

УДК 681.3.06

Г.І. Коваль

ПІДХІД ДО МОДЕЛЮВАННЯ ЯКОСТІ СІМЕЙСТВ ПРОГРАМНИХ СИСТЕМ

Аналізуються проблеми моделювання якості сімейств програмних систем (СПС), визначаються вимоги до моделі якості СПС та пропонується підхід до моделювання, який полягає у комбінуванні різнотипних моделей відповідно до рівнів архітектури СПС (ієрархічних, аналітичних та байєсівських моделей), а також методу аналізу ієрархій для встановлення пріоритетів вимог до характеристик якості і наступного визначення інтегрального показника якості СПС.

Вступ

Однією з новітніх парадигм програмної інженерії є парадигма генерувального програмування, заснована на ідеї інженерії сімейств програмних систем у певній предметній області (ПрО) і генерації окремих програмних систем (ПС) як членів сімейства [1, 2, 3]. Її використання надає організаціям-розробникам низку переваг, а саме: зниження вартості та ризику проєктів окремих ПС, постійне підвищення кваліфікації співробітників, вдосконалення процесу розроблення, стимулювання еволюції СПС і, найголовніше, досягнення встановлених вимог до якості ПС та забезпечення їх конкурентноздатності [4].

Однак, відповідність вимогам до якості кінцевих ПС та інших не функціональних характеристик може бути забезпечена лише в тому разі, якщо забезпечується якість програмних та не програмних артефактів, створених під час розроблення СПС, які накопичуються в репозитарії інтегрованого середовища розроблення (IDE, від Integrated Software Development) СПС як готові ресурси (ГОР) та повторно використовуються в окремих ПС – членах сімейства.

Забезпечення якості ГОР різних категорій, наприклад, окремих програмних компонентів, композицій компонентів, варіантів архітектури, фреймворків, а також сімейства ПС загалом, потребує застосування відповідних категорій моделей якості, які враховують ресурсні обмеження розроблення СПС і експлуатації ПС.

У роботі аналізуються проблеми моделювання якості ГОР СПС окремих кате-

горій та пропонується диференційований підхід до моделювання їх якості.

Проблеми моделювання не функціональних характеристик СПС

Сімейство ПС представлено у репозитарії IDE множиною:

1) програмних ГОР – скриптів, початкового (вихідного) коду, бібліотек, виконуваних (exe-) файлів тощо, якими є:

- окремі програмні компоненти;
- композиції компонентів;

– варіанти архітектури СПС, подані множиною взаємопов'язаних компонентів;

– фреймворки (компонентні каркаси), на яких «розгортається» архітектура у IDE;

- ПС – члени сімейства;

2) не програмних (інформаційних) ГОР – документів, моделей (схем). Це, зокрема, характеристичні моделі ПрО, онтології, функціональні та поведінкові специфікації СПС та ПС, специфікації архітектури СПС та окремих її варіантів.

Характеристики якості ГОР СПС розглядаються в загальному контексті не функціональних характеристик ПС, проєктів та виконуваних процесів, серед яких також присутні ресурсні характеристики – час, трудовитрати, та характеристики програмно-технічних умов (обмежень) експлуатації – комп'ютерного, мережаного обладнання, продуктивності використовуваного програмного забезпечення тощо. Відповідно модель не функціональних вимог до СПС має включати дві складові – модель якості та модель ресурсів СПС.

Модель якості визначає множину характеристик якості та взаємозв'язки між ними і є базисом для специфікації вимог до якості СПС та її оцінювання [5]. Модель ресурсів визначає ресурсні обмеження, як щодо розроблення СПС і ПС, так і щодо роботи встановленої ПС, у межах яких є адекватною модель якості.

На відміну від функціональних вимог до ПС, не функціональні вимоги (НФВ) змоделювати та перевірити набагато складніше. Якщо функціональні вимоги моделюються відомими методами і засобами (FODA, FORM, FeatureRSEB тощо) і реалізуються через програмні ГОР, присутність яких у складі ПС і відповідність вимогам легко перевірити (вони або реалізовані, або ні), – не функціональні вимоги не моделюються через безпосереднє «розкладення» не функціональних характеристик по компонентах. Їх моделювання викликає низку проблем.

По-перше, не функціональні вимоги є суб'єктивними, по різному інтерпретуються й оцінюються учасниками проекту.

По-друге, не функціональні характеристики взаємопов'язані і можуть чинити позитивний або негативний вплив одна на одну, наприклад:

- підвищення рівня зручності використання зумовлює зниження рівня ефективності;
- підвищення рівня ефективності спричиняє зниження рівня готовності і модифікованості;
- підвищення рівня готовності зумовлює зниження рівня безпечності;
- підвищення рівня захищеності спричинює зниження рівня зручності використання й ефективності;
- підвищення рівня будь-якої характеристики якості зумовлює підвищення вартості системи.

По-третє, не функціональні вимоги пов'язані з реалізацією функціональних вимог, перш за все на рівні архітектури, оскільки архітектурні рішення можуть суттєво впливати на досягнення НФВ.

Множина конкретних цільових не функціональних характеристик СПС визначається лише після того, як досить повно визначені функціональні потреби Про

й архітектурні можливості їх реалізації, оскільки тільки тоді НФВ можуть бути декомпозовані за рівнями і окремими компонентами архітектури. Така декомпозиція сприяє оптимізації архітектури з урахуванням загальних функціональних і не функціональних вимог до СПС, а також побудові ПС з найкращою у сенсі якості архітектурою.

Деякі характеристики якості, наприклад, ефективність, захищеність або модифікованість досить суттєво залежать від архітектури СПС і мають «вбудовуватися» у систему на рівні архітектурних рішень. До того, реалізація кожної характеристики якості зазвичай не може бути локалізована у частині архітектури (що прийнятно щодо реалізації окремої функціональної вимоги). Зміна вимог до значення характеристики може спричинити широкомасштабну зміну всієї архітектури (приклад – «вбудовування» коду з захисту інформації у ПС).

Таким чином, варіабельність характеристик якості має інтегруватися у процес систематизованого керування варіабельністю архітектури СПС, включаючи як програмну, так і технічну конфігурацію.

Доцільно виділити три категорії ситуацій, у яких компроміс щодо якості СПС досягається шляхом забезпечення варіабельності окремих характеристик якості [6]:

1) компроміс між окремими характеристиками якості q_1 і q_2 . Замість того, щоб розробляти одну систему з «усередненими» значеннями конфліктуючих характеристик q_1 і q_2 , можна розробити два варіанти її архітектури (в одному перевага надається характеристиці q_1 у збиток характеристиці q_2 , в іншому – навпаки) і надавати право користувачеві СПС комплектувати ПС за власним уподобанням. Прикладом є технічне рішення щодо довжини паролю доступу в систему. Відомо, що, чим довший пароль, тим вище рівень захисту інформації, але нижче ефективність системи, і навпаки. Розробнику ПС на основі СПС треба надати можливість вибрати потрібну комбінацію значень характеристик захищеності й ефективності;

2) компроміс між характеристикою якості q_i і ресурсом проекту r_i . Рівень значення характеристики якості q_i може варі-

юватися відповідно до рівня наявних ресурсів r_i . Приклад – чим вищий рівень захищеності системи потрібний користувачеві, тим дорожче це йому коштуватиме. У СПС треба забезпечувати встановлення (і реалізацію) декількох рівнів кожної характеристики і надавати користувачеві право вибирати найбільш прийнятний;

3) компроміс між характеристикою якості q_i і експлуатаційним ресурсом c_i . Приклад – чим вище роздільна здатність монітора, тим ширша кольорова гама і тим вище значення характеристики зручність використання за відповідним критерієм. У СПС треба реалізовувати механізм масштабування ПС з урахуванням технічних обмежень і надавати користувачеві можливість встановлювати значення характеристики відповідно до класу та призначення ПС (обчислення, 3D-графіка тощо).

Хоча засобом специфікації характеристик якості у СПС є моделі характеристик, вони не передбачають можливості варіювання значень, зокрема, у неперервний спосіб, наприклад, в інтервалі (0, 1). Крім того, не ясно, яким чином змодельовати двосторонні впливи характеристик (наприклад, q_1 впливає на q_2 , а q_2 впливає на q_1).

Одним з виходів є «трансформація» характеристик якості у функціональні характеристики, що, з одного боку, забезпечить можливість пошуку в СПС потрібного варіанта реалізації ПС, а з іншого – спричинить проблему суттєвого переважання характеристичних моделей. Наприклад, в характеристичній діаграмі можуть виникнути такі функції, як «функція захисту з довгим паролем», «функція захисту з коротким паролем» тощо. Інший вихід – замість розроблення стратегій пошуку варіанта архітектури, який відповідатиме встановленому рівню якості певної характеристики, розробити метод прогнозування якості можливих варіантів архітектури (композицій компонентів) у СПС.

Загальні вимоги до якості СПС та моделей якості

Якість СПС асоціюється з множиною характеристик, поданих у табл. 1.

Таблиця 1. Характеристики якості СПС

Назва характеристики	Зміст та підстави для забезпечення
Повторна використовуваність ГОР	Можливість багаторазового використання ГОР через застосування концепції і методів розроблення ПС з повторно використовуваних компонентів (ПВК)
Гнучкість	Забезпечення композиції незалежних компонентів на основі спільності властивостей
Еволюційність	Застосування модельного підходу до розроблення СПС та реалізація механізмів варіабельності СПС
Ефективність	Наявність механізмів генерації ПС як членів СПС
Прозорість	Підтримка метаопису (анотування) ГОР СПС
Концептуальна цілісність	Забезпечення несуперечливості різних категорій ГОР та відсутності категорій ГОР, не пов'язаних з іншими жодними механізмами
Повнота	Достатність (на момент оцінювання) ГОР певної категорії для потреб Про. Приклад – повнота переліку характеристик у моделі характеристик
Коректність	Забезпечення відповідності ГОР певної категорії вимогам до їх подання у репозитарії IDE (дотримання уніфікованості опису ГОР)
Доступність	Наявність в IDE механізмів пошуку ГОР певної категорії та маніпулювання ними. Відсутність ГОР, які технічно не можуть бути використані для побудови ПС
Стандартизованість	Застосування стандартизованих засобів для опису ГОР (рішень OMG, W3C); інтерфейсів ГОР; процесу розроблення СПС і ПС
Продуктивність	Використання готових програмно-технологічних рішень для побудови і ведення ГОР, зокрема готових інструментів

Для того щоб модель якості СПС ефективно використовувалась щодо ГОР

різних категорій, вона має відповідати наступним вимогам [7].

По-перше, вона має бути гнучкою з огляду на контекст її використання, який включає:

- тип організації, яка замовляє ПС і має власне бачення множини характеристик її якості. Підхід до моделювання якості СПС має забезпечувати адаптацію моделі до потреб кожної з організацій–учасників розроблення ПС на основі СПС;

- клас розроблюваної ПС і особливості її експлуатації користувачами різних категорій. Модель має бути придатною для оцінювання якості Веб-застосувань, вбудованого програмного забезпечення тощо, а також враховувати погляди на характеристики якості різних категорій споживачів – менеджерів, персоналу супроводу, користувачів ПС;

- виконуваний процес. Модель має адаптуватися до рівня зрілості організації-розробника ПС, досконалості процесів життєвого циклу (ЖЦ), готовності об'єктів до застосування щодо них програми вимірювання, починаючи на самих ранніх стадіях ЖЦ, а також припускати поповнення даними у ЖЦ ПС та повторне уточнююче оцінювання. Оцінюваними об'єктами є артефакти, розміщені в репозитарії IDE і такі, що стосуються Про, процесу розроблення і ресурсів [3]. Модель має поєднувати кількісні (вимірні) та якісні (експертно оцінені) дані щодо об'єктів, які уточнюються в ході ЖЦ ПС.

По-друге, модель має бути повторно використовуваною, придатною для відображення узагальненого досвіду за множиною проектів (відповідно до парадигми сімейства процесів [3]) і припускати отримання екземпляру моделі для нового проекту, максимально наближеного до вже представлених у сімействі. З одного боку, це підвищить ефективність і зменшить трудовитрати на моделювання якості для нових ПС, а з другого – підвищить точність і ефективність оцінювання якості завдяки накопиченню досвіду оцінювання.

Нарешті, модель якості має бути прозорою щодо складу і змісту характеристика їх взаємозв'язків і забезпечувати їх обґрунтованість та ідентифікацію.

Ієрархічні моделі якості програмних компонентів СПС

Аналіз підходів до моделювання якості СПС показав, що існують труднощі в застосуванні ієрархічних моделей якості, більшість з яких засновується на еталонній моделі характеристик у ISO/IEC 9126 [8]. Вони є статичними за природою і не описують, як робити проєкцію метрик від їх поточних значень на певному етапі проєкту до нових значень на його подальших стадіях, що важливо для визначення ризику проєкту. Крім того, деякі з цільових вимог до якості було б природніше включати у функціональні вимоги до СПС і не вдаватися для їх оцінювання до методів метричного аналізу. Наприклад, характеристика «здатність до взаємодії» може трансформуватися у функціональні вимоги до інтерфейсу ПС, а її досягнення перевіряється тестуванням. Аналогічно можна вчинити і з характеристикою «переносність».

У проблематиці побудови СПС ієрархічні моделі можуть використовуватися з метою специфікації окремих ГОР, що є COTS компонентами (готовими компонентами на ринку програмної продукції), які потребують сертифікації перед включенням у репозитарій IDE СПС.

Постачальники COTS компонентів найбільшу увагу приділяють технічним аспектам їх проєктування і реалізації у гіпотетичних середовищах, тоді як розробники СПС і ПС на їх основі – асемблюванню систем з компонентів та розгортанню у конкретних цільових середовищах. З цієї позиції важливим для них є невизначеність щодо якості, і, перш за все, надійності компонентів через невідомість щодо сценаріїв їх використання (тобто, неможливість визначення їх операційних профілів [9]). Виникає потреба тестування компонентів (вже після придбання) у складі ПС. Проблем додають відсутність доступного початкового коду і детальної документації для супроводу компонентів.

Повторне використання компонентів у цільових середовищах має гарантуватися через чітке специфікування програмних архітектур та сценаріїв, у яких можливе функціонування компонентів, а також обмежень щодо композицій компонентів, у

яких може спостерігатися зниження якості (наприклад, надійності або реактивності) системи.

Іншою проблемою є відсутність супроводжуваності та заміної здатності компонентів, що могло б зменшити залежність від постачальників. На теперішній час для вирішення цієї проблеми можуть складатися договори на гарантійне обслуговування (заміну) компонентів як умова їх сертифікації перед випуском.

Крім того, компоненти мають бути гнучкими, адаптуватися до змін у потребах користувачів, і припускати, у відповідь, розширення специфікацій вихідних вимог до них з попередньою перевіркою непопущності обмежень.

Об'єднуваною властивістю компонентів є їх довірча придатність – властивість підтримувати довіру користувачів до якості компонентів, які використовуються. Ця властивість може забезпечуватися через сертифікацію компонентів стороннім органом сертифікації. Як варіант (для некритичних компонентів) може застосовуватися сертифікація організацій-розробників компонентів. Процес сертифікації компонентів полягає у верифікуванні рівнів забезпечуваних показників якості компонентів та надаванні сертифікату відповідності стандартам, а також адекватності заданій множині встановлених вимог. Процедура сертифікації потребує формування належного середовища тестування компонентів. До того ж необхідні стандартизовані моделі якості компонентів для проведення сертифікації.

Одну з таких моделей запропоновано в [10]. Її структуру показано на рис. 1. Порівняно з еталонною моделлю якості в ISO/IEC 9126 у модель введено додаткові підхарактеристики якості – зовнішні (подані в прямокутниках), оцінювані під час виконання (конфігурованість, масштабованість), і внутрішні (подані в заокруглених прямокутниках), оцінювані під час розроблення (автономність, повторна використовуваність). Крім того, виключено з моделі характеристику «аналізованість» та змінено назву підхарактеристики «здатність до встановлення» на «розгортваність».



Рис. 1. Ієрархічна модель якості COTS

На додаток до характеристик якості запропоновано оцінювати ресурсну характеристику компонентів «ринкова придатність», важливу для сертифікаційного процесу. Вона представлена підхарактеристиками:

- час розроблення – час, необхідний для розроблення компонента;
- вартість – ціна компонента;
- час до випуску – час, необхідний для випуску компонента на ринок;
- ринковий обсяг – цільові обсяги випуску компонента на ринок;
- попит – наскільки компонент користується попитом;
- ліцензованість – наявність ліцензій компонента.

Хоча ці характеристики не мають безпосереднього відношення до якості, вони є важливими для формування думки споживачів (розробників СПС) щодо довірчої придатності компонентів. Так, наприклад, дані про час, необхідний для доведення компонента до випуску, може свідчити про його завершеність.

Поряд з представленою є й інші комбіновані моделі, які включають характеристики не лише якості, але й ресурсів (наприклад, модель Т. Джілба [11], моделі COQUALMO [12], SQUID [13] тощо).

Вони розширюють можливості ієрархічних моделей і можуть адаптуватися до потреб окремої організації або класу ПС. Ці моделі враховують більшість чинників, що впливають на якість ПС [8], допомагаючи уникнути конфліктів між різними вимогами до якості, дають можливість указувати множину вимірів для функціонально розрізнених компонентів ПС (обчислювальних, інтерфейсних, інформаційних (баз даних)) і встановлювати різні правила вимірювань.

Особливості моделювання архітектури в контексті якості СПС

Архітектура СПС визначає концепти, структуру та внутрішню будову СПС, необхідні для досягнення варіабельності характеристик ПрО у варіантах ПС (членах ПС) із забезпеченням максимального спільного використання частин (композицій компонентів) реалізації. Мета моделювання архітектури – передбачення якості ПС до її побудови, виходячи із знань, який вплив матимуть архітектурні рішення на показники якості. Інша мета – з'ясування думок щодо адекватності архітектури вимогам до СПС і визначення її слабких сторін. Існують два способи відображення якості в архітектурі:

- використання архітектурних стилів та шаблонів для «вбудовування» вимог до якості у ПС;

- використання профілів якості (наприклад, розширень профілів UML), які доповнюють архітектурну модель певними аспектами якості та є засобом зіставлення вимог до якості й архітектурних рішень з метою оцінювання архітектури.

Ці способи подання якості застосовуються у сучасних методологіях проектування та оцінювання архітектури (HoPLAA [14], QADA [15] тощо).

Однак, для забезпечення повної відповідності вимогам до моделей не функціональних характеристик СПС у моделювання якості треба залучати методи інтелектуального аналізу даних, зокрема, засновані на байєсівському підході [16]. В роботі [17] обґрунтовано доцільність його застосування для моделювання умовних зв'язків чинників якості та їх впливу на

цільову характеристику якості ПС. Практичної значущості цей підхід набув лише з появою байєсівських мереж (Bayesian Network), на базі яких будуються графічні байєсівські моделі (БМ) [18]. Основна перевага БМ – підтримка прогнозування заданого показника якості ПС і діагностики найбільш вірогідних причин (джерел) його не досягнення. До того ж, апарат байєсівських мереж дозволяє будувати достатньо складні моделі (сотні вершин), які не можуть бути реалізовані іншими методами.

Використання БМ у проблематиці розроблення СПС дозволяє враховувати різні аспекти і джерела варіабельності – відмінні вимоги замовників щодо функціональних та не функціональних характеристик множини ПС, створюваних на основі СПС, різновиди середовищ експлуатації цих ПС, особливі потреби щодо їх супроводження, еволюції тощо.

БМ репрезентує знання експертів у ПрО і досвід виконання подібних проектів. Вона стає невід'ємним ГОР у IDE, який повторно використовується в ЖЦ СПС для якісного аналізу впливу конфігурацій варіантів рішень на показники якості.

Приклад використання БМ для аналізу впливу варіантів архітектури на її якість показано на рис. 2. Це фрагмент байєсівської мережі SAABNet (Software Architecture Assessment Belief Network) для оцінювання архітектури [19].

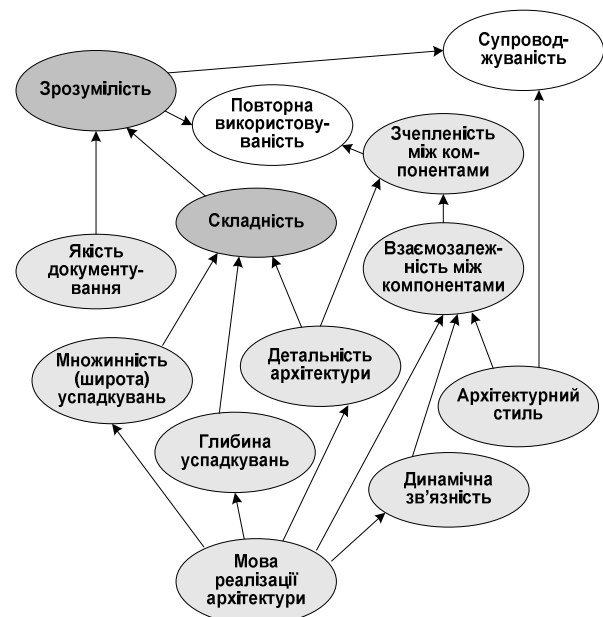


Рис. 2. Модель оцінювання архітектури

Вершини БМ представляють:

- атрибути архітектури на різних рівнях подання (об'єктно-орієнтована мова програмування архітектури, архітектурний стиль тощо). Ці атрибути подано на рис. 2 вершинами БМ світло-сірого кольору;

- підхарактеристики якості та відповідні їм метрики (складність, зрозумілість тощо). Вони подані на рис. 2 вершинами БМ темно-сірого кольору;

- характеристики якості, заявлені у вимогах до ПС, наприклад, повторна використовуваність, супроводжуваність (вершини білого кольору).

Дуги БМ встановлюють умовні залежності між сутностями у вершинах.

Іншим відомим підходом до моделювання якості ПС, який використовує байєсівські мережі і задовольняє встановленим вимогам до моделей якості СПС, є підхід Prometheus [7]. Це ціле-орієнтований підхід до вимірювання та оцінювання якості СПС, який інтегрує кількісні та якісні методи моделювання. Процес моделювання включає три фази:

- фаза специфікації вимог до якості, під час якої будується модель якості. Послідовність кроків щодо специфікації моделі якості показано на рис. 3;

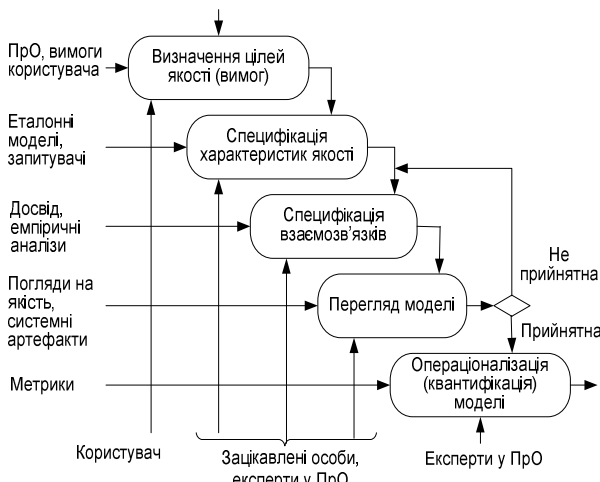


Рис. 3. Кроки специфікації моделі якості

- фаза застосування, в ході якої виконується оцінювання ступеня досягнення вимог з використанням моделі якості;

- фаза накопичення досвіду, під час якої інформація щодо застосування моделі збирається і аналізується з метою покращення моделі та її повторного використання в інших проектах ПС.

Вимоги до якості визначаються із застосуванням методології вимірювання «Ціль-Питання-Метрика» (GQM, від Goal-Question-Measure) [8]. Відповідно до методології вони конкретизуються до переліку вимірюваних характеристик і підхарактеристик оцінюваних програмних об'єктів сімейства. Потім встановлюються взаємозв'язки між характеристиками (підхарактеристиками) і вимірами. Розрізняються два типи зв'язків – декомпозиція характеристик (згори – вниз) і вплив рівня (значення) однієї характеристики на значення іншої. Метрики можуть мати не кількісне, а якісне (експертне) подання (з використанням, наприклад, порядкової шкали).

Байєсівські мережі використовуються в Prometheus для квантифікації зв'язків (впливів), а також комбінування кількісних і якісних даних. Модель якості будується через опитування експертів у ПрО, які визначають цілі та характеристики якості.

Недоліки названого підходу стосуються специфікації характеристик якості. По-перше, Prometheus не використовує ISO/IEC 9126 як орієнтир для побудови моделі якості, а по-друге, не надає формалізованої структури специфікації вимог до якості.

Схема моделювання якості СПС

Найбільш прийнятним підходом до моделювання якості СПС є підхід, який розвиває підхід Prometheus і полягає у комбінуванні різноманітних моделей відповідно до рівнів архітектури, наприклад:

- ієрархічних моделей на рівні окремих компонентів;

- аналітичних моделей залежностей для композицій компонентів;

- байєсівських моделей щодо частин архітектури, для яких не можуть бути встановлені аналітичні залежності та на момент оцінювання відсутні значення вимірів якості у репозитарії IDE;

- методу аналізу ієрархій (MAI) для збалансування вимог до характеристик якості й/або визначення інтегрального показника якості СПС.

Схему моделювання якості за запропонованим підходом показано на рис. 4.

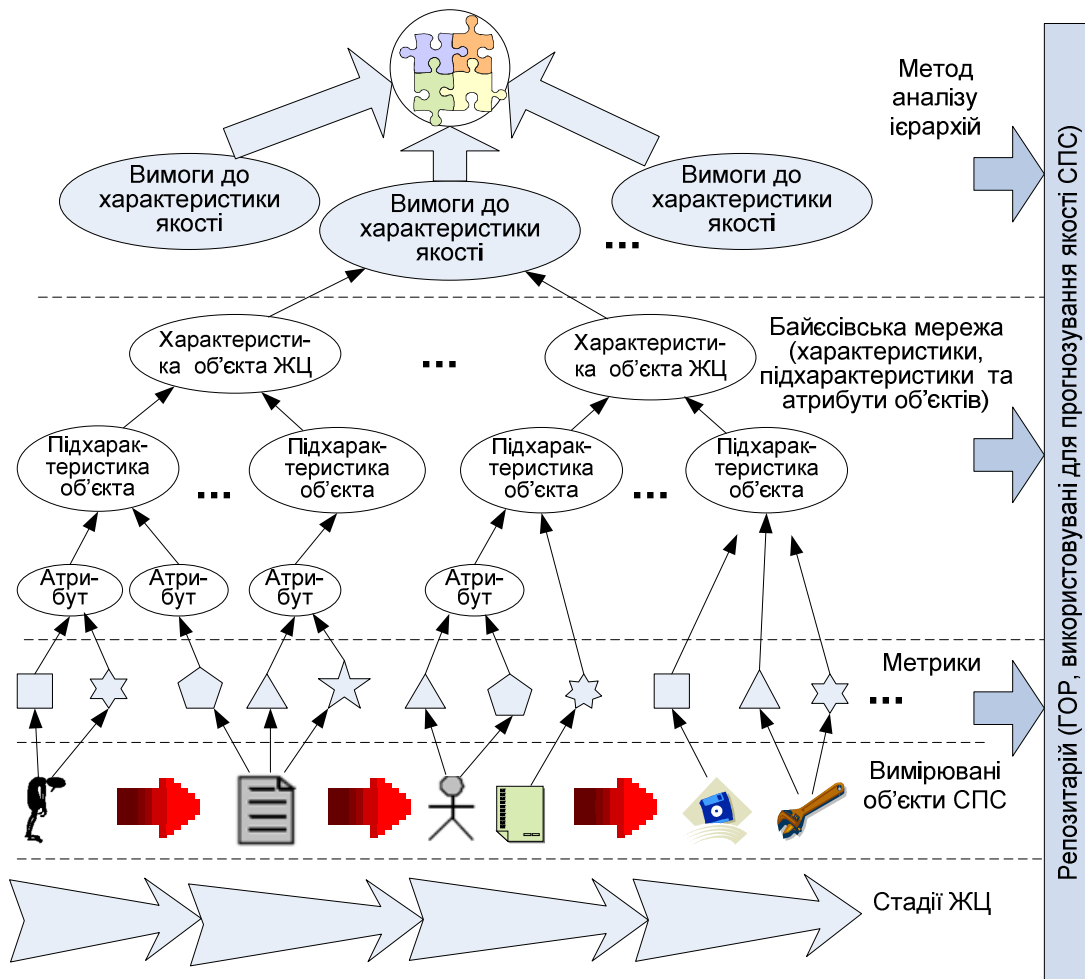


Рис. 4. Схема моделювання якості СПС

Для кожної стадії ЖЦ СПС визначаються оцінювані об'єкти (щодо Про, виконуваних процесів і використовуваних ресурсів, включаючи людські) та їх атрибути, встановлюються процедури їх вимірювання, вибираються метрики оцінювання та будується байєсівська мережа.

Залежно від оцінюваної характеристики якості, метрики можуть бути простими, рекомендованими ISO/IEC 9126, або складними аналітичними моделями (наприклад, для оцінювання надійності компонентів [18]).

Специфікація вимог до якості ГОР виконується за шаблоном (табл. 2).

Для формалізації процесу встановлення пріоритетів вимог до якості, розв'язання конфліктів з урахуванням поглядів на якість всіх учасників проекту СПС, а також для кількісного порівняння різних варіантів архітектурних рішень з позицій якості, використовується МАІ.

Цей метод вже використовувався в проблематиці інженерії якості для встановлення пріоритетів компонентів функціональної архітектури одиначної ПС і наступного розподілу за ними цільових вимог до якості ПС та оцінювання внеску компонентів у досягнення якості за встановленою характеристикою [20].

Запропонований підхід забезпечує можливості:

- виконання оцінювання, починаючи з ранніх стадій ЖЦ;
- ефективне навчання впродовж розроблення нових ПС на основі СПС;
- уточнення моделі якості за послідовними проектами ПС;
- інтеграцію кількісних (заснованих на вимірюванні) і якісних методів;
- комбінування різних контекстів індивідуальних поглядів на якість СПС (розробників, користувачів) і оцінюваних об'єктів (процесів, продуктів, ресурсів).

Таблиця 2. Шаблон специфікації вимог

Атрибут специфікації	Зміст атрибута специфікації
Назва характеристики	Стисле визначення або посилання на стандарт, в якому подано визначення характеристики
Опис вимоги	Обґрунтування потреб щодо забезпечення вимоги
Контекст (фокус)	Об'єкт, якого стосується вимога (ГОР, ПС, процес)
Джерело	Джерело інформації щодо вимоги (документи, замовники, розробники тощо)
Зв'язки	Зв'язок з характеристиками якості в еталонній моделі ISO/IEC 9126
Категорія	Приналежність вимоги до рівнів якості (внутрішня, зовнішня, експлуатаційна [8])
Пріоритет	Значущість вимоги для утримувачів СПС (максимальна, середня, низька)
Обов'язковість	Обов'язкове / необов'язкове виконання вимоги
Вплив	Дії у процесах ЖЦ, інші категорії вимог, які зазнають впливу даної вимоги
Чутливість	Вплив з боку інших характеристик (позитивний / негативний)

Висновки

Якість ПС, розроблюваних за парадигмою генерувального програмування, обумовлюється якістю окремих ГОР, використаних на різних стадіях побудови ПС на основі СПС, зокрема, програмних ГОР та їх архітектурних сполучень.

Для моделювання якості програмних ГОР можуть використовуватися відомі ієрархічні моделі, але вони не застосовні для моделювання якості СПС.

Моделювання якості СПС стикається з низкою проблем, які стосуються, насамперед, його архітектури, яка має бути прийнятною для будь-якої ПС сімейства та є критичною для досягнення таких характеристик якості як ефективність, безпечність, супроводжуваність.

З іншого боку, зміни вимог до якості можуть зумовити зміну архітектури.

Модель якості, яка б враховувала означені проблеми, має бути гнучкою, прозорою та повторно використовуваною для всіх ПС сімейства. Встановленим вимогам відповідають байєсівські графічні моделі, які інтегрують кількісні та якісні оцінки щодо вимірюваних об'єктів СПС. Запропонований підхід до моделювання якості СПС полягає у комбінуванні різнотипних моделей відповідно до рівнів архітектури СПС (ієрархічних, аналітичних та байєсівських моделей), а також методу аналізу ієрархій для встановлення пріоритетів характеристик якості СПС і, за необхідності, визначення інтегрального показника якості.

1. Лаврищева К.М. Генерувальне програмування програмних систем і їх сімейств // Проблеми програмування. – 2009. – № 1. – С. 3 – 16.
2. Лаврищева К.М., Коваль Г.І., Коротун Т.М. Підходи інженерії якості сімейств програмних систем // Проблеми програмування. – 2008. – № 2-3. – С. 219 – 228.
3. Лаврищева К.М., Коваль Г.І., Слабоспицька О.О., Колесник А.Л. Особливості процесів керування при створенні сімейств програмних систем // Проблеми програмування. – 2009. – № 3. – С. 40 – 49.
4. Framework for Software Product Line Practice, version 5 – <http://www.sei.cmu.edu/productlines/index.html>
5. ISO/IEC 9126-1:2001 Software Engineering – product quality. Part 1 Quality model.
6. Myllarniemi V., Mannisto T, Raatikainen M. Quality Attribute Variability within a Software Product Family Architecture – <http://www.soberit.hut.fi/vmyllarn/publications/Myllarniemi06bQualityAttributeVariability.pdf>
7. Trendowicz A., Punter T. Quality Modeling for Software Product Lines – http://www-ctp.di.fct.unl.pt/QUASAR/QAOOSE2003/papers/QAOOSE_2003_TrendowiczPunter_Quality_Modeling_for_Software_Product_Lines.pdf
8. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Суслов // 2-е издание. – Киев.: Академперіодика, 2007. – 672 с.
9. Коваль Г.И., Мороз Г.Б., Коротун Т.М. Концепция профилей в инженерии надежности программных систем // Математичні машини і системи. – 2004. – № 1. – С. 166 – 184.
10. Alvaro A., Almeida E.S., Meira S.R. Quality Attributes for a Component Quality Model – <http://research.microsoft.com/~cszypers/events/WCOP2005/09%20-%20Alvaro.pdf>
11. Gilb T. Principles of Software Engineering Management, Addison-Wesley. – 1988. – 442 p.

12. *Madachy R., Boehm B.* Assessing Quality Processes with ODC COQUALMO // Lecture Notes in Computer Science. – 2008. – Vol. 5. – P.198 – 209.
13. *Kitchenham B., Linkman S., Paquini A.* The SQUID approach to defining a quality model // Software Quality Journal. – 1997. – Vol. 6. – P. 211 – 233.
14. *Olumofin F.G., Misisic V.B.* Extending the ATAM Architecture Evaluation to Product Line Architectures. – <http://www.cs.umanitoba.ca/~vmisisic/pubs/tr0502.pdf>
15. *Mainilassi M., Niemela E., Dobrica L.* Quality-driven architecture design and quality analysis method. – <http://www.vtt.fi/inf/pdf/publications/2002/P456.pdf>
16. *Моррис У.* Наука об управленні. Байєсовський підход. – М.: Мир, 1971. – 304 с.
17. *Коваль Г.І.* Байєсівські мережі як засіб оцінювання та прогнозування якості програмного забезпечення // Проблеми програмування. – 2005. – № 2. – С. 15 – 23.
18. *Коваль Г.І., Коротун Т.М., Мороз Г.Б.* Байєсівські мережі: застосування для керування програмними проектами // Вісник МНТУ. – 2008. – № 2. – С. 125 – 132.
19. *van Gurp J.* Variability in Software Systems. The Key to Software Reuse – <http://www.jillesvangurp.com/static/Lic/licentiatethesis.pdf>
20. *Коваль Г.І., Мороз Г.Б.* Моделювання вимог до якості програмних систем оброблення даних // Проблеми програмування (Спец. вип. конф. УкрПРОГ-2006). – 2006. – № 2-3. – С. 237 – 244.

Про автора:

Коваль Галина Іванівна,
кандидат фізико-математичних наук,
старший науковий співробітник відділу.

Місце роботи автора:

Інститут програмних систем
НАН України
03187, Київ-187,
Проспект Академіка Глушкова, 40.
Тел.: (044) 526 4579.
e-mail: gagaips@ukr.net

Отримано 17.07.2009