

ISO/IEC 9126 – ANALIZA MODELU JAKOŚCI PRODUKTÓW PROGRAMOWYCH

Andrzej Kobyliński
Katedra Informatyki Gospodarczej
Szkoła Główna Handlowa w Warszawie
andrzej.kobylinski@sgh.waw.pl

Streszczenie

W minionym ćwierćwieczu powstało kilka modeli jakości technicznej oprogramowania. Wyróżniają one najistotniejsze atrybuty jakości (charakterystyki), atrybuty podrzędne (podcharakterystyki) i miary (metryki) oprogramowania. W pracy przedstawiono ogólną budowę modeli jakości oprogramowania i zaprezentowano historię powstania i budowę modelu opisanego w normie międzynarodowej ISO/IEC 9126.

Słowa kluczowe: *ISO/IEC 9126, ocena oprogramowania, model oceny jakości.*

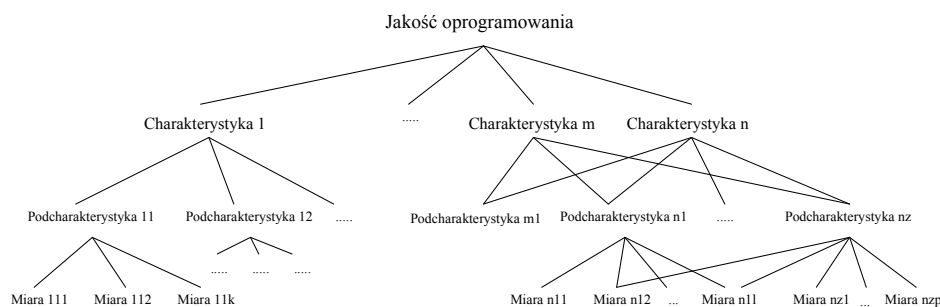
1. Ogólna budowa modeli jakości

W okresie powszechnej nadprodukcji wszelkich dóbr, coraz większego znaczenia nabiera problematyka jakości. Coraz trudniej jest sprzedać produkt słaby jakościowo, a jeśli już jest to możliwe, to następuje kosztem dużych ustępstw cenowych.

Jakość produktu składającego się z wielu składowych zwykle odbierana jest przez nabywcę poprzez pryzmat najsłabszego jakościowo elementu. Niestety, nader często, za taki element uważane jest oprogramowanie. Wprawdzie obiektywna ocena jakości jest bardzo trudna, a może nawet niemożliwa do przeprowadzenia, gdyż każdy z odbiorców produktu programowego może zwracać uwagę na inne aspekty jakości, niemniej można próbować wyróżnić najważniejsze czynniki jakości oprogramowania.

Próby stworzenia modelu tzw. jakości technicznej, czyli możliwie abstrahującej od subiektywnej opinii użytkowników (patrz np. [Koby02] s. 534-5, 538), trwały od drugiej połowy lat 70. ubiegłego stulecia. Wszystkie zaproponowane modele mają zbliżoną budowę. Wyróżnia się w nich zbiór podstawowych atrybutów (charakterystyk) jakości. Są to atrybuty bardzo ogólne, abstrakcyjne, wieloznaczne, możliwe do przedstawienia wyłącznie w sposób opisowy. Ponieważ techniczna definicja jakości zakłada obiektywizm, którego nie da się uzyskać, przedstawiając wartości atrybutów w sposób opisowy, powszechnie stosuje się metodę podziału atrybutów głównych na atrybuty szczegółowe (podcharaktery-

styki). Na każdy atrybut główny składa się kilka atrybutów szczegółowych. Z kolei każdemu atrybutowi szczegółowemu można przypisać jedną lub kilka miar (metryk), którymi w wymierny sposób można określić moc osiągnięcia poszczególnych atrybutów szczegółowych. Typowy schemat modelu jakości technicznej przedstawiony został na rys. 1.



Rys. 1. Ogólna postać modeli jakości technicznej

Źródło: Opracowanie własne

Schematy poszczególnych modeli różnią się nieznacznie. Wszystkie wyróżniają charakterystyki i podcharakterystyki, jak również sugerują (choć nie zawsze wprost ujawniają) istnienie miar. Praktycznie jedyną odmianą w modelach jest to, że w niektórych modelach podcharakterystyki przypisane są jednoznacznie do charakterystyk - struktura hierarchiczna (na rys. 1 sytuację taką obrazuje *Charakterystyka 1* i jej podcharakterystyki); w innych ta sama podcharakterystyka może być przypisana do kilku charakterystyk - struktura sieciowa (na rys. 1 sytuację taką obrazują *Charakterystyka m* oraz *Charakterystyka n* i ich podcharakterystyki).

Dwa pierwsze modele jakości oprogramowania powstały w końcu lat 70. Pierwszym był model opracowany przez zespół badawczy kierowany przez McCalla [McCa77]; drugim - model Boehma z zespołem [BBKL+78]. Wspólną cechą obu tych modeli jest to, że próbują opisać atrybuty istotne dla użytkownika/wytwórcy oprogramowania w trzech wyróżnionych okresach: w czasie operacyjnego wykorzystywania produktu (atrybuty istotne dla użytkownika), w czasie konserwowania produktu (atrybuty istotne dla programisty konserwującego) oraz w czasie przekształcania produktu w nowy produkt (atrybuty istotne dla programisty dokonującego transferu). W modelu McCalla wyróżnionych zostało 11 atrybutów głównych i 25 szczegółowych; w modelu Boehma wyróżniono 8 charakterystyk i 12 podcharakterystyk.

W późniejszym okresie powstało jeszcze co najmniej kilka modeli: Boeinga [DeWi88], FURPS (Hewlett-Packard) [FiHi98], CUPRIMDSO (IBM) [FiHi98].

2. Geneza modelu ISO/IEC 9126

W praktyce wytwarzania oprogramowania za podstawowy, a często jedyny, atrybut jakości uważana była niezawodność. Skonstruowanie prawie w tym samym czasie kilku odmiennych modeli jakości nie było szczęśliwą okolicznością, gdyż sugerowało stosowanie odmiennych atrybutów podczas dokonywania oceny jakości. Potrzeba powstania jednego, uniwersalnego modelu jakości oprogramowania dostrzeżona została dość wcześnie, w 1985 r wspólny komitet techniczny International Organisation for Standardisation (ISO) i International Electrotechnical Commission (IEC) rozpoczął prace nad opracowaniem normy, określającej jednolity, powszechnie obowiązujący model jakości oprogramowania. Norma została oficjalnie opublikowana w 1991 r pod numerem ISO/IEC 9126. Była ona bardzo krótka: część normatywna liczyła zaledwie kilka stron i wymienionych w niej było wyłącznie 6 charakterystyk jakości. Podcharakterystyki umieszczone zostały w załączniku i nie miały charakteru normatywnego, a wyłącznie informacyjny. Brak w niej było jakichkolwiek odwołań do miar oprogramowania. Wkrótce rozpoczęły się prace nad modyfikacją normy. Ich celem była rozbudowa tak lapidarnego do tej pory standardu. Nowa norma nosi ten sam numer główny i nazwę: *Inżynieria oprogramowania – Jakość produktu (Software engineering - Product quality)* i w założeniu miała się składać z 4 części:

- Część 1: Model jakości (*Quality model*).
- Część 2: Miary zewnętrzne (*External metrics*).
- Część 3: Miary wewnętrzne (*Internal metrics*).
- Część 4: Miary jakości użytkowej (*Quality in use metrics*).

Część 1 opublikowana została w 2001 r i stała się normą obowiązującą [ISO9126], natomiast pozostałe części nie zostały do tej pory opublikowane. Można mieć nadzieję, że części te zostaną upublicznione w przyszłości, gdyż w części 1 często następują odwołania do 3 pozostałych części.

Omawiana norma ISO/IEC 9126-1:2001 stanowi rewizję starej normy ISO/IEC 9126:1991. Różnice między nimi są stosunkowo nieznaczne. Wszystkie 6 charakterystyk występujących w starej edycji znalazło się w nowej; podcharakterystyki, które w starym wydaniu były w załączniku i miały charakter informacyjny, zostały przeniesione do części normatywnej; tak więc obecnie norma zawiera prawie pełny model, składający się z charakterystyk i podcharakterystyk (brakuje jeszcze miar). Równocześnie utworzona została nowa norma ISO/IEC 14598 [ISO14598], która opisuje proces dokonywania oceny produktu programowego.

Norma specyfikuje 6 charakterystyk jakości (funkcjonalność, niezawodność, użyteczność, efektywność, konserwowalność, przenośność), podcharakterystyki przypisane poszczególnym charakterystykom i odwołuje się do zewnętrznych

oraz wewnętrznych miar oprogramowania, które znajdują się w 2 i 3 części normy. Model jest uniwersalny i daje się zastosować do dowolnego typu oprogramowania, w tym również do oprogramowania typu firmware.

Zdefiniowany model jakości może być użyty do:

- „walidacji kompletności definicji wymagań,
- identyfikacji wymagań oprogramowania,
- identyfikacji celów projektu oprogramowania,
- identyfikacji celów testowania oprogramowania,
- identyfikacji kryteriów zapewniania jakości,
- identyfikacji kryteriów akceptacji produktu przez użytkownika po zakończeniu prac nad produktem.” [ISO9126 s. 1]

Omawiana norma ISO/IEC 9126:2001 odwołuje się i jest zgodna z wieloma normami z zakresu jakości, niezawodności, wymagań ergonomii, cyklu życia oprogramowania itd.

3. Opis normy ISO/IEC 9126-1:2001

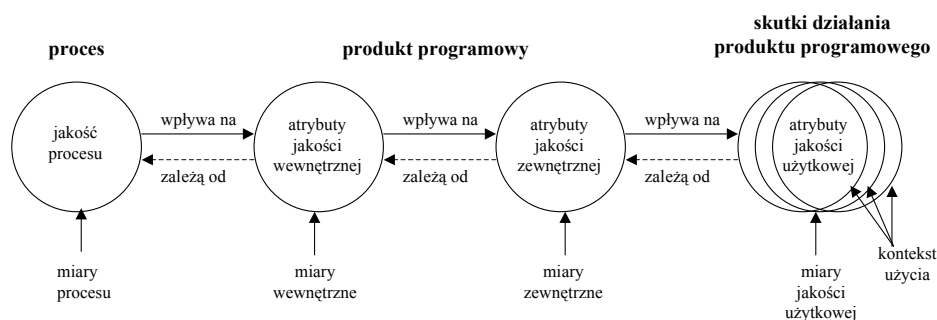
Norma składa się z 7 rozdziałów. Cztery początkowe mają charakter formalny (opisują zakres normy, powołania, zgodność z innymi normami, definicje), natomiast znaczenie merytoryczne mają rozdziały 5-7.

Rozdział 5. Związki jakości

W rozdziale tym przedstawione zostały różne możliwe podejścia do jakości: jakość wewnętrzną (ang. *internal quality* – określaną charakterystykami wewnętrznymi produktu programowego, odnoszącymi się głównie do kodu źródłowego), jakość zewnętrzną (ang. *external quality* – określaną przez charakterystyki zewnętrzne produktu programowego), jakość użytkową (ang. *quality in use* - odczuwaną przez użytkownika, gdy produkt pracuje w środowisku docelowym, mierzona przez satysfakcję użytkowników i wzrost efektywności jego pracy). Bardzo cenne jest to, że twórcy normy dostrzegli, że jakość nie jest pojęciem statycznym, ale ewoluuje w trakcie postępu prac nad produktem, a jakość docelowa nie jest niemożliwą praktycznie do osiągnięcia jakością idealną, ale jakością dostateczną – konieczną i wystarczającą w danych warunkach ([Koby03]).

W normie stwierdza się, że dobry jakościowo produkt (którego jakość mierzy się zgodnie z normą ISO/IEC 9126) jest efektem realizacji dobrych jakościowo procesów wytwórczych (które z kolei można ocenić za pomocą poziomu CMM lub ISO/IEC 15504 [Koby00]), a jakość produktu ma wpływ na jakość użytkową produktu ([Siko00]). Dlatego ocena i poprawa procesu jest środkiem do poprawy

jakości produktu, a ocena i poprawa jakości produktu jest środkiem do poprawy jakości użytkowej. Sytuację tę obrazuje rys. 2.



Rys. 2. Jakość w cyklu życia

Źródło: [ISO9126 s. 3]

Jakość produktu programowego może być mierzona przy pomocy miar wewnętrznych (zwykle przy pomocy statycznych miar kodu źródłowego programu) lub miar zewnętrznych (zwykle jako zachowanie się programu w czasie jego wykonywania).

Biorąc pod uwagę proces wytwarzania oprogramowania, we wczesnych fazach rozwoju można dokonywać wyłącznie oceny dostępnych zasobów oraz procesów wytwórczych. W miarę postępu prac, gdy dostępne stają się produkty pośrednie (specyfikacja, projekt, kod źródłowy), można ocenić ich jakość za pomocą miar wewnętrznych. Miary te można zastosować w celu przewidywania wartości miar zewnętrznych. Założenie to jest bardzo dyskusyjne, podobnie zresztą jak to, że usprawnione procesy programowe prowadzą do uzyskania lepszego jakościowo produktu [KoCh02]. Jakość gotowego produktu mierzona jest przy pomocy miar zewnętrznych. Przy ich pomocy można zmierzyć zachowanie się produktu w jego otoczeniu, pozwalają one ocenić jakość oprogramowania w trakcie operacyjnego wykorzystywania. Do oceny produktu z punktu widzenia końcowego użytkownika służą miary jakości użytkowej. W normie zwraca się uwagę na to, że istnieje różnica pomiędzy oceną jakości produktu programowego a oceną systemu, którego tylko jednym z elementów jest oprogramowanie.

Rozdział 6. Model jakości dla jakości zewnętrznej i wewnętrznej

Rozdział 6 zawiera najważniejszy element normy: model jakości produktu programowego. Model jakości zdefiniowany został jako zbiór charakterystyk i związków pomiędzy nimi, które stanowią podstawę do specyfikacji wymagań jakościowych i oceny jakości gotowego produktu. Rys. 3 przedstawia strukturę

modelu. Na rysunku tym nie zostało to zaznaczone, ale zakłada się, że ocena każdej z 6 głównych charakterystyk wymaga zgodności z innymi, w normie nie wymienionymi, zasadami i normami, dotyczącymi odpowiednio: funkcjonalności, niezawodności itd.



Rysunek 3. Jakość produktu programowego (Źródło: [ISO9126 s. 7])

Wspomniano już, że jakość produktu programowego jest czym innym, niż jakość użytkowa. Osiągnięcie jakości użytkowej zależy od spełnienia wymagań nałożonych przez miary zewnętrzne odpowiednich podcharakterystyk, które z kolei zależą od osiągnięcia odpowiednich wartości właściwych miar wewnętrznych. Zaznacza się jednak, że ponieważ jakość użytkowa jest oceną subiektywną (w oczach odbiorcy produktu), to spełnienie założonych wartości miar wewnętrznych nie jest wystarczające do tego, by osiągnąć odpowiednie wartości miar zewnętrznych; a osiągnięcie odpowiednich wartości miar wewnętrznych niekoniecznie skutkuje osiągnięciem odpowiedniej jakości użytkowej.

W przypadku dużego produktu programowego nie jest możliwy pomiar (czy to wewnętrzny, czy też zewnętrzny) wszystkich części produktu pod kątem wszystkich podcharakterystyk wyszczególnionych w modelu. Dlatego zaleca się, by dokonywać pomiaru wyłącznie wyróżnionych, najistotniejszych pod względem biznesowym, części produktu.

W rozdziale 6 zdefiniowane zostały ponadto charakterystyki i podcharakterystyki modelu jakości, którego struktura przedstawiona jest na rys. 3. W tab. 1 zaprezentowano skrócone definicje wszystkich 6-ciu charakterystyk oraz, z braku miejsca, zaledwie czterech przykładowych podcharakterystyk.

Tabela 1.

Definicje charakterystyk i podcharakterystyk jakości

Charakterystyka (ang. <i>characteristic</i>)	Zdolność oprogramowania do ...
1. funkcjonalność (ang. <i>functionality</i>)	... udostępnienia funkcji wymaganych przez użytkownika, przy wykorzystywaniu oprogramowania w ustalonych warunkach.
1.1 dopasowanie (ang. <i>suitability</i>)	... dostarczania odpowiedniego zbioru funkcji.
1.2 zgodność (ang. <i>accuracy</i>)	... dostarczania właściwych i uzgodnionych wyników.
1.3 łatwość współdziałania (ang. <i>interoperability</i>)	... współdziałania z innymi, ustalonymi systemami.
1.4 zabezpieczenie (ang. <i>security</i>)	... zapobiegania nieupoważnionemu dostępowi i atakom zewnętrznym.
2. niezawodność (ang. <i>reliability</i>)	... wykonywania przewidzianych zadań w ustalonych warunkach.
3. użyteczność (ang. <i>usability</i>)	... zrozumiałości, łatwości nauki, łatwości użycia.
4. efektywność (ang. <i>efficiency</i>)	... dostarczenia żądanej wydajności.
5. konserwowalność (ang. <i>maintainability</i>)	... dokonywania w nim modyfikacji, polegających na poprawianiu błędów, udoskonalaniu produktu, dostosowywaniu do zmieniających się warunków zewn.
6. przenośność (ang. <i>portability</i>)	... przekształcenia go do użycia w innym środowisku (organizacyjnym, sprzętowym, systemowym).

Źródło: opracowanie własne na podstawie [ISO9126]

Rozdział 7. Model jakości dla jakości użytkowej

Rozdział ten zawiera model jakości, który powinien być używany, gdy chce się ocenić jakość użytkową. Zasadą powinno być, że mierzy się przy tym rezultaty stosowania oprogramowania, a nie charakterystyki samego oprogramowania. Model zawiera zaledwie 4 atrybuty. Schemat modelu jakości użytkowej przedstawia rys. 4, a definicje charakterystyk zawiera tab. 2.



Rys. 4. Model jakości dla jakości użytkowej

Źródło: [ISO9126 s. 12]

Tabela 2.

Definicje charakterystyk jakości użytkowej

Charakterystyka (ang. <i>characteristic</i>)	Definicja
1. efektywność (ang. <i>effectiveness</i>)	Zdolność oprogramowania do pełnego osiągnięcia założonych celów z zakładaną dokładnością.
2. produktywność (ang. <i>productivity</i>)	Relacja zużytych zasobów do osiągniętej efektywności.
3. bezpieczeństwo (ang. <i>safety</i>)	Zdolność oprogramowania do unikania awarii, powodującej katastrofę w środowisku jego pracy.
4. satysfakcja (ang. <i>satisfaction</i>)	Zadowolenie klienta w czasie używania produktu.

Źródło: opracowanie własne na podstawie [ISO9126]

Dodatki

Relacje między charakterystykami a miarami, jak również między miarami wewnętrznymi a zewnętrznymi, zawarte są w dodatku A. Co ciekawe, dodatek ten ma charakter normatywny. Dodatek B (informacyjny) zawiera użyte w normie definicje pobrane z innych norm.

4. Zakończenie

Nie ulega wątpliwości, że w ostatnich latach do oceny jakości procesów programowych przykładana jest o wiele większa waga, niż do oceny produktów programowych. Świadczy o tym chociażby fakt, że w Polsce wszystkie znaczące firmy wytwarzające oprogramowanie uzyskały certyfikat systemu zarządzania jakością ISO 9001, kilka zdecydowało się na ocenę CMM, natomiast krótka norma dotycząca oceny jakości produktów programowych – ISO/IEC 9126, mimo że powstała 12 lat temu, nie została nawet przetłumaczona na język polski. Sytuacja wydaje się być niezrozumiała, chociaż istnieją argumenty za takim postępowaniem [KoCh02]. Można jednak przypuszczać, że w niedalekiej przyszłości sytuacja musi ulec zmianie – w rzeczywistości dla odbiorcy oprogramowania wcale nie jest istotne, jak dojrzałe i ustabilizowane procesy realizuje wykonawca, natomiast żywotne znaczenie ma jakość nabywanego produktu programowego.

Wyjaśnienia wymaga jeszcze kwestia właściwych polskich tłumaczeń poszczególnych atrybutów. W literaturze polskojęzycznej brak jest ustalonych i dobrze brzmiących określeń z tego zakresu. Dlatego też, z konieczności, niektóre pojęcia stanowią autorską propozycję wprowadzenia do polskiego piśmiennictwa określeń z tego obszaru.

Literatura

- [BBKL+78] Boehm B., Brown J.R., Kaspar H., Lipow M., MacLeod G. J., Merritt M. J.: Characteristics of Software Quality, Amsterdam, 1978.
- [DeWi88] Deutsch M.S., Willis R.R., Software Quality Engineering: A Total Technical and Management Approach, Prentice-Hall, 1988.
- [FiHi98] Fitzpatrick R., Higgins C., Usable Software and its Attributes: A Synthesis of Software Quality, European Community Law and Human-Computer Interaction, http://ganymede2.kst.dit.ie/rfitzpatrick/Papers/hci98_25.pdf
- [ISO9126] ISO/IEC 9126:2001 Software engineering - Product Quality, Part 1: Quality model, ISO, Geneva 2001.
- [ISO14598] Information technology – Software product evaluation, Parts 1-6, ISO, Geneva, 1998-2001.
- [Koby00] Kobyliński A., „Standardy oceny i doskonalenia procesów produkcji oprogramowania - próba porównania”, w: Systemy Wspomagania Organizacji 2000, Wydawnictwo AE w Katowicach, Katowice 2000, str. 155-171.
- [Koby02] Kobyliński A., „Definiowanie jakości produktu programowego - podejście ewolucyjne”, w: Nowoczesne technologie informacyjne w zarządzaniu, red. Niedzielska E., Dudycz H., Dyczkowski M., Prace Naukowe Akademii Ekonomicznej we Wrocławiu nr 995, Wrocław 2002.
- [KoCh02] Kobyliński A., Chodoła T., „Produkty i procesy programowe - dwa punkty widzenia na ocenę jakości oprogramowania”, w: Systemy Wspomagania Organizacji 2002, Wydawnictwo AE w Katowicach, Katowice 2002, str. 149-156.
- [Koby03] Kobyliński A.: „Dostatecznie dobra jakość produktów programowych”, referat przygotowywany na konferencję NTIZ '03, Karpacz, 2003 (w druku).
- [McCa77] McCall i in.: Factors in Software Quality, 1-3, RAD-TR-77-369, US Rome Air Development Center, Griffiss Air Force Base, NY 13441-5700, November 1977.
- [Siko00] Sikorski M.: Zarządzanie jakością użytkową w przedsiębiorstwach informatycznych, Monografie 17, Wydawnictwo Politechniki Gdańskiej, Gdańsk 2000.

ISO/IEC 9126 – SOFTWARE PRODUCT QUALITY MODEL ANALYSIS

A few software technical quality models were established last 25 years. All of them distinguish the most important software quality attributes (characteristics), sub-attributes (sub-characteristics) and measures (metrics). The general construction of the software quality models had been presented. The history and the structure of international standard ISO/IEC 9126 has been shown.

Key words: *ISO/IEC 9126, software product assessment, quality assessment model.*