, przy czym 119 spośród nich jest aktywnych<sup>14</sup>. W tabeli 3 zaprezentowano kilka wybranych standardów.

Tabela 3

Wybrane standardy IEEE z zakresu inżynierii oprogramowania

Numer normy	Nazwa normy
IEEE Std. 1012-2012	System and Software Verification and Validation
IEEE Std. 1633-2008	Recommended Practice on Software Reliability
IEEE Std. 24765-2010	Systems and software engineering – Vocabulary
ISO/IEC/IEEE 42010-2011	Systems and software engineering – Architecture description

Źródło: [http://standards.ieee.org/cgi-bin/lp\\_index?status=active&pg=40&type=standard&coll=12&offset=80](http://standards.ieee.org/cgi-bin/lp_index?status=active&pg=40&type=standard&coll=12&offset=80) [dostęp 25.01.2013].

### 3. Wady i zalety standaryzacji

Żaden z utworzonych standardów dotyczących produkcji oprogramowania nie wynikał wyłącznie z rozważań teoretycznych. Doprowadzenie do uchwalenia wszystkich standardów z tej dziedziny zawsze jest poprzedzone wieloletnim doświadczeniem. Dlatego też żaden ze standardów nie jest zbiorem teoretycznych wskazówek, lecz raczej sprawdzonych empirycznie reguł i zasad. Standardy zawierają najlepsze doświadczenia zdobyte przez wiodące firmy oraz przez doświadczonych członków i ekspertów komitetów normalizacyjnych. Nie jest to zbiór jednostkowych doświadczeń zdobytych przez jedną osobę w określonym otoczeniu, ale uogólniona „mądrość” zdobywana w wielu różnych środowiskach pracy przez wie-

<sup>13</sup> Institute of Electrical and Electronic Engineers: <http://www.ieee.org>.

<sup>14</sup> [http://standards.ieee.org/cgi-bin/lp\\_index?amp;pg=40&type=standard&coll=12&status=all](http://standards.ieee.org/cgi-bin/lp_index?amp;pg=40&type=standard&coll=12&status=all).



le lat. Dobrze świadczy o tym procedura uchwalania norm przez ISO, która trwa wiele (nawet 10) lat, szeroko dostępne i dyskutowane są bezpłatne próbne wersje, a w głosowaniu wymaganych jest aż 75% głosów popierających projekt. Innym faktem świadczącym o tym, że sytuacja musi dojrzeć do uchwalenia normy, jest to, że wspomniana już we wstępie pierwsza na świecie norma z zakresu budowy systemów informatycznych powstała dopiero po 30 latach doświadczeń w konstruowaniu oprogramowania.

Normy powstają na zasadzie uzgodnień osób reprezentujących organizacje uczestniczące w ich uchwalaniu. Pozycje wyjściowe osób przystępujących do rokowań są różne. Uzyskanie kompromisu wymaga oczywiście, by ustąpiły one ze swych początkowych stanowisk. Rezultat nie jest więc zwykle ideałem z punktu widzenia żadnej strony, natomiast stanowi zbiór sformułowań na tak wysokim poziomie uogólnienia, że wszyscy mogą się pod nim podpisać, lub taki zbiór stwierdzeń/wskaźników, z którymi wszyscy mogą się zgodzić. Normy są regularnie przeglądane i kontrolowane w świetle dotychczasowych doświadczeń ich użycia w praktyce, a zmieniane są dopiero po uzyskaniu powszechnej zgody.

Zaletą standardów jest to, że pozwalają ustalić<sup>15</sup>:

- terminologię,
- modele i procedury,
- wzorzec.

Ustalenie terminologii, jak również znaczenia stosowanych symboli graficznych jest trudne do przecenienia. Chyba wszyscy spotkali się z trudnościami w komunikacji wynikającymi z różnego rozumienia pojęć (synonimiczność, wieloznaczność). Ujednolicenie terminologii jest więc bardzo istotne.

Standardy opisują typowe modele cyklu życia z podziałem na fazy, czynności wykonywane we wszystkich fazach, dokumenty (ich formę, zawartość treściową, autora, odbiorcę) powstające w poszczególnych fazach, jak również typowe działania wykonywane przez cały cykl życia (głównie dotyczące zarządzania całym przedsięwzięciem, planowaniem, zapewnianiem jakości, zarządzaniem ryzykiem, zarządzaniem konfiguracją itp.).

Ze względu na to, że standard stanowi zbiór wymagań, na który zgodzili się wszyscy eksperci zaangażowani w jego powstanie, można uznać, że stanowi on pewne minimum, którego można wymagać od każdego, kto takim standardem się posługuje. Pozwala to uznać standard za pewien minimalny wzorzec. Jeśli proces lub produkt nie spełnia wymagań normy – oznacza to, że nie spełnia oczekiwanych wymagań opisanych we wzorcu.

Wadą i najpowszechniej podnoszonym problemem w czasie wdrażania dowolnego standardu inżynierii oprogramowania jest to, jak powiązać wymagania standardu z potrzebami i celami organizacji. Nie zawsze jest to łatwe. Niekiedy

---

<sup>15</sup> K. El Emam et al.: *op. cit.*

może się wydawać, że wymagania stawiane przez standard mogą mieć negatywny wpływ na osiąganie przez firmę wytyczonych celów i że narzucają reguły, które mogą zmniejszyć elastyczność i szybkość reakcji na bodźce zewnętrzne, że stanowią sztywny gorset ograniczający rozwój organizacji.

Zdarza się niekiedy, że w sytuacjach kryzysowych (np. braku czasu) firmy, które wdrożyły standardy, zarzucają posługiwanie się nimi i powracają do starej „metodyki” pracy chaotycznej. Świadczy to wyłącznie o niedojrzałości procesów produkcyjnych realizowanych w tych firmach. Powstałe w ubiegłym dziesięcioleciu metody oceny procesów produkcji oprogramowania (np. CMMI<sup>16</sup>, ISO/IEC 15504<sup>17</sup> i in.) pozwalają na zakwalifikowanie firm wytwarzających oprogramowanie na odpowiedni poziom dojrzałości. W firmach o wysokim poziomie realizowanych procesów takie przypadki się nie zdarzają, procesy są ustabilizowane i chwilowe przeszkody i niepowodzenia nie są w stanie skłonić kierownictwa ani pracowników do rezygnacji z posługiwania się standardami. Z drugiej strony, w środowisku istnieje przekonanie, że zdobywanie najwyższych poziomów dojrzałości nie wpływa korzystnie na efektywność ekonomiczną organizacji (co wynika z nadmiernych wymogów biurokratycznych).

Wadą niektórych norm jest ich nadmierna generalizacja. W takich przypadkach bezpośrednie zastosowanie norm nie jest możliwe. Wtedy zdarza się, że organizacje standaryzacyjne opracowują wytyczne ich stosowania. Przykładem takiej sytuacji jest PN-ISO/IEC 90003:2007 *Inżynieria oprogramowania – Wytyczne stosowania ISO 9001:2000 do oprogramowania komputerów*.

Różne jest w poszczególnych organizacjach wytwarzających oprogramowanie podejście do swobody twórczej programistów. W niektórych organizacjach obowiązują restrykcyjne standardy kodowania, w innych pozostawia się programistom swobodę stylu. W tych pierwszych obowiązują przewodniki stylu określające np., jaka jest maksymalna wielkość modułu, maksymalna głębokość zagnieżdżenia instrukcji warunkowych, w jaki sposób należy tworzyć identyfikatory itp. Przewodniki stylu są to niekiedy kilkudziesięciostronicowe podręczniki. Celem ustanowienia standardów kodowania jest ułatwienie pracy programistom w fazie konserwacji. Jeśli jednak stosowanie się do restrykcyjnych reguł stylu uczyni trudniejszą pracę programistów opracowujących oprogramowanie, to nie można się dziwić, że będą oni omijali ustalone ogólnie zasady. Nadmiernie ograniczające standardy kodowania przyczyniają się do spadku wydajności programistów oraz jakości wytwarzanego przez nich kodu, jeśli granice ich „wolności twórczej” są zbyt wąskie. Stąd wydaje się konieczne, by przewodniki stylu stosować w sposób elastyczny i nie obawiać się modyfikować standardów nawet już w trakcie realizacji przedsięwzięcia.

<sup>16</sup> CMMI Institute: <http://cmmiinstitute.com>.

<sup>17</sup> ISO/IEC 15504-1:2004 Information technology – Process assessment – Part 1–10.

## Podsumowanie

Standaryzacja w naszym codziennym życiu jest niedostrzegalna. Ale chociaż tak się do niej przyzwyczailiśmy, że jej nie zauważamy, to jest ona obecna we wszystkich aspektach naszej egzystencji. Jej zawdzięczamy przystępne ceny, prostotę obsługi, przewidywalność, bezpieczeństwo. Kiedy kupujemy komputer od dowolnego z setek dostawców, to możemy mieć pewność, że da się na nim zainstalować nasz ulubiony system operacyjny. Kiedy dostaniemy plik w formacie RTF, to możemy mieć pewność, że da się przeczytać przy pomocy dowolnego edytora tekstów. Kiedy kupujemy żarówkę z dużym gwintem, to mamy pewność, że będzie pasowała do naszej przedwojennej jeszcze lampy. Kiedy jedziemy samochodem – ufamy, że inni użytkownicy drogi, podobnie jak i my, będą się zachowywać w sposób ustandaryzowany, opisany w ustawie, nie powodując zagrożenia. Standaryzacja jako koncepcja rozwoju cywilizacji od setek lat wykazuje swoją skuteczność i stała się kluczem do postępu.

Produkcja oprogramowania jest nową dziedziną, mającą dopiero niespełna siedemdziesięcioletnią historię. Celem artykułu było wykazanie, że jeśli produkcja oprogramowania ma bardziej przypominać produkcję przemysłową niż rzemieślniczą, to powinna zostać w większym stopniu ustandaryzowana. Podstawy w tym kierunku zostały już uczynione. Istnieje kilkaset standardów dotyczących różnych aspektów inżynierii systemów i oprogramowania, nic nie stoi na przeszkodzie, by je wykorzystywać. Oczywiście nikt odgórnie nie narzuci firmom, by je wykorzystywały. Ale taka jest nieuchronna konieczność. W związku z postępującą globalizacją produkcji, w tym i produkcji oprogramowania, często analiza systemu, który ma być wykorzystywany w Europie, dokonywana jest na miejscu, projektowanie i implementacja przeniesione są np. do Indii, a testowaniem zajmują się ludzie pracujący w Europie, Indonezji i USA, tak że proces testowania odbywa się nieustannie przez całą dobę. Taki podział pracy powoduje obniżkę kosztów i skrócenie cyklu wytwórczego.

W ostatnim wieku idea standaryzacji zyskała w innych dziedzinach przemysłu rangę powszechnie stosowanego i twórczego strategicznego podejścia do biznesu. Produkcję oprogramowania czeka jeszcze długa droga, by dzięki standaryzacji w projektowaniu, implementowaniu, wdrażaniu i eksploataowaniu dało się osiągnąć podobne sukcesy. Ale początki są już zrobione – istnieją już standardy, teraz trzeba przekonać środowisko, że ich użycie może być korzystne dla różnych zainteresowanych stron.

## Literatura

1. *CMMI Institute*: <http://cmmiinstitute.com>.
2. El Emam K., Drouin J.-N., Melo W. (eds.): *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*, Wiley-IEEE Computer Society, 1997.

3. *Institute of Electrical and Electronic Engineers*, <http://www.ieee.org>.
4. *International Electrotechnical Commission*: <http://www.iec.ch>.
5. *International Organization for Standardization*: <http://www.iso.org>.
6. ISO/IEC 15504-1:2004 *Information technology – Process assessment – Part 1-10*.
7. *Military Standard. Software Development and Documentation*. MIL-STD-498, Department of Defense USA, 1994.
8. PN-EN ISO 9000:2006(U) *System zarządzania jakością – Podstawy i terminologia*.
9. PN-EN ISO 9001:2009 *System zarządzania jakością – Wymagania*.
10. PN-EN ISO 9004:2010 *Zarządzanie mające na celu osiągnięcie trwałego sukcesu organizacji – Podejście przez zarządzanie jakością*.
11. PN-EN ISO 19011:2012 (U) *Wytuczne dotyczące auditowania systemów zarządzania*.
12. PN-ISO/IEC 90003:2007 *Inżynieria oprogramowania – Wytuczne stosowania ISO 9001:2000 do oprogramowania komputerów*.
13. *Polski Komitet Normalizacyjny*, <http://www.pkn.pl>.
14. Zahran S.: *Software Process Improvement. Practical guidelines for business success*, Addison-Wesley, 1998.

## THE ROLE AND SCOPE OF STANDARDS IN SOFTWARE DEVELOPMENT

### Summary

The existence of standards for software engineering is not obvious to many professionals. Meanwhile, there are a number of standards in this area, which has been designed by different standardization bodies. The use of existing standards, with a relatively small inconvenience, can bring very significant benefits to organizations. As the rise globalization processes of the world economy, the role of common standards is increasing, also in the field of software development. The paper presents the types of standards, standardization bodies as well as the benefits and disadvantages of standardization.

*Translated by Andrzej Kobyliński*