

ANDRZEJ KOBYLIŃSKI
Kolegium Analiz Ekonomicznych
Szkoła Główna Handlowa w Warszawie

Ewolucja norm jakości produktu programowego

1. Wstęp

Kwestia jakości oprogramowania budzi wiele kontrowersji: wprawdzie wszyscy oczekują oprogramowania dobrej jakości, ale poszczególne grupy zainteresowanych osób mogą różnić się swym stosunkiem do problemu jakości. Dla niektórych użytkowników dobrym jakościowo oprogramowaniem jest to, które jest niezawodne i rzadko ulega awariom, dla innych istotne jest, by było intuicyjne w użyciu i przyjazne, jeszcze inni zwracają szczególną uwagę na to, by było dobrze dostosowane do ich potrzeb, zapewniając odpowiedni zestaw funkcjonalności. Użytkownicy, których dotyczą ograniczenia finansowe, chcą, aby nie wiązały się z nim wygórowane wymagania sprzętowe i aby działało szybko nawet na gorszym sprzęcie, a dodatkowo, by było tanie, a jeszcze lepiej bezpłatne. Niektóre osoby, szczególnie uwrażliwione na kwestię prywatności, mogą uznać, że najważniejsze w oprogramowaniu jest to, by było ono bezpieczne. Z kolei osoby, których zakres obowiązków obejmuje konserwację (pielęgnację, utrzymanie, ang. *maintenance*) oprogramowania, mogą uważać, że dobrym jakościowo oprogramowaniem jest takie, które jest łatwe w utrzymaniu – łatwo się z niego usuwa zgłaszane błędy, dostosowuje je do zmieniających uwarunkowań zewnętrznych (zmian w prawie, zmian sprzętu i oprogramowania systemowego), wbudowuje w nie nowe funkcjonalności.

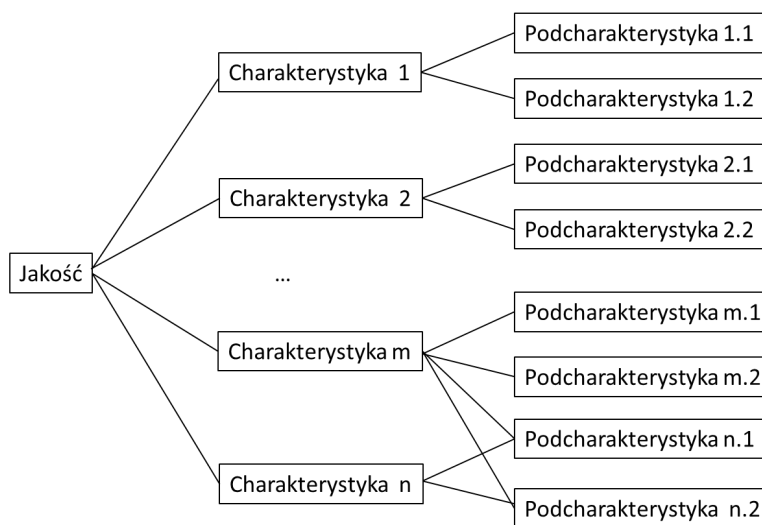
Próby ustandaryzowanego podejścia do jakości oprogramowania mają już stosunkowo długą historię. Za pierwszą historycznie normę odnoszącą się do budowy produktów programowych uznaje się standard zapewniania jakości oprogramowania powstały w 1972 r. w Departamencie Obrony Stanów

Zjednoczonych¹. Pod koniec lat 70. i w latach 80. XXw. zaproponowano kilka modeli jakości oprogramowania. Na początku lat 90. został opublikowany pierwszy międzynarodowy model jakości oprogramowania (norma ISO/IEC 9126). Stopniowo norma ta była udoskonalana.

Celem pracy jest zaprezentowanie ewolucji tego modelu jakości oprogramowania (obecnie opisanego w zestawie norm serii ISO/IEC 25000) i przedstawienie jego niedogodności i zalet.

2. Krótka (pre)historia modeli jakości oprogramowania

Model jakości można zdefiniować jako taki, którego celem jest opis, ocena i/lub przewidywanie jakości. W zasadzie wszystkie opracowane dotychczas modele jakości oprogramowania mają strukturę, którą można przedstawić tak, jak zostało to zrobione na rysunku 1.

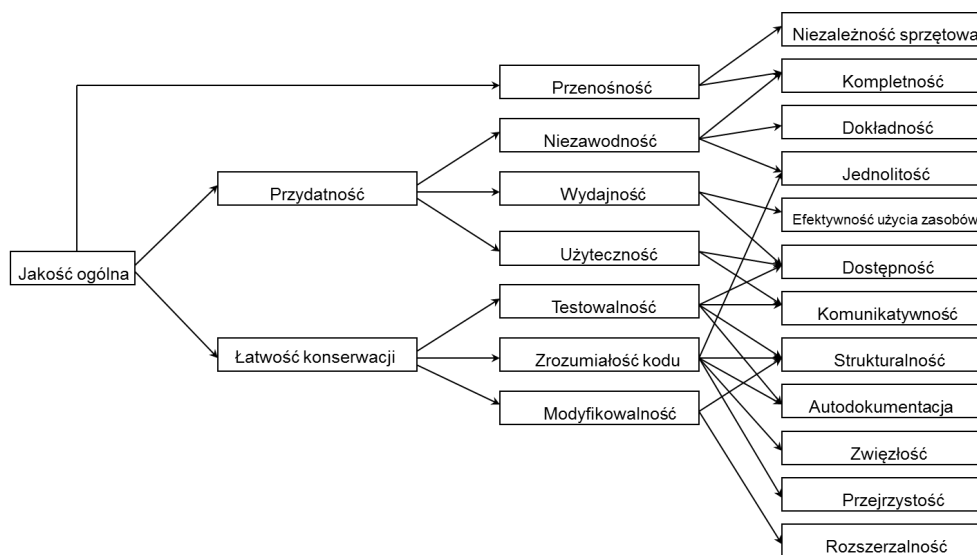


Rysunek 1. Ogólna struktura modeli jakości oprogramowania

Źródło: opracowanie własne.

¹ *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*, red. K. El Emam, J.-N. Drouin, W. Melo, IEEE Computer Society, Los Alamitos 1998, s. 3.

Przy tego typu podejściu tak trudną do opisaną kategorię, jaką jest jakość, opisuje się za pomocą kilku lepiej zdefiniowanych, łatwiejszych do zmierzenia właściwości (zwanych też cechami, atrybutami, charakterystykami). Zwykle w celu dokonania łatwiejszego pomiaru te charakterystyki wysokopoziomowe dekomponuje się na atrybuty niższego poziomu, zwane zazwyczaj podcharakterystykami jakości. Im z kolei przypisywane są bezpośrednio mierzalne wskaźniki, zwane miarami (metrykami) jakości. Poszczególne podcharakterystyki mogą być jednoznacznie przypisane do charakterystyk (struktura hierarchiczna, którą na rysunku 1 obrazują charakterystyka 1 i charakterystyka 2 oraz przypisane im podcharakterystyki) albo podcharakterystyki mogą być jednocześnie przypisane do kilku charakterystyk (struktura sieciowa, którą na rysunku 1 obrazują podcharakterystyka n.1 oraz podcharakterystyka n.2). Dla przykładu, na rysunku 2 został zaprezentowany pochodzący z 1978 r. model jakości oprogramowania autorstwa B. Boehma. W tym samym okresie i nieco później, czyli w latach 80., powstało wiele podobnych modeli: McCalla, Boeinga, FURPS, CUPRIMDSO i inne².



Rysunek 2. Modele jakości B. Boehma

Źródło: B.W. Boehm, J.R. Brown, H. Kaspar, M. Lipow, G.J. MacLeod, M.J. Merritt, *Characteristics of Software Quality*, Amsterdam 1978.

² Krótki opis tych i innych jeszcze modeli można znaleźć w: A. Kobylński, *Modele jakości produktów i procesów programowych*, Oficyna Wydawnicza SGH, Warszawa 2005, s. 72–82.

3. Pierwsza norma międzynarodowa – model ISO/IEC 9126

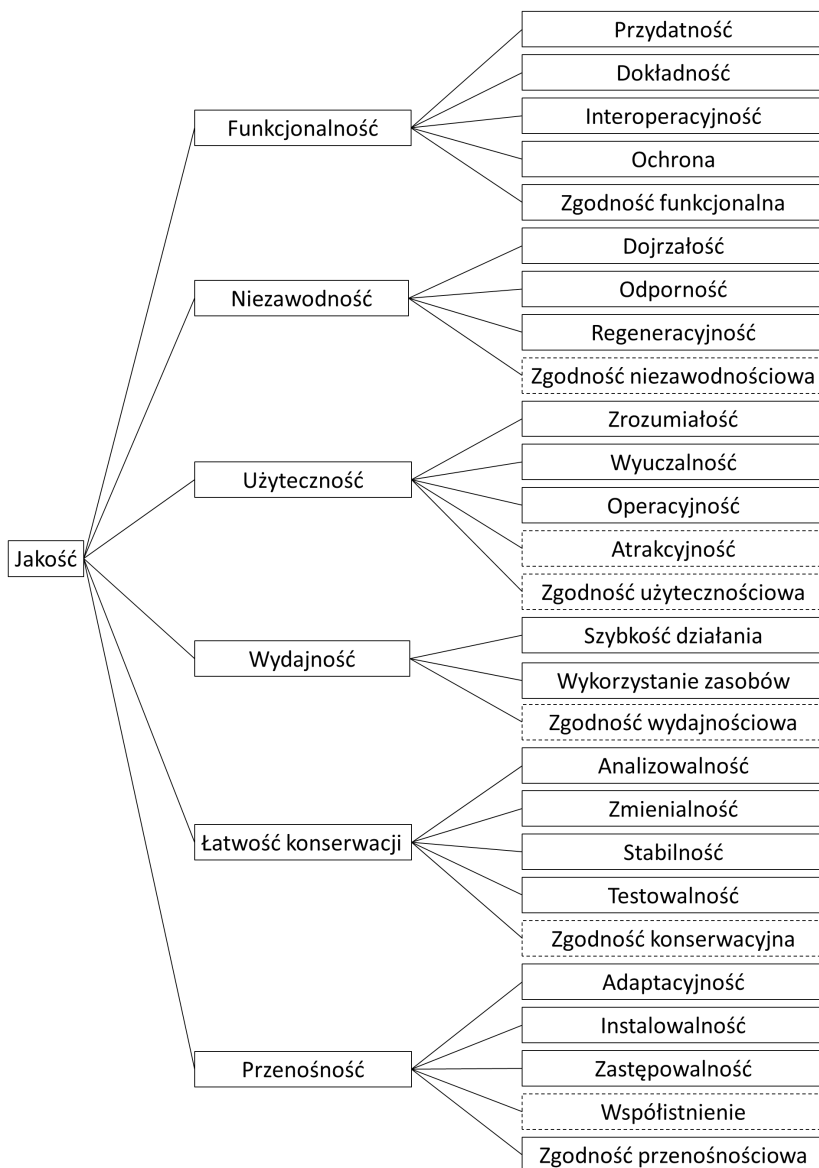
Istnienie licznych, konkurencyjnych modeli jakości oprogramowania spowodowało, że pojawiła się koncepcja ujednoczenia dotychczasowych modeli. Organizacją stworzoną do tego rodzaju działań jest ISO (International Organization for Standardization), która we współpracy z IEC (International Electrotechnical Commission) powołała wspólny komitet techniczny (ang. *Joint Technical Committee*) ISO/IEC JTC 1. W skład takiego komitetu wchodziły narodowe organizacje normalizacyjne zainteresowanych krajów. Prace rozpoczęły się w 1985 r., a w 1991 r. norma została oficjalnie zaaprobowana jako *ISO/IEC 9126:1991 Information technology – Software product evaluation – Quality characteristics and guidelines for their use*. Zgodnie z zasadami, aby norma została przegłosowana, musi za nią zagłosować co najmniej 75% uprawnionych do głosowania komitetów narodowych (przedstawicielem Polski w ISO jest Polski Komitet Normalizacyjny). Konieczność osiągnięcia tak dużej zgodności powoduje, że zazwyczaj normy mają charakter bardzo ogólny i są neutralne pod względem technologii, która zmienia się bardzo szybko, podczas gdy normy są zwykle aktualizowane nie częściej niż co kilka lat.

Standard ISO/IEC 9126:1991 był wyjątkowo lapidarny: część normatywna liczyła zaledwie 8 stron, a część informacyjna – 5. W części normatywnej został opisany zarówno model jakości produktu programowego, zawierający specyfikację i opis charakterystyk jakości, jak i bardzo ogólnie procedura dokonywania oceny jakości³. Dodatek A (w części informacyjnej) zawierał uzupełnienie modelu jakości o specyfikację i definicje podcharakterystyk jakości. Rysunek 3 przedstawia model jakości ISO/IEC 9126:1991. Charakterystyki i podcharakterystyki zostały zaznaczone na rysunku linią ciągłą.

W dalszej pracy nad standardem rozdzielono model jakości od procesu dokonywania oceny. W latach 1998–2001 wydano sześć części normy *ISO/IEC 14598 Software engineering – Product evaluation*, w której szczegółowo opisano kwestie związane z planowaniem, zarządzaniem i realizacją procesu oceny jakości. Natomiast nowa wersja normy ISO/IEC 9126, której pierwsza część została opublikowana w 2001 r.⁴ (kolejne trzy części wydawano aż do 2004 r.), zawierała znacznie rozbudowany w stosunku do starej wersji model jakości.

³ Sposoby dokonywania oceny jakości i proces pomiaru leżą poza zakresem tematycznym artykułu.

⁴ *ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model*.



Atrybuty dodane w normie z 2001 r. otoczone zostały ramką z linii przerywanej.

Rysunek 3. Modele jakości ISO/IEC 9126:1991

Źródło: opracowanie własne na podstawie: *ISO/IEC 9126:1991 Information technology – Software product evaluation – Quality characteristics and guidelines for their use*; *ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model*.

Rozbudowa ta miała charakter bardziej formalny niż merytoryczny. Przede wszystkim do nowej wersji zostało przeniesionych ze starej wersji bez żadnych

zmian sześć atrybutów jakości. Główna zmiana polegała na tym, że podcharakterystyki, do tej pory znajdujące się w części informacyjnej normy, zostały przesunięte do części normatywnej. Model został uzupełniony o kilka dodatkowych podcharakterystyk, które na rysunku 3 zostały zaznaczone linią przerywaną.

Kolejną, tym razem istotną merytorycznie zmianą było opracowanie dodatkowego modelu jakości użytkowej. Twórcy normy uznali, że model jakości z 1991 r., zmodyfikowany w 2001 r., specyfikujący sześć atrybutów jakości, ma szczególne znaczenie dla twórców oprogramowania. Doszli jednocześnie do wniosku, że przydatny byłby alternatywny model – nadrzędny w stosunku do sześciu cech jakościowych i obrazujący postrzeganie jakości produktu programowego z punktu widzenia użytkownika końcowego, który może wykorzystywać oprogramowanie w różnych warunkach i którego niekoniecznie muszą interesować kwestie utrzymania i przenośności oprogramowania. Model jakości użytkowej pozostaje poza zakresem tematycznym niniejszego artykułu.

Model jakości ISO/IEC 9126:2001 był zawarty w części 1 normy, a w częściach 2, 3 i 4 zostały opisane odpowiednio miary jakości wewnętrznej, zewnętrznej i użytkowej.

4. Seria norm ISO/IEC 25000

Rezultatem rozdzielenia krótkiej normy ISO/IEC 9126:1991 na dwie powiązane normy wieloczęściowe – ISO/IEC 9126 (*Jakość produktów programowych*) i ISO/IEC 14598 (*Ocena produktów programowych*) – było to, że – wprawdzie obie serie mają wspólne korzenie i stanowiły komplementarny zbiór norm – jednak ich niezależne cykle życia doprowadziły ostatecznie do znacznych niespójności. Innym dość powszechnie podnoszonym problemem było nieuwzględnianie przez model atrybutów powszechnie uważanych za istotne, np. bezpieczeństwa (ang. *safety*). Doprowadziło to podjęcia inicjatywy SQuaRE (*Software Product Quality Requirement and Evaluation*), mającej na celu doprowadzenie do ponownego uspoźnienia tych dwóch nurtów: specyfikacji wymagań wobec jakości oprogramowania i oceny jakości oprogramowania. Efektem prac nad nową serią norm jest powstanie całej grupy standardów o numeracji 25000: pierwsza norma powstała w 2005 r., a obecnie (grudzień 2014 r.) jest ich 21, przy czym część z nich nie została jeszcze oficjalnie zaaprobowana.

Poszczególne normy serii 25000 dotyczą różnych aspektów jakości, ale najważniejsza w kontekście tematyki niniejszego artykułu jest norma ISO/IEC

25010:2011 *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*, stanowi ona bowiem bezpośrednią kontynuację normy ISO/IEC 9126:2001 i w niej są zawarte model jakości oprogramowania i model jakości użytkowej. Model jakości oprogramowania został przedstawiony na rysunku 4.

Functional suitability (przydatność funkcjonalna)	<ul style="list-style-type: none"> • Functional completeness (kompletność funkcjonalna) • Functional correctness (poprawność funkcjonalna) • Functional appropriateness (adekwatność funkcjonalna)
Performance efficiency (efektywność wykonania)	<ul style="list-style-type: none"> • Time behaviour (szybkość działania) • Resource utilisation (wykorzystanie zasobów) • Capacity (przepustowość)
Compatibility (zgodność)	<ul style="list-style-type: none"> • Co-existence (współistnienie) • Interoperability (interoperacyjność)
Usability (użyteczność)	<ul style="list-style-type: none"> • Appropriateness recognisability (właściwa rozpoznawalność) • Learnability (wyczuwalność) • Operability (operacyjność) • User error protection (ochrona przed błędami użytkownika) • User interface aesthetics (estetyka interfejsu) • Accessibility (dostępność)
Reliability (niezawodność)	<ul style="list-style-type: none"> • Maturity (dojrzałość) • Availability (gotowość) • Fault tolerance (tolerancja błędów) • Recoverability (odzyskiwalność)
Security (ochrona)	<ul style="list-style-type: none"> • Confidentiality (poufność) • Integrity (integralność) • Non-repudiation (niezaprzeczalność) • Accountability (rozliczalność) • Authenticity (autentykacja)
Maintainability (łatwość konserwacji)	<ul style="list-style-type: none"> • Modularity (modularność) • Reusability (możliwość powtórnego użycia) • Analysability (analizowalność) • Modifiability (modyfikowalność) • Testability (testowalność)
Portability (przenośność)	<ul style="list-style-type: none"> • Adaptability (łatwość adaptacji) • Installability (łatwość instalowania) • Replaceability (zastępowalność)

Rysunek 4. Modele jakości ISO/IEC 25010

Źródło: opracowanie własne na podstawie: *ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*.

Modyfikacje w stosunku do modelu z 2001 r. są stosunkowo znaczne, szczególnie jeśli porównamy to z wielkością zmian dokonanych w 2001 r. Przede wszystkim do pozostawionych bez zmian sześciu atrybutów jakościowych zostały dołączone dwa dodatkowe – „ochrona” i „zgodność”, przy czym „zgodność” jest zupełnie nowym atrybutem, natomiast „ochrona” była poprzednio

podcharakterystyką przypisaną do funkcjonalności. Poniżej zostały opisane szczegółowe zmiany dotyczące wszystkich atrybutów jakości.

Ogólną zasadą odnoszącą się do wszystkich sześciu atrybutów jakości jest usunięcie podcharakterystyk: „zgodność” (odpowiednio: funkcjonalna, niezawodnościowa, użytecznościowa, wydajnościowa, konserwacyjna, przenośnościowa).

- Sama nazwa atrybutu „funkcjonalność” została zastąpiona lepiej odzwierciedlającą sytuację nazwą „przydatność funkcjonalna”. Podcharakterystyki „interoperacyjność” i „ochrona” zostały przesunięte w inne miejsca. „Dokładność” została teraz nazwana „poprawnością funkcjonalną”, a nazwę atrybutu „przydatność” zamieniono na „adekwatność funkcjonalną”. Dodano nową podcharakterystykę – „kompletność funkcjonalna”.
- Nazwę atrybutu „wydajność” zmieniono na „efektywność wykonania”. Dodano nową podcharakterystykę – „przepustowość”.
- „Zgodność” została dodana jako nowa charakterystyka. Utworzono ją, przenosząc do niej podcharakterystyki „współistnienie” (z „przenośności”) oraz „interoperacyjność” (z „funkcjonalności”).
- Do „użyteczności” dodano nowe podcharakterystyki – „ochrona przed błędami użytkownika” oraz „dostępność”. Nazwę podcharakterystyki „rozumiałość” zmieniono na „właściwą rozpoznawalność”, a „atrakcyjność” na „estetykę interfejsu”.
- Atrybut „niezawodność” został uzupełniony o podcharakterystykę „gotowość”.
- „Ochrona” jest nowym atrybutem. Składają się na nią podcharakterystyki: „poufność”, „integralność”, „niezaprzeczalność”, „rozliczalność” i „autentykacja”.
- „Łatwość konserwacji” została wzbogacona o „modularność” i „możliwość powtórnego użycia”, a „zmienialność” i „stabilność” zostały zastąpione przez jedną podcharakterystykę – „modyfikowalność”.
- Z „przenośności” do „zgodności” przesunięto „współistnienie”.

Podobnie jak wersja z 2001 r., ISO/IEC 25010 zawiera znacznie teraz rozbudowany model jakości użytkowej. Nie jest on jednak przedmiotem analizy dokonanej w tym artykule.

5. Dyskusja na temat modelu jakości ISO/IEC 25010

Seria norm 25000 stanowi kolejne ogniwo w łańcuchu ewolucyjnym norm jakości produktu programowego. Zdaniem autora trudno powiedzieć, że ewolucja

ta zmierza we właściwym kierunku. Głównym zarzutem jest obszerność standardu. W ciągu 20 lat z normy o objętości 13 stron przekształciła się w serię 21 norm (a będzie jeszcze więcej), każda kilkudziesięciostronicowa i w cenie jednostkowej wynoszącej ponad 100 CHF. Trudno przypuszczać, by osoby potencjalnie zainteresowane zaleceniami standardu chciały, po pierwsze, zakupić tak duży zestaw norm, a po drugie – przeanalizować i zastosować w praktyce przedstawione w nich zapisy.

W odniesieniu do konkretnej normy, w której został przedstawiony będący w centrum dokonywanej analizy model jakości, czyli normy ISO/IEC 25010, można wysunąć zarzut podobnej natury. Zdaniem autora model jest nazbyt obszerny – 8 charakterystyk i 31 podcharakterystyk w modelu jakości produktu (plus dodatkowo 5 charakterystyk i 11 podcharakterystyk w modelu jakości użytkowej) to o wiele za dużo. Co więcej, podcharakterystyk jest tak wiele, że niektórych nie da się odróżnić (praktycznie są to synonimy). Kolejne zastrzeżenia można mieć odnośnie do mierzalności niektórych podcharakterystyk. Na przykład: jak zmierzyć „estetykę interfejsu”? Zdziwienie może budzić wprowadzenie do modelu jakości podcharakterystyki „modularność”. Programy modularyzuje się po to, by osiągnąć różne inne atrybuty jakości, natomiast nie powinien to być cel sam w sobie.

Natomiast pomimo zasygnalizowanych wyżej niedoskonałości, samo istnienie modelu jakości, i to bez względu na to, jak bardzo jest rozbudowany, jest bezsprzecznie korzystne. Przede wszystkim sama lista charakterystyk i podcharakterystyk może służyć jako lista kontrolna wykorzystywana podczas zawierania kontraktu i podpisywania umowy na dostawę oprogramowania, a następnie podczas dokonywania odbioru zamówionego produktu. I wtedy, podczas rozliczania wykonawcy z tego, czy udało mu się osiągnąć założony cel, istnienie listy kontrolnej, do której można się odwołać, jest nie do przecenienia. Oczywiście, nie da się opracować listy kompletnej i z pewnością nie jest taką model ISO/IEC 25010. O wiele bardziej rozbudowany model, zawierający jeszcze większą liczbę podcharakterystyk, został przedstawiony w innej pracy autora niniejszego artykułu⁵. Ale nawet przy istnieniu i powszechnym zaakceptowaniu jakiegось modelu zawsze będą się toczyły dyskusje dotyczące tego, gdzie powinny być przypisane poszczególne podcharakterystyki – każda lista będzie budzić kontrowersje.

Jeszcze jedna, dodatkowa obawa dotyczy polskojęzycznej wersji normy. W odróżnieniu od poprzednich norm jakości oprogramowania z lat 1991 i 2001,

⁵ A. Kobyliński, op.cit.

cztery ze standardów z grupy SQuaRE zostały przetłumaczone na język polski. Ale niepokojące jest to, że od 2010r. prace nad tłumaczeniem norm na język polski całkowicie zamarły. I, co gorsza, nie przetłumaczono na język polski najważniejszego standardu z całej serii – ISO/IEC 25010.

6. Podsumowanie

Modele jakości oprogramowania opisane standardami międzynarodowymi mają już ponad 20-letnią historię. Niestety, wydaje się, że ewolucja norm zmierzła w niewłaściwym kierunku. Rozrost samego modelu jakości, jak również obudowanie modelu licznymi normami uzupełniającymi spowodowały, zdaniem autora, zaciemnienie (zamazanie) idei leżącej u podstaw myśli twórców pierwszej wersji normy z 1991 r.: opracowanie modelu jakości, który mógłby stanowić podstawę do specyfikowania atrybutów jakościowych oprogramowania, a następnie rozliczania wykonawców ze zrealizowania (lub niezrealizowania) celów jakościowych. Pomimo wypunktowanych wad modelu samo jego istnienie jest nie do przecenienia.

Zaniepokojenie może budzić nikła świadomość istnienia standardowego modelu jakości oprogramowania zarówno w środowisku osób wyłącznie wykorzystujących oprogramowania (co jeszcze można by uznać za usprawiedliwione), jak i w środowisku ich wytwórców. Autor artykułu wielokrotnie prowadził nieformalne ankiety dotyczące wiedzy o samym istnieniu norm jakościowych. Świadomość istnienia tego rodzaju norm dotyczy jedynie kilku procent osób.

Bibliografia

- ISO/IEC 9126:1991, Software engineering – Product quality.*
ISO/IEC 9126-1:2001, Software engineering – Product quality – Part 1: Quality model.
ISO/IEC TR 9126-2:2003, Software engineering – Product quality – Part 2: External metrics.
ISO/IEC TR 9126-3:2003, Software engineering – Product quality – Part 3: Internal metrics.
ISO/IEC TR 9126-4:2004, Software engineering – Product quality – Part 4: Quality in use metrics.

IOS/IEC 14598–1..6:1998..2001, Information technology – Software product evaluation.
ISO/IEC 25010:2011, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.

Kobyliński A., *Modele jakości produktów i procesów programowych*, Oficyna Wydawnicza SGH, Warszawa 2005.

SPICE: The Theory and Practice of Software Process Improvement and Capability Determination, red. K. El Emam, J.-N. Drouin, W. Melo, IEEE Computer Society, Los Alamitos 1998.

* * *

Software product quality standards evolution

Summary: The paper presents the evolution of software quality standards, starting from the first quality models developed by individual researchers. The general structure of software quality models has been presented. The first standard quality model ISO/IEC 9126:1991 has been described, then its modifications from 2001, ending with contemporary quality model described in ISO/IEC 25010:2011. The differences between subsequent versions of the model have been indicated. The disadvantages of the evolution's direction have been discussed.

Keywords: software quality model, ISO/IEC 9126, ISO/IEC 25010