

**Говорущенко Т.О.**

Хмельницький національний університет

**Медзатий Д.М.**

Хмельницький національний університет

**Боднар М.А.**

Хмельницький національний університет

## КОМПЕНСАТОРНА МОДЕЛЬ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МОДЕЛІ SQUARE

*У статті розроблено компенсаторну модель якості програмного забезпечення на основі моделі SQuaRE, яка забезпечує можливість надання коректних кількісних оцінок якості, враховуючи важливість та взаємовпливи атрибутів якості ПЗ. Розроблена модель дає змогу зробити висновок: наявність яких атрибутів є пріоритетною та необхідною для забезпечення відповідного рівня достовірності оцінювання якості ПЗ.*

**Ключові слова:** програмне забезпечення (ПЗ), якість ПЗ, характеристики якості, атрибути якості, компенсаторна модель.

**Постановка проблеми.** Ключовим фактором забезпечення ефективного застосування програмного забезпечення (ПЗ) та однією з основних вимог користувачів і зацікавлених осіб до сучасного ПЗ є досягнення високих значень показників його якості. Якість ПЗ є основним чинником для його успішного впровадження та експлуатації. Потреба у забезпеченні якості ПЗ випливає з того, що помилки та відмови ПЗ загрожують катастрофами, які призводять до людських жертв, екологічних катаклізмів, значних часових втрат та фінансових збитків [1].

Загальний підхід до моделювання якості ПЗ полягає в тому, щоб спочатку ідентифікувати невеликий набір характеристик якості найвищого рівня абстракції, а потім у напрямку «згоризонтизу» розбити ці характеристики на підхарактеристики та набори підлеглих атрибутів. Стандарт ISO 25010 [2], який описує модель якості SQuaRE, є типовим прикладом такого загального підходу та найбільш використовуваною моделлю для оцінки якості ПЗ.

Під час оцінювання якості ПЗ для усунення проблеми суб'єктивного оцінювання та формального задоволення якості варто враховувати як можливість обчислення та ступінь вираженості характеристик, підхарактеристик та атрибутів якості, так і їх значущість.

Якщо атрибути, які входять до складу декількох підхарактеристик та (або) характеристик якості, визначені неточно або відсутні, то одно-

часне використання цих атрибутів суттєво вплине на достовірність отриманих оцінок якості ПЗ. За такої ситуації важливою є умова пом'якшення впливу взаємної кореляції таких характеристик та підхарактеристик під час використання їх у моделі якості. Таке пом'якшення здійснюється шляхом виявлення спільних атрибутів, забезпечення їх наявності, підвищення точності їх значень або за можливості обмеження одночасного задіявання наборів підхарактеристик, що містять однакові атрибути.

Отже, нині актуальною є проблема розроблення багатоатрибутної компенсаторної моделі якості ПЗ, яка враховує важливість кожного атрибута та показує ступінь впливу атрибута на якість кінцевого продукту.

**Аналіз останніх досліджень і публікацій.** Модель якості, створена в межах стандарту [2], визначається 8 загальними характеристиками якості продукту: функційна придатність, ефективність, сумісність, зручність використання, надійність, захищеність, супроводжуваність або ремонтпридатність, можливість переносу. Кожна характеристика якості ПЗ є функцією від декількох підхарактеристик якості (всього 31 підхарактеристика, згідно зі стандартом [2]). Нижній рівень ієрархії представляють атрибути якості ПЗ, які підлягають точному опису та вимірюванню. Атрибути якості ПЗ визначені та описані у стандарті ISO 25023 [3]. Аналіз робіт [4–7] та стандарту [3] дав можливість визначити залеж-

ність підхарактеристик якості від 203 атрибутів. Викладена концепція оцінювання якості ПЗ представлена на рис. 1.



Рис. 1. Сучасна концепція оцінювання якості ПЗ у рамках моделі SQuaRE (ISO 25010)

Натепер оцінювання атрибутів якості ПЗ виконується суб'єктивно, оскільки відсутні єдині стандарти щодо їх оцінювання. Інтерпретація величин атрибутів також здійснюється суб'єктивно, оскільки відсутні стандартизовані «еталонні» значення атрибутів. Градування шкали оцінки знову-таки суб'єктивне, тому що залежить від можливих ступенів відповідності атрибуту накладеним обмеженням, а ступені відповідності не стандартизовані і визначаються софтверною організацією [8–11].

Отже, оцінювання якості ПЗ як функції основних восьми характеристик є суб'єктивним, оскільки софтверна організація інтерпретує одержані значення атрибутів як максимальні, градуєє шкалу оцінки кожної характеристики, виходячи з власної інтерпретації значень атрибутів та можливих ступенів відповідності атрибутів обмеженням, у результаті чого одержує максимальні значення кожної характеристики, а, відповідно, і максимальне значення якості ПЗ [8–11]. Насправді ж відбувається лише формальне задоволення якості ПЗ унаслідок неповного покриття стандартами об'єктів стандартизації, а також унаслідок вибору розробником вигідних для себе стандартів та пристосування цих стандартів до своїх потреб.

Основна ідея моделі SQuaRE [2] полягає в тому, що оцінювання якості ПЗ, а також її характеристик та підхарактеристик має проводитись комплексно, з урахуванням усіх зазначених характеристик, підхарактеристик та атрибутів відповідно. Але відсутні комплексні методології, які дадуть змогу оцінити не лише вплив кожної окремої характеристики на якість ПЗ (цьому питанню присвячена низка робіт), але й забезпечать можливість оцінити наявність усіх атрибутів, необ-

хідних для визначення всіх підхарактеристик та характеристик якості (достатність інформації), а також оцінити взаємовплив характеристик.

Аналіз робіт [4–7] та стандартів ISO 25010 [2], ISO 25023 [3] дав можливість зробити висновок, що є атрибути, від яких залежать більше однієї підхарактеристики та характеристики якості ПЗ, тобто має місце кореляція підхарактеристик та характеристик за певними атрибутами (так, згідно зі стандартом [3] підхарактеристики якості залежать від 203 атрибутів, але всього від 138 різних атрибутів). Існування взаємозв'язків (кореляції) між характеристиками та підхарактеристиками за атрибутами впливає на значущість та вагу атрибутів якості ПЗ [12]. Однією з проблем моделей якості натепер є саме визначення значущості атрибутів якості.

З огляду на потребу врахування значущості та ваги атрибутів під час оцінювання якості ПЗ, варто для моделювання якості ПЗ використати компенсаторну модель. Компенсаторна модель – це багатоатрибутна модель процесу прийняття рішень на етапі оцінювання альтернатив, в якій один атрибут або група атрибутів є компенсуючими для іншого атрибута або групи атрибутів у загальній оцінці об'єкта або ідеї [13]. Компенсаторними такі моделі названі тому, що низька оцінка, одержана для одного атрибута, може бути компенсована високим балом за іншим атрибутом. В основі дії компенсаторних моделей лежить оцінка ступеня впливу атрибутів на кінцеву властивість (якість ПЗ) з урахуванням значущості цих атрибутів. Однією з відомих компенсаторних моделей є модель сумарної оцінки за атрибутами з урахуванням їх значущості, у якій кінцева властивість оцінюється з урахуванням ступеня вираженості та значущості атрибута [14]. Труднощами під час розроблення компенсаторної моделі є саме визначення впливовості та значущості атрибутів якості.

У [15] розроблено метод оцінювання вагових коефіцієнтів атрибутів якості ПЗ, який складається з таких етапів: 1) визначення спільних атрибутів для характеристик якості ПЗ; 2) визначення спільних атрибутів для підхарактеристик якості ПЗ; 3) розрахунок вагових коефіцієнтів атрибутів якості ПЗ на основі кількості підхарактеристик, які залежать від цих атрибутів – ваговий коефіцієнт  $h$ -го атрибута обчислюється за формулою:  $w_h^m = \frac{ksch_h}{km}$ , де  $ksch_h$  – кількість підхарактеристик, які залежать від  $h$ -го спільного атрибута;  $km$  – загальна кількість атрибутів (оскільки натепер підхарактеристики якості залежать від 138 різних атрибутів, то  $km = 138$ ). Згідно з розробленим методом, у [15]

проведено розрахунок вагових коефіцієнтів атрибутів якості ПЗ, які будуть використовуватись у майбутньому розробленні компенсаторної моделі якості ПЗ.

**Постановка завдання.** Проведений аналіз наявних досліджень та публікацій у галузі оцінювання якості ПЗ на основі моделі SQuaRE показав, що нині невирішеною є проблема розроблення багатоатрибутної компенсаторної моделі якості ПЗ, яка враховує важливість кожного атрибута та показує ступінь впливу атрибута на якість кінцевого продукту. Натепер проведено розрахунок вагових коефіцієнтів атрибутів якості ПЗ, визначених у стандарті ISO 25023. Отже, метою цього дослідження є розроблення компенсаторної моделі якості ПЗ на основі моделі SQuaRE.

**Виклад основного матеріалу дослідження.** Якість ПЗ ( $Q$ ) відповідно до стандарту [2] є функцією від восьми основних характеристик:  $Q = f(Fs, Pe, Ub, Rb, Cb, Scr, Mb, Pb)$ , де:  $Fs$  – функційна придатність,  $Pe$  – ефективність,  $Ub$  – зручність використання,  $Rb$  – надійність,  $Cb$  – сумісність,  $Scr$  – захищеність,  $Mb$  – супроводжуваність,  $Pb$  – можливість переносу.

Кожна з вищевказаних характеристик якості є функцією від декількох підхарактеристик якості. Наприклад, функційна придатність є функцією від функційної повноти ( $FCom$ ), функційної коректності ( $FCor$ ) та функційної доцільності ( $FAppr$ ):  $Fs = f_1(FCom, FCor, FAppr)$ . Водночас кожна підхарактеристика якості ПЗ є функцією певних атрибутів якості ПЗ, описаних у стандарті ISO 25023 [3]. Наприклад, функційна повнота є функцією від кількості функцій ( $Nof$ ), повноти функційної реалізації ( $Ficn$ ), функційної адекватності ( $Faq$ ), покриття функційної реалізації ( $Fic$ ).

Якість ПЗ може бути виражена не тільки функцією від підхарактеристик якості, але й зваженим середнім арифметичним з урахуванням значень атрибутів та коефіцієнтів їхньої вагомості. Отже, представимо компенсаторну модель якості програмного забезпечення на основі моделі SQuaRE в узагальненому теоретико-множинному вигляді:

$$Q = f(Fs, Pe, Ub, Rb, Cb, Scr, Mb, Pb) = f(f_1(FCom, FCor, FAppr), \dots) = \phi(w_i^m \cdot qms_i),$$

де  $w_i^m$  – ваговий коефіцієнт  $i$ -го атрибута якості ПЗ,  $i = 1, 138$ ,  $qms_i \in QMS$ ,  $QMS = \{qms_1, \dots, qms_{138}\}$  – множина атрибутів якості ПЗ.

Враховуючи вагові коефіцієнти атрибутів якості ПЗ, розраховані у [15], розробимо компенсаторну модель якості програмного забезпечення на основі моделі SQuaRE:

$$Q = \phi \left( \frac{11}{138} \cdot Nof, \frac{2}{138} \cdot Ficn, \frac{2}{138} \cdot Faq, \frac{17}{138} \cdot Fic, \frac{1}{138} \cdot Ot, \frac{8}{138} \cdot Nic, \frac{1}{138} \cdot Ndi, \frac{1}{138} \cdot Ca, \frac{2}{138} \cdot Pc, \frac{2}{138} \cdot Nmot, \frac{1}{138} \cdot Rt, \frac{2}{138} \cdot Noe, \frac{1}{138} \cdot Tnt, \frac{1}{138} \cdot Tskt, \frac{2}{138} \cdot Mathr, \frac{6}{138} \cdot Nofl, \frac{2}{138} \cdot Niore, \frac{1}{138} \cdot Uwt, \frac{1}{138} \cdot Nmre, \frac{1}{138} \cdot Ntre, \frac{1}{138} \cdot Tcc, \frac{1}{138} \cdot Iou, \frac{1}{138} \cdot Nolcd, \frac{1}{138} \cdot Ioll, \frac{1}{138} \cdot Mmu, \frac{1}{138} \cdot Mtu, \frac{1}{138} \cdot Mote, \frac{1}{138} \cdot Ncu, \frac{1}{138} \cdot Cbw, \frac{1}{138} \cdot Sdb, \frac{1}{138} \cdot Nott, \frac{1}{138} \cdot Niodi, \frac{1}{138} \cdot Cnd, \frac{1}{138} \cdot Fua, \frac{1}{138} \cdot Uaio, \frac{1}{138} \cdot Efl, \frac{1}{138} \cdot Hfq, \frac{1}{138} \cdot Eudhs, \frac{1}{138} \cdot Haa, \frac{1}{138} \cdot Cudhf, \frac{1}{138} \cdot Ecr, \frac{1}{138} \cdot Nsf, \frac{2}{138} \cdot Nuec, \frac{1}{138} \cdot Nac, \frac{1}{138} \cdot Nop, \frac{1}{138} \cdot Niwccd, \frac{1}{138} \cdot Nmi, \frac{2}{138} \cdot Nie, \frac{1}{138} \cdot Pha, \frac{1}{138} \cdot Neum, \frac{1}{138} \cdot Nurs, \frac{1}{138} \cdot Optdo, \frac{1}{138} \cdot Nouheo, \frac{1}{138} \cdot Niewusc, \frac{1}{138} \cdot Nacie, \frac{1}{138} \cdot Necwusc, \frac{1}{138} \cdot Tnect, \frac{1}{138} \cdot Nfuet, \frac{1}{138} \cdot Tnfric, \frac{1}{138} \cdot Tniop, \frac{1}{138} \cdot Nige, \frac{1}{138} \cdot Dipu, \frac{1}{138} \cdot Disu, \frac{1}{138} \cdot Dea, \frac{1}{138} \cdot Drwmu, \frac{1}{138} \cdot Ewscbuusd, \frac{1}{138} \cdot Ewusd, \frac{1}{138} \cdot Ffrusd, \frac{1}{138} \cdot Susd, \frac{1}{138} \cdot Ppsa, \frac{3}{138} \cdot Noft, \frac{1}{138} \cdot Ps, \frac{5}{138} \cdot Ntc, \frac{4}{138} \cdot Nrf, \frac{1}{138} \cdot Ncf, \frac{1}{138} \cdot Fdatc, \frac{1}{138} \cdot Frn, \frac{1}{138} \cdot Frl, \frac{1}{138} \cdot Mtbf, \frac{1}{138} \cdot Tmy, \frac{1}{138} \cdot Elfd, \frac{1}{138} \cdot Fdy, \frac{1}{138} \cdot Tidwsis, \frac{1}{138} \cdot Nob, \frac{1}{138} \cdot Tdt, \frac{2}{138} \cdot Nbd, \frac{3}{138} \cdot Nio, \frac{1}{138} \cdot Tir, \frac{1}{138} \cdot Dt, \frac{1}{138} \cdot Nrs, \frac{1}{138} \cdot Nrn, \frac{1}{138} \cdot Ray, \frac{1}{138} \cdot Ndfri, \frac{1}{138} \cdot Ndfbe, \frac{1}{138} \cdot Nip, \frac{1}{138} \cdot Deay, \frac{2}{138} \cdot Nidc, \frac{2}{138} \cdot Nat, \frac{2}{138} \cdot Ncr, \frac{2}{138} \cdot Aca, \frac{1}{138} \cdot Ndiced, \frac{1}{138} \cdot Ndbired, \frac{1}{138} \cdot Net, \frac{1}{138} \cdot Nepuds, \frac{1}{138} \cdot Nernrp, \frac{1}{138} \cdot Nasdrsl, \frac{1}{138} \cdot Naao, \frac{1}{138} \cdot Npam, \frac{1}{138} \cdot Nmm, \frac{1}{138} \cdot Nv, \frac{1}{138} \cdot Nm, \frac{1}{138} \cdot Fcy, \frac{1}{138} \cdot Nfcy, \frac{1}{138} \cdot Vrn, \frac{1}{138} \cdot Aay, \frac{1}{138} \cdot Tay, \frac{1}{138} \cdot Cra, \frac{2}{138} \cdot Ert, \frac{1}{138} \cdot Nirbl, \frac{1}{138} \cdot Ndfri, \frac{1}{138} \cdot Atc, \frac{1}{138} \cdot Nrv, \frac{1}{138} \cdot Ccca, \frac{1}{138} \cdot Ntcpbm, \frac{1}{138} \cdot Ntspam, \frac{1}{138} \cdot Nbtfr, \frac{1}{138} \cdot Ntdos, \frac{1}{138} \cdot Ncp, \frac{1}{138} \cdot Puf, \frac{1}{138} \cdot Nds, \frac{1}{138} \cdot Eoi, \frac{1}{138} \cdot Aads, \frac{1}{138} \cdot Hea, \frac{1}{138} \cdot Sea, \frac{1}{138} \cdot Nofitwnca, \frac{1}{138} \cdot Tnfwde, \frac{1}{138} \cdot Nso, \frac{1}{138} \cdot Nis$$

де  $Nof$  – кількість функцій,  $Ficn$  – повнота функційної реалізації,  $Faq$  – функційна адекватність,  $Fic$  – покриття функційної реалізації,  $Ot$  – час роботи,  $Nic$  – кількість неточних обчислень,  $Ndi$  – кількість елементів даних,  $Ca$  – обчислювальна точність,  $Pc$  – точність,  $Nmot$  – кількість задач,  $Rt$  – час реакції,  $Noe$  – кількість оцінок,  $Tnt$  – час обробки,  $Tskt$  – час задачі,  $Mathr$  – середнє значення пропускнуої здатності,  $Nofl$  – кількість відмов,  $Niore$  – кількість помилок введення-виведення,  $Uwt$  – час очікування користувача під час введення-виведення,  $Nmre$  – кількість помилок пам'яті,  $Ntre$  – кількість помилок передачі даних,  $Tcc$  – потужність передачі даних,  $Iou$  – кількість буферів під час введення-виведення,  $Nolcd$  – безпосередня кількість рядків коду,  $Ioll$  – ліміт завантаження пристроїв введення-виведення,  $Mtu$  – максимум використовуваної пам'яті,  $Mtu$  – максимум використовуваної пере-

дачі даних, *Mote* – середня поява помилки передачі, *Ncu* – кількість одночасних користувачів, *Cbw* – ширина смуги комунікації, *Sdb* – розмір бази даних, *Nott* – кількість посібників, *Niodi* – кількість елементів даних введення-виведення, *Cnd* – повнота описів, *Fua* – зрозумілість функціоналу, *Uaio* – зрозумілість входів та виходів, *Efl* – простота вивчення функціоналу, *Hfq* – частота звертань до довідки, *Eudhs* – ефективність системи допомоги, *Haa* – доступність довідки, *Cudhf* – повнота довідкового фонду, *Ecr* – частота корекції помилок, *Nsf* – кількість екранів або форм, *Nues* – кількість помилок або змін користувача, *Nac* – кількість спроб налаштування, *Nop* – кількість операцій, *Niwccd* – кількість елементів, які можна перевірити на наявність дійсних даних, *Nmi* – кількість виконаних повідомлень, *Nie* – кількість елементів інтерфейсу, *Pha* – фізична доступність, *Neum* – кількість легко зрозумілих повідомлень, *Nurs* – кількість невдало відновлених ситуацій, *Optdo* – час роботи під час спостережень, *Nouheo* – кількість випадків операційних помилок користувача, *Niewusc* – кількість вхідних помилок, які користувач успішно виправляє, *Nacie* – кількість спроб коригування вхідних помилок, *Necwusc* – кількість помилкових умов, які користувач успішно виправляє, *Tnect* – загальна кількість перевірених помилкових умов, *Nfiuet* – кількість функцій, виконаних з толерантністю до помилки користувача, *Tnfrtc* – загальна кількість функцій, що вимагають можливості толерантності, *Tniop* – загальна кількість некоректних шаблонів операцій, *Nige* – кількість графічних елементів інтерфейсу, *Dipu* – ступінь збільшення задоволення користувача, *Disu* – ступінь збільшення задоволеності потреб користувача, *Dea* – ступінь ергономічної привабливості, *Drwmu* – ступінь використання метафор реального світу, *Ewsebuusd* – обсяг, до якого ПЗ можуть користуватись користувачі з обмеженими можливостями, *Ewusd* – ефективність роботи користувачів з обмеженими можливостями, *Efrusd* – свобода від ризику для користувачів з обмеженими можливостями, *Susd* – задоволення потреб користувачів з обмеженими можливостями, *Ppsa* – наявність властивостей, які підтримують доступність, *Noft* – кількість збоїв, *Ps* – розмір продукту, *Ntc* – кількість тестових випадків, *Nrf* – кількість фіксованих відмов, *Nef* – кількість усунутих збоїв, *Fdate* – щільність відмов відносно до тестових випадків, *Frn* – роздільна здатність відмов, *Frl* – усунення збоїв, *Mtbf* – середній час між відмовами, *Tmy* – випробувальний термін, *Elfd* –

орієнтовна щільність прихованих збоїв, *Fdy* – щільність збоїв, *Tidwsis* – загальний час, протягом якого програма перебуває у стані зростання, *Nob* – кількість спостережуваних несправностей, *Tdt* – загальна тривалість простоїв, *Nbd* – кількість несправностей, *Nio* – кількість недозволених операцій, *Ttr* – час ремонту, *Dt* – тривалість простою, *Nrs* – кількість перезапусків, *Nrm* – кількість відновлень, *Ray* – відновлюваність, *Ndfrt* – кількість форматів даних, пов'язаних інструментом, *Ndfbe* – кількість форматів даних для обміну, *Nip* – кількість протоколів інтерфейсу, *Deay* – обмінність даних, *Nidc* – кількість випадків пошкодження даних, *Nat* – кількість типів доступу, *Ncr* – кількість контрольованих вимог, *Aca* – керованість доступу, *Ndiced* – кількість елементів даних, які правильно зашифровані та розшифровані, *Ndibred* – кількість елементів даних, які потребують шифрування та розшифрування, *Nepuds* – кількість подій, оброблених за допомогою цифрового підпису, *Nernrp* – кількість подій, що вимагають властивості невідхилення, *Nasdrsl* – кількість доступів до системи та даних, записаних у системний журнал, *Naa0* – кількість дійсно набутих доступів, *Npam* – кількість наданих методів перевірки автентичності, *Nmm* – кількість внесених змін, *Nv* – кількість змінних, *Nm* – кількість модулів, *Fcy* – функційна спільність, *Nfey* – нефункційна спільність, *Vrn* – міцність варіабельності, *Aay* – застосовність, *Tay* – пристосованість, *Cra* – заміна компонентів, *Ert* – час помилки, *Nirbl* – кількість елементів, необхідних для запису в журналі, *Ndfr* – кількість необхідних діагностичних функцій, *Atc* – можливість аудиту сліду, *Nrv* – кількість переглянутих версій, *Ccca* – можливість управління змінами, *Ntepbm* – кількість проблем протягом певного періоду до модифікації, *Ntspam* – кількість проблем за той же період після модифікації, *Nbfr* – кількість необхідних вбудованих тестових функцій, *Nidos* – кількість тестових залежностей на інших системах, *Ncp* – кількість контрольних точок, *Puf* – портативна дружність користувача, *Nds* – кількість структур даних, *Aads* – адаптивність структур даних, *Hea* – адаптивність апаратного оточення, *Sea* – адаптивність програмного оточення, *Noftwnca* – кількість операційних функцій, завдання яких не були виконані або неадекватні, *Tnfwtdc* – загальна кількість функцій, які були випробувані за різних умов, *Nso* – кількість операцій налаштування, *Nis* – кількість кроків інсталяції, *Eoi* – простота інсталяції, *Net* – кількість сутностей.

**Висновки.** Актуальність проблем підвищення якості ПЗ зумовлює необхідність розроблення моделей оцінювання якості ПЗ, які б враховували вплив та важливість різних факторів під час управління якістю.

Аналіз моделі SQuaRE (ISO 25010) показав, що низка атрибутів якості входять до складу декількох підхарактеристик та характеристик. Якщо такі атрибути відсутні, то одночасне їх використання під час визначення якості ПЗ суттєво вплине на достовірність отриманих оцінок якості ПЗ, тобто кореляція характеристик за атрибутами може погіршувати точність та достовірність визначення якості ПЗ.

Існування кореляції характеристик і підхарактеристик якості за атрибутами збільшує важливість таких атрибутів під час оцінювання якості,

відтак збільшує вагові коефіцієнти атрибутів якості ПЗ.

У статті розроблено компенсаторну модель якості ПЗ на основі моделі SQuaRE, яка забезпечує можливість надання коректних кількісних оцінок якості, враховуючи важливість та взаємовпливи атрибутів якості ПЗ. У разі побудови такої моделі використано раніше отримані вагові коефіцієнти атрибутів. Чисельник вагового коефіцієнта кожного атрибута якості вказує на кількість підхарактеристик якості ПЗ, які залежать від такого атрибута, а знаменник відображає загальну кількість атрибутів якості в моделі SQuaRE (стандарт ISO 25023 [3]). Розроблена компенсаторна модель дає змогу зробити висновок: наявність яких атрибутів є пріоритетною та необхідною для забезпечення відповідного рівня достовірності оцінювання якості ПЗ.

#### Список літератури:

1. Говорущенко Т.О., Малярчук Р.А., Вовк І.В. Аналіз інцидентів, спричинених помилками програмного забезпечення. *Інтелектуальні технології в системному програмуванні* : матеріали Всеукр. наук.-практ. конф. молодих учених та студентів (Хмельницький, 18–19 квітня 2013 р.). Хмельницький, 2013. С. 187–198.
2. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. Introduced 01.03.2011. Geneva (Switzerland), 2011. 34 p. (International standard).
3. ISO 25023:2016. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). Measurement of system and software product quality. Introduced 31.03.2016. Geneva (Switzerland), 2016. 45 p. (International standard).
4. Abran A., Al-Quitash R.E., Desharnais J.-M., Habra N. ISO-based models to measure software product quality. *Software Quality Measurement: Concepts and Approaches*. 2014. Chapter 5. Pp. 61–96.
5. Montagud S., Abrahao S., Insfrán E. A systematic review of quality attributes and measures for software product lines. *Software Quality Journal*. 2012. No. 20 (3–4). Pp. 425–486.
6. A framework for evaluating reusability of core asset in product line engineering / J. Sun Her and others. *Information and Software Technology*. 2007. No. 49. Pp. 740–760.
7. Biscoglio I., Marchetti E. Definition of software quality evaluation and measurement plans: a reported experience inside the audio-visual preservation context. *The 9-th International Joint Conference on Software Technologies: Proceedings* (Vienna, August 29–31, 2014). Vienna (Austria), 2014. Pp. 63–80.
8. Макконнелл С. Совершенный код. Мастер-класс. Москва : Изд. «Русская редакция», 2013. 896 с.
9. Maedche A., Botzenhardt A., Neer L. Software for people: fundamentals, trends and best practices (Management for professionals). Springer-Verlag Berlin Heidelberg, 2012. 293 p.
10. Липаев В.В. Программная инженерия: методологические основы. Москва–Берлин : Директ-Медиа, 2015. 608 с.
11. Глухих М., Ицыксон В.М. Программная инженерия. Обеспечение качества программных средств методами статического анализа : учебное пособие. Санкт-Петербург : Изд. политехн. ун-та, 2011. 150 с.
12. Sugiyanto S., Rochiman S. Integration of DEMATEL and ANP methods for calculate the weight of characteristics software quality based model ISO 9126. *The 5-th International Conference on Information Technology and Electrical Engineering: Proceedings* (Yogyakarta, October 7–8, 2013). Yogyakarta (Indonesia), 2013. Pp. 143–148.
13. Decision making. URL: <http://www.cindy.sifonis.com/Classes/Cog316/cogline/Decision.pdf> (дата звернення: 23.05.2019).
14. Бидихова И.С. Методология и методика социологического анализа конкурентоспособности потребительских товаров. URL: <http://cheloveknauka.com/metodologiya-i-metodika-sotsiologicheskogo-analiza-konkurentosposobnosti-potrebitelskih-tovarov> (дата звернення: 23.05.2019).
15. Novorushchenko T. Method of Evaluating the Weights of Software Quality Measures and Indicators. *Application and Theory of Computer Technology*. 2017. Vol. 2. No. 2. Issue 2. Pp. 16–25.

## КОМПЕНСАТОРНАЯ МОДЕЛЬ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ МОДЕЛИ SQUARE

*В статье разработана компенсаторная модель качества программного обеспечения (ПО) на основе модели SQuaRE, которая обеспечивает возможность предоставления корректных количественных оценок качества, учитывая важность и взаимное влияние атрибутов качества ПО. Разработанная модель позволяет сделать вывод: наличие каких атрибутов является приоритетным и необходимым для обеспечения соответствующего уровня достоверности оценки качества ПО.*

**Ключевые слова:** программное обеспечение (ПО), качество ПО, характеристики качества, атрибуты качества, компенсаторная модель.

## COMPENSATORY MODEL OF SOFTWARE QUALITY BASED ON SQUARE MODEL

*The paper devotes to the compensatory model of software quality based on the SQuaRE model, which provides the possibility of providing accurate quantitative estimates of quality, taking into account the importance and mutual influences of software quality measures. The developed model makes it possible to conclude about the priority and necessity of the presence of certain measures for providing the appropriate level of reliability of the evaluation of software quality.*

**Key words:** software, software quality, software quality characteristics, software quality measures, compensatory model.