

УДК 681.3

П.П. Ігнатенко, В.М. Бистров

ТЕХНОЛОГІЧНІ АСПЕКТИ СТВОРЕННЯ ЖИТТЄЗДАТНИХ ПРОГРАМНИХ СИСТЕМ В УМОВАХ ГЕНЕРУЮЧОГО ПРОГРАМУВАННЯ

Розглядається подальший розвиток підходу авторів до створення програмних систем з використанням моделі життєздатності. Розвиток здійснено в напрямку технологізації їх підходу в умовах генеруючого програмування

Вступ

У колі наукових дисциплін і теорій нині активно розвивається теорія життєздатності складних систем (ТЖС) [1] – живих систем, суспільства та людини, а також складних гібридних систем, засновниками якої для біологічних систем є Р. Міллер [2], а для організацій С. Бір [3]. У свою чергу вони базувалися в своїх дослідженнях на роботах кібернетика У.Р. Ешбі [4].

У теорії життєздатності складних систем одним із базових її понять є поняття моделі життєздатності таких систем, яка має:

- включати в число об'єктів моделювання не лише саму систему, а й актуальне для неї оточуюче середовище;
- адекватно відображати з єдиних позицій сімейство систем;
- мати способи опису складних ієрархічних систем;
- враховувати наявність обмежень на архітектуру та поведінку системи;
- враховувати зміни архітектури та поведінки системи, а також зміни середовища їх існування;
- адекватно враховувати багатокритеріальність оптимізації архітектури, функцій і поведінки систем.

ТЖС є теоретичним підґрунтям виконання проекту щодо життєздатності складних програмних систем (ПС). Зазначимо при цьому деякі доведені загальні твердження із ТЖС, особливо важливі для нашого дослідження щодо життєздатності складних ПС:

- розвинуті життєздатні системи є гармонізованими внутрішньо та тим самим відносно стабільними;

- система тим більш життєздатна й стійка, чим менший її кордон із зовнішнім середовищем;

- система може підвищити життєздатність за рахунок переструктурування й агрегування елементів та кооперування їх у підсистеми без виходу за рамки своєї характерної структури чи організації;

- ефект мінімізації небезпечного кордону може бути досягнуто шляхом агрегування даної системи з іншими системами із актуального середовища.

Всі вищезазначені положення із ТЖС у повній мірі належать предмету нашого дослідження – розробка теоретичних та методологічних основ життєздатності програмних систем.

Уперше поняття життєздатності для систем організаційного типу було запропоновано С. Біром, який при цьому опирався на закон необхідної різноманітності У. Р. Ешбі. Ним було запропоновано формалізовану модель – Модель Життєздатних Систем (МЖС) для організацій.

При цьому життєздатна система (структурно – у вигляді структури МЖС) має складатися із життєздатних підсистем і самостійно здійснювати деякі специфічні функції.

С. Бір за допомогою МЖС вирішував задачі підвищення якості управління організацією як умови її життєздатності та ефективного функціонування. Запропонована С. Біром декомпозиція спрощує систему прийняття рішень на кожному

рівні ієрархії (зменшує невизначеність і знімає обмеження), а рознесення діяльності за видом і розвитком забезпечує «нормування» за кінцевим результатом – виживанню системи в цілому.

На відміну від життєздатності складних систем взагалі та систем організаційного управління, зокрема, питання життєздатності ПС у спеціальній літературі до недавнього часу були мало досліджуваними. Вони почали активно досліджуватися після виходу роботи С. Херрінга [5] у 2002 році, в якій він, опираючись на розробки С. Біра, сформулював інтелектуальну парадигму управління для ПС, слідуючи якій вони могли розвиватися узгоджено із складним навколишнім середовищем (коєволюціонувати). Основним теоретичним результатом роботи С. Херінга є обґрунтоване твердження, що *поняття життєздатності визначає головну властивість наступного покоління ПС.*

Авторами раніше розглянуто проблему життєздатності ПС у рамках запропонованого ними підходу до створення життєздатних ПС з використанням моделі життєздатності для об'єктно-орієнтованого та компонентного програмування [6–12], які, в основному, орієнтовані на створення одиночних систем.

Одним із ключових принципів програмної інженерії є її розвиток у напрямку індустріалізації створення ПС [13]. Генеруюче програмування – це саме той напрямок розвитку програмної інженерії, який направлений на індустріалізацію створення ПС та потребує свого підходу до вирішення проблеми забезпечення життєздатності членів сімейства ПС. У роботі представлено такий підхід та описано технологічні аспекти створення життєздатних членів сімейства ПС в умовах генеруючого програмування.

Перші результати в цьому напрямку були опубліковані в попередній роботі авторів [14], присвяченій особливостям забезпечення життєздатності ПС в умовах генеруючого програмування. Дана робота містить розвиток цього дослідження в напрямку технологічних аспектів створення життєздатних ПС в цих умовах.

Результати роботи отримано в рамках відомчої теми Інституту програмних систем НАН України "Розробка теоретичних основ генеруючого програмування, прикладних та інструментальних засобів його підтримки". Науковий керівник теми – доктор фізико-математичних наук, професор К.М. Лавріщева.

1. Забезпечення життєздатності ПС в умовах генеруючого програмування

В основі нашого підходу до забезпечення життєздатності ПС для різних методологій їх створення лежить застосування моделі життєздатності. Особливостями її використання в умовах генеруючого програмування і визначаються технологічні аспекти створення життєздатних ПС сімейства.

Відповідно до застосування парадигми генеруючого програмування та концепції забезпечення життєздатності ПС для інженерії предметної області розглядається така життєздатна ПС. По-перше, пристосована до внесення змін (здатна змінюватись) за участю розробника системи відповідно до зміни зовнішнього середовища (предметної області та вимог користувача) у визначених межах. По-друге, володіє засобами пристосування до змін на основі певної інфраструктури повторного використання, засобів генеруючого програмування та засобів підтримки життєздатності ПС у рамках процесу еволюційної розробки. По-третє, розроблюється та супроводжується за наявності обмеженості ресурсів (зокрема, часових, фінансових, робочих) відповідно до певних економічних моделей.

Проблему забезпечення життєздатності ПС сімейства на основі означеного підходу ми пропонуємо розглядати, базуючись на наступних ключових принципах:

розробка на основі інваріантів – зміни в ПС та її еволюційні перетворення здійснюються при наявності та збереженні певних артефактів (на різних рівнях абстракції), які визначають такі характеристики життєздатності як ідентичність (identity) та автономність (autonomy) системи, її адаптованість (adaptable-ability),

стійкість (stability) при внесенні змін та їх простеження (traceability), оптимальність структури та оптимальність процесу еволюційної розробки системи; інваріанти визначають організацію і структуру життєздатної ПС та направляють процес розробки для досягнення системою якості життєздатності (viability quality) [15]; характеристики життєздатності залучаються до обраної моделі якості інженерії ПС та узгоджуються із нею відповідно до певних критеріїв життєздатності;

“інтелектуалізація” технології еволюційної розробки та зворотний зв'язок – зовнішнє середовище, середовище розробки, інфраструктура підтримки життєздатності, програмна система (in focus) взаємодіють із використанням механізмів зворотного зв'язку на основі моделі прийняття рішень і повторно використовуюваного інтелектуального ресурсу в області архітектури та інженерії ПС; ПС проявляє активність, впливаючи на зовнішнє середовище;

оцінювання показників життєздатності (якісних показників ПС та її економічних показників) – якісні показники життєздатності визначають якість сімейства ПС у рамках домену та інженерні характеристики якості ПС згідно стандарту ISO/IEC 9126-1; економічні показники життєздатності визначаються у відповідності з «принципом повернення інвестицій» (return of investment, ROI) та прогнозованістю необхідного і достатнього часу існування системи;

розробка та використання моделі життєздатності – модель життєздатності забезпечує на мета-рівні мультипарадигмений підхід до створення життєздатних ПС на основі визначення і використання інваріантів домену та застосувань.

Модель життєздатності включає наступні складові.

Еталонну модель архітектури життєздатних ПС, яка забезпечує *якість життєздатності* ПС на основі *визначення та підтримки* архітектурних інваріантів домену та застосувань.

Засоби формування архітектури ПС, яка базується на еталонній моделі, та

засоби моделювання ПС сімейства. Пропонується використовувати уніфіковану мову моделювання UML та засоби розширення UML для підтримки еталонної моделі.

Засоби підтримки еволюції на основі моделі простежування змін, узагальнених сценаріїв та моделі прийняття рішень.

Методологію запровадження моделі життєздатності для певної предметної області на основі побудови відповідної еталонної моделі домену (domain reference model).

Еталонна модель визначає базові концепції архітектури життєздатних ПС та встановлює зв'язок між ними. Ключовим поняттям еталонної моделі є *архітектурний інваріант*. Архітектурні інваріанти відповідають інваріантам моделей життєздатних систем або інваріантам предметної області, які визначаються на основі *критеріїв життєздатності*. При утворенні архітектури ПС вони об'єднуються відповідно до певного архітектурного стилю, який забезпечує досягнення ПС якості життєздатності. Наприклад, архітектурні інваріанти, які відповідають інваріантам моделі життєздатних систем Біра, можуть бути об'єднані на основі архітектурного стилю *Парадигма Керування* (Control Paradigm) [5], а архітектурні інваріанти, які відповідають інваріантам LST Дж. Міллера, можуть бути об'єднані на основі архітектурного стилю *Модель-Представлення-Оброблювач* (Model-View-Controller, MVC). Наприклад, архітектурні інваріанти предметної області у вигляді архітектурних аспектів можуть бути об'єднані в рамках еталонної архітектури сімейства ПС за умовами досягнення оптимальних значень показників якості ПС. Архітектурні інваріанти підтримуються механізмами, які забезпечують пристосування ПС до змін у зовнішньому середовищі (адаптивність – при функціонуванні та здатність до адаптації при модифікаціях) у рамках певних еволюційних перетворень, її стійкий розвиток (стабільність) та простежуваність змін у системі протягом життєвого циклу (трасованість), а також можливість забезпечення якості життєздатності (qua-

lity of viability) на основі певної моделі якості системи.

Проведені дослідження показали, що при реалізації означеного підходу критичним фактором щодо забезпечення життєздатності ПС є встановлення зв'язку між артефактами ПС сімейства та характеристиками предметної області. Такий зв'язок досягається на основі застосування шаблонів проектування в рамках Domain-driven approach. Зокрема, це Product-line Architecture Design Approach та Pattern-driven Architecture Design Approach, які видобувають абстракції архітектурного проектування із моделі предметної області (domain model). Найбільш придатним для підтримки архітектури лінійки продуктів (product line architecture, PLA) з точки зору концепції забезпечення життєздатності є Domain-Specific Software Architecture (DSSA) approach [16].

Отже, важливим технологічним аспектом створення життєздатних ПС є створення її архітектури – еталонної архітектури домену, архітектури сімейства ПС та Каркасу життєздатної ПС.

Також, як показали наші попередні дослідження, базований на шаблонах та керований шаблонами підхід до створення ПС на основі моделі життєздатності вимагає наявності моделі якості, яка містить характеристики життєздатності та інженерні характеристики якості.

Важливими технологічними аспектами створення життєздатних ПС є вибір критеріїв життєздатності, визначення характеристик життєздатності та створення метрик оцінювання характеристик життєздатності ПС за її моделлю.

Крім того, ще одним важливим технологічним аспектом є створення моделей прийняття рішень щодо вибору варіанта життєздатної ПС, який найбільше задовольняє вимоги користувача.

Враховуючи напрацювання С. Біра, Ч. Херрінга та авторів статті, необхідним технологічним аспектом є узгодження моделей системи, що забезпечують її життєздатність та створення інтегрованої моделі життєздатності доменів та застосувань.

2. Створення інтегрованої моделі життєздатності членів сімейства ПС

В аспекті мультипарадигмної концепції до створення ПС проблему забезпечення їх життєздатності пропонуємо розглядати як проблему розробки інтегрованої моделі забезпечення життєздатності доменів та застосувань. Така інтегрована модель вміщує характеристики життєздатності відповідно до узагальненої моделі якості, правила об'єднання зразків та компонентів реалізації відповідно до моделі конфігурування, параметри мінливості відповідно до моделі мінливості а також критерії життєздатності відповідно до моделі життєздатності.

Інтегрована модель забезпечення життєздатності доменів та застосувань враховує наступні аспекти створення ПС:

кібернетичний аспект у тій його частині, який стосується концепцій зворотного зв'язку та необхідності застосування інформаційної моделі системи для забезпечення керування системою та її еволюції. Цей аспект може бути реалізовано, зокрема, на основі моделі життєздатних систем С. Біра, або моделі, яка відповідає теорії живучих систем Дж. Міллера. В рамках інженерії ПС такому підходу відповідає, зокрема, розробка ПС на основі архітектурного стилю Модель-Представлення-Оброблювач (Model-View-Controller, MVC);

аспекти інженерії ПС у тій їх частині, які розглядають архітектуру ПС як визначальний артефакт щодо забезпечення необхідної якості ПС (architecture-centric approach), визначають архітектурні інваріанти ПС щодо її організації та структури, підтримують керовану моделями розробку з шаблонами, підтримують розробку оптимальної структури системи та оптимальність процесу еволюційної розробки системи на основі моделі якості та шаблонів проектування. До таких підходів належить, зокрема, орієнтований на шаблони аналіз та проектування (Pattern Oriented Analysis and Design, POAD), керована моделями розробка (Model Driven Engineering, MDE) з шаблонами та

ресурсами, що повторно використовуються тощо;

аспекти інженерії Про для сімейства ПС у тій їх частині, яка стосується визначення еталонної архітектури сімейства ПС, інваріантів предметної області та оптимальної конфігурації компонентів ПС. До таких підходів, зокрема,

належить породжуюча розробка програмного забезпечення (Generative Software Development, GSD), предметно-орієнтоване моделювання (Domain-Specific Modeling, DSM) тощо. Схематично формування інтегрованої моделі життєздатності членів сімейства ПС представлено на рис. 1.

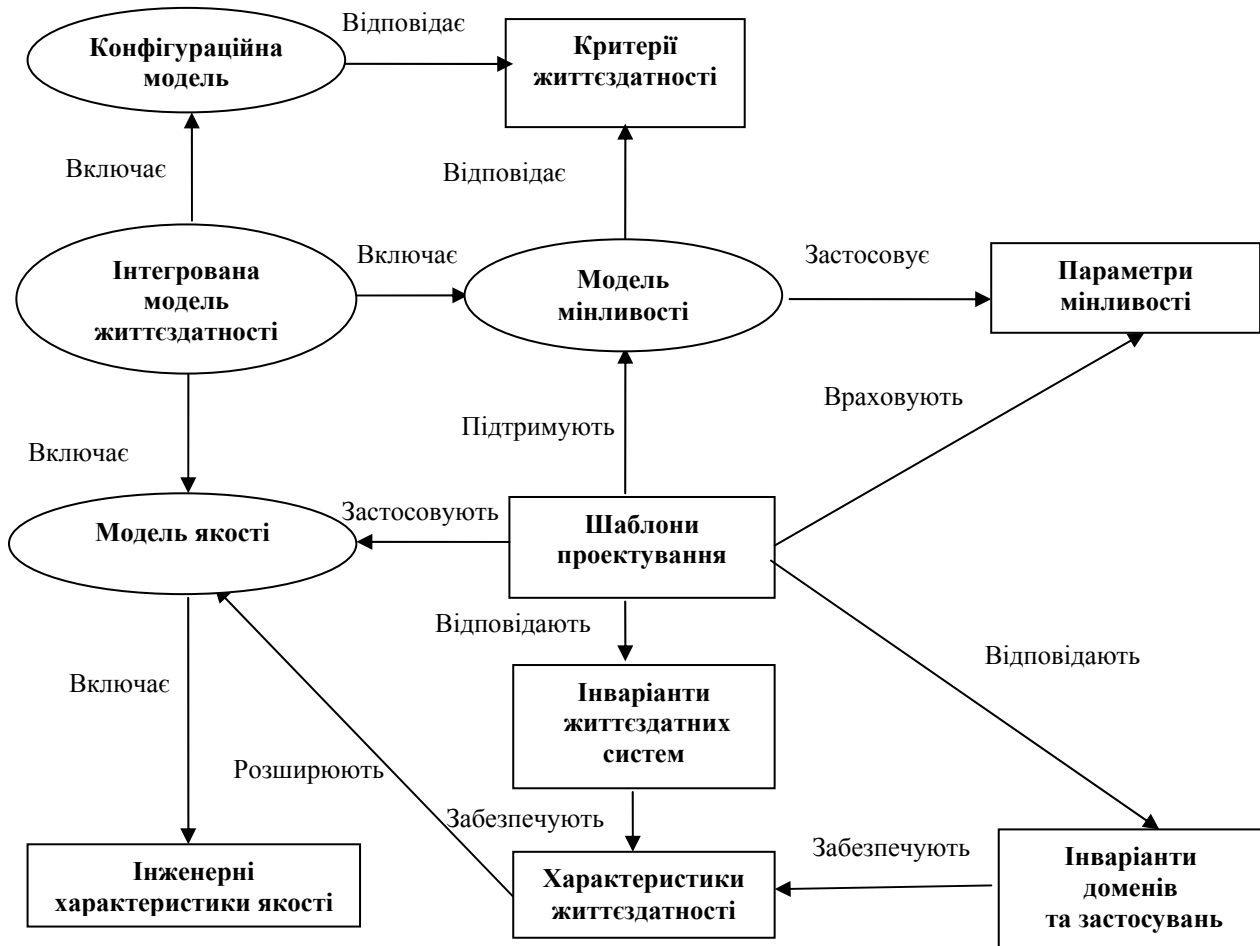


Рис. 1. Інтегрована модель життєздатності членів сімейства ПС

3. Характеристики життєздатності членів сімейства ПС та їх оцінювання

Як уже зазначено, важливим технологічним аспектом моделювання життєздатних ПС є оцінювання їх показників життєздатності за моделю.

Показники життєздатності, які задіяні в прийнятті проектних рішень щодо ПС розділяються на два типи показників – показники якісного та економічного характеру.

Показники якісного характеру. За класифікацією стандарту ISO/IEC 9126-1 до характеристик життєздатності ПС можна віднести такі характеристики якості ПС як *супроводжуваність (maintainability)* – властивості ПС, що обумовлюють можливість її ефективною модифікації, включаючи коригування, удосконалення або адаптацію ПС до зміни середовища, вимог та функціональних специфікацій, а також *переносимість (portability)* – властивості ПС, що обумовлюють її здатність бути перенесеною з одного середовища до іншого. Наведених

характеристик якості недостатньо для оцінювання життєздатності членів сімейства ПС. Тим більше вони не дають змоги оцінювати життєздатність довільних складних систем.

Для вирішення проблеми нами запропоновано три важливі характеристики довільних систем, пов'язаних із їх життєздатністю. Вони також актуальні для інженерії предметної області. Ці характеристики можуть розглядатися, як доповнення до відомої моделі якості ПС. Характеристики життєздатності «стабільність», «модифікуємість», «оптимізуємість», що пропонуються до розгляду, збагачують модель якості ПС, їх значення визначається та діє у рамках розглядуваної предметної області (домену).

Характеристика «стабільність». «Стабільність» – це властивість життєздатної системи, що визначає ступінь її адаптивності при функціонуванні.

Показник «стабільність» тісно пов'язаний з такими математичними поняттями теорії систем [17], як: зворотній зв'язок, координуючий сигнал, кординуємість, сумісність, узгодженість. Характеристика «стабільність» визначається із сукупності показників, що визначають стан координованості системи в теорії систем шляхом їх інтеграції. У програмній інженерії показник «стабільність» визначається із сукупності показників, що визначають стан адаптивності ПС, шляхом їх інтеграції [18].

Характеристика модифікуємість». «Модифікуємість» – це властивість життєздатної системи, що визначається наявністю інфраструктури щодо організації її модифікації в середовищі функціонування.

Характеристика «модифікуємість» тісно пов'язана з такими математичними поняттями теорії систем, як: сумісність, узгодженість, декомпозиція, координація шляхом зміни цілей, координація шляхом зміни обмежень. Користуючись цими поняттями можна формально ввести поняття «модифікуємість», побудувати метод його обчислення та дослідити його необхідність і достатність для забезпечення життєздатності складної системи.

У програмній інженерії ця характеристика тісно пов'язана з поняттям реінжинірингу ПС. Вона фактично визначає ступінь придатності ПС до реінжинірингу. Характеристика «модифікуємість» визначається із сукупності показників, що вводяться в теорії реінжинірингу ПС, шляхом їх інтеграції [19].

Характеристика оптимізуємість». «Оптимізуємість» – це властивість життєздатної системи, що визначається її здатністю до рефакторингу – оптимізації функціонування, або покращувати характеристики функціонування без зміни її функціональності.

Поняття рефакторингу відображає реальну ситуацію для багатьох областей діяльності у зв'язку з технічним прогресом у комп'ютеризації та програмній інженерії.

Наприклад, питання децентралізації обчислень і заміна мейнфреймовських комп'ютерів на мережу локальних обчислювальних машин. Воно може розглядатися як проблема досягнення оптимуму в організації процесу комп'ютерних обчислень. Однак, як вказує теорія систем, цілі децентралізованої моделі обчислень досягаються лише в тому випадку, якщо вона забезпечує узгодженість глобальних і локальних цілей. Хоча кожний «обчислювач» намагається покращити свій процес, одночасно покращуються характеристики ПС в цілому, так що в результаті майже кожний виграє.

Якраз ця узгодженість цілей, виражена в постулаті сумісності теорії систем, служить реальним виправданням децентралізації обчислень.

Характеристика «оптимізуємість» тісно пов'язана з такими математичними поняттями теорії систем, як: координованість, сумісність, еталонне керування, принцип оцінки взаємодій, локальна функція затрат. Вони дозволяють формально ввести це поняття, побудувати метод його обчислення та дослідити його необхідність і достатність для забезпечення життєздатності складної системи.

Характеристика «оптимізуємість» у програмній інженерії базується на інтег-

рації сукупності показників, що визначаються в теорії рефакторингу ПС.

Метрики оцінки показників якісного характеру. В програмній інженерії метрики традиційно використовуються для попередження невірною проектування на ранній стадії життєвого циклу ПС. Знайдені невідповідності та дефекти можуть бути модифіковані та попереджені з меншими затратами коштів та зусиль ніж на пізніших етапах проектування або на стадії супроводу. Метрики забезпечують розробнику швидкий зворотній зв'язок – шляхом аналізу зібраних даних можна прогнозувати життєздатність ПС. При відповідному використанні метрик можливе значне зменшення вартості всієї розробки та покращення якості кінцевого програмного продукту, що в свою чергу, веде до зменшення витрат на його підтримку.

Метрики застосовуються в контексті сценаріїв, які реалізуються за допомогою представлення процесу (діаграм взаємодії у середовищі Rational Rose). Це дозволить оцінювати параметри системи не загалом, а в рамках деяких задач або процесів, що цікавлять нас особливо. Таким чином метрики будуть застосовуватися не до всіх класів та їх методів а лише до тих, що пов'язані спільним сценарієм.

Деякі конкретні метрики для показників якісного характеру та приклади їх застосування описано в [20].

Характеристики життєздатності економічного характеру. Характеристики економічного характеру в схемі створення життєздатних ПС мають велике значення. Побудовані системи мають бути доступні для покупки Замовником та в майбутньому забезпечити дохід інвестора.

Найбільш важливими економічними характеристиками ПС для управління при розробленні ПС є розмір проекту, трудомісткість, час, необхідний для виконання та вартість. Серед цих характеристик три останні у великій мірі залежать від першої характеристики – розміру ПС [21].

Впровадження прикладної ПС у організації можна розглядати як інвестиційний проект, у ході якого замовник вит-

рачає певні кошти на придбання активу, який дозволяє зменшити видатки через оптимізацію бізнес-процесів або збільшити надходження, запропонувавши ринку нові продукти.

Прикладна ПС є життєздатною, якщо грошові потоки – Cash Flow (CF), що генеруються її проектом, забезпечують дві умови – покриття початкової інвестиції та віддачу на вкладені гроші (дохід інвестора).

Найбільш вживаними (основними) показниками ефективності інвестиційних проектів є такі [22]: *чиста приведена вартість* – відображає різницю між видатками та прибутками, коли всі кошти приведені у часі до теперішнього моменту часу; внутрішня норма прибутку – так називають ставку дисконтування, за якої величина чиста приведена вартість стає нульовою; період окупності – час, необхідний для того, щоб сума, інвестована в той чи інший проект, повністю повернулася за рахунок коштів, одержаних в результаті основної діяльності за даним проектом; приведений період окупності враховує „часову вартість” грошей. Через такі фактори, як інфляція, банківський процент або інші привабливі проекти, одна гривня сьогодні коштує більше, ніж одна гривня завтра; коефіцієнт прибутковості інвестицій – прибуток до виплати процентів і податків, поділений на суму довготермінових зобов'язань плюс акції.

Безумовно, на економічні показники (а отже життєздатність) прикладних ПС впливають показники ефективності генеруючого середовища, з допомогою якого створювалася прикладна ПС.

Метрики оцінки економічних показників. Оскільки з чотирьох основних економічних характеристик розроблення ПС (розмір проекту, трудомісткість, час, необхідний для виконання та вартість) три останні у великій мірі залежать від першої характеристики – *розміру* ПС – оцінювання цих характеристик можна розбити на два етапи:

- оцінювання розміру;
- оцінювання трудомісткості, вартості та тривалості з використанням уже отриманого значення розміру.

Для розрахунку оцінок трьох останніх характеристик добре підходить модель COCOMO [23], оскільки вона розроблялася саме з метою оцінки бюджету, вартості та тривалості проектів та є нині найбільш широко застосовуваною моделлю. Модель COCOMO використовує як вхідний параметр *розмір* ПС у рядках вихідного коду (SLOC [24]). Отримати точний розмір у таких одиницях можна лише після завершення проекту. На будь-якому попередньому етапі розмір має визначатися за допомогою експертних оцінок або за допомогою деякої методики оцінювання розміру проекту, яка може дати оцінку в одиницях SLOC. Водночас найкращим вибором для оцінювання розміру проекту є метод FPA [24]. Він має об'єктивний характер, простий у застосуванні, є найбільш широко застосованим у індустрії розробки програмного забезпечення, має статус міжнародного стандарту та є загально доступним. Розмір ПС, отриманий за допомогою FPA, можна використати у моделі COCOMO, оскільки для багатьох мов програмування існують залежності між розміром ПС у одиницях UFP та одиницях SLOC. Існуючі методи експертних оцінок також можуть бути включені в схему, оскільки застосування більш формалізованих методів у певних випадках є неможливим або невигідним.

У поєднанні з методами визначення характеристик якості системи за UML - діаграмами це дасть нові можливості для керування проектами з розробки, модифікації та реінжинірингу ПС.

При отриманні оцінок економічних характеристик ПС, що не задовольняють користувача із-за їх великих значень, необхідно повернутися до етапу моделювання та оцінювання характеристик життєздатності ПС, понизити вимоги до характеристик життєздатності, одержати компромісний проект та одержати задовільні оцінки його економічних характеристик.

Оцінювання характеристик, важливих в економічному аспекті, при виконанні проектів за розробкою та модифікацією ПС є необхідною складовою процесу керування розробкою та супроводженням

життєздатних ПС. Застосування методів оцінювання економічних характеристик дозволяє більш точно прогнозувати як матеріальні ресурси, необхідні для реалізації проекту, так і людські та часові ресурси. Це дає можливість приймати обгрунтовані рішення щодо формування пакету нових розробок ПС в організації, а також щодо доцільності проектів модифікацій супроводжуваних ПС.

Для здійснення оцінювання характеристик життєздатності ПС відповідні необхідні дані мають бути відображені у UML – моделі ПС [25].

4. Моделювання та забезпечення життєздатності створюваних членів сімейства ПС

Моделювання та забезпечення життєздатності створюваних ПС сімейства в умовах генеруючого програмування здійснюється в рамках інтегрованої моделі забезпечення життєздатності доменів та застосувань, яка представлена на рис. 1. Відповідно до цієї моделі розглядаються наступні технологічні аспекти використання моделі життєздатності ПС у середовищі генеруючого програмування: 1) технологічна схема генеруючого програмування; 2) технологічна схема забезпечення життєздатності доменів; 3) технологічна схема забезпечення життєздатності застосувань (рис. 2). Еволюційний процес розробки життєздатних ПС сімейства реалізується на основі взаємодії цих схем.

Технологічна схема генеруючого програмування. У рамках цієї технології застосовується конфігураційна модель, яка визначає правила об'єднання компонентів реалізації та зразків (шаблонів проектування). Конфігураційна модель базується на моделі життєздатності сімейства ПС, за допомогою якої відповідно до критеріїв життєздатності визначаються життєздатні компоненти реалізації, архітектурні стилі для формування еталонної моделі та відповідної еталонної архітектури сімейства. Відповідність системи зразків проектування (design patterns) та проблемно-залежних зразків проектування (domain-specific design patterns) зразкам предметної області (domain patterns)

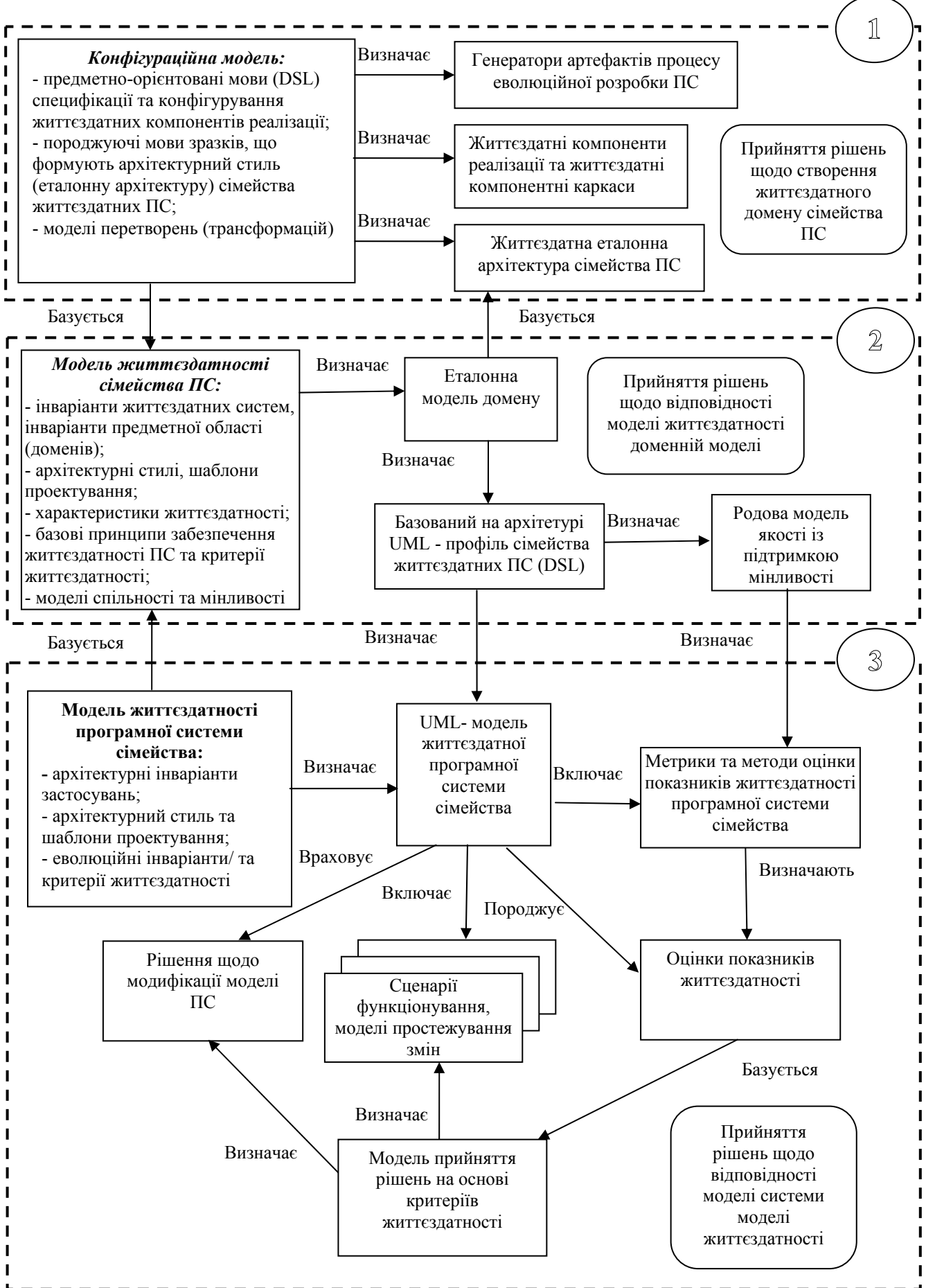


Рис. 2. Моделювання та забезпечення життєздатності створюваних ПС:

1 – технологічна схема генеруючого програмування; 2 – технологічна схема забезпечення життєздатності домену; 3 – технологічна схема забезпечення життєздатності застосувань

відображається у предметно-орієнтованій мові конфігурування та мови конфігурування компонентів реалізації відповідно до простору задач та простору рішень. За допомогою генератора може бути здійснено переклад з однієї мови на іншу [16]. Певні механізми підтримки життєздатності можуть бути представлені також за допомогою аспектичних зразків [26].

Таким чином, конфігураційна модель у рамках технології генеруючого програмування забезпечує та підтримує розробку:

- життєздатної еталонної архітектури сімейства ПС;
- генераторів артефактів процесу еволюційної розробки ПС;
- життєздатних компонентів реалізації та життєздатних компонентних каркасів.

Розробка здійснюється на основі моделі прийняття рішень щодо створення життєздатного домену сімейства ПС.

Технологічна схема забезпечення життєздатності доменів. У рамках цієї технології застосовується модель життєздатності сімейства ПС, на якій базується як конфігураційна модель технології генеруючого програмування, так і модель життєздатності ПС сімейства технології забезпечення життєздатності застосувань.

Модель життєздатності визначає наступні підходи до створення життєздатних ПС на основі інваріантів:

підхід на основі архітектурних інваріантів, яким відповідають інваріанти життєздатних систем (інваріанти VSM Біра, інваріанти LST Міллера тощо);

підхід на основі інваріантів предметної області, для визначення і підтримки яких використовуються:

- модель якості, узагальнені сценарії, критерії життєздатності на основі оцінки характеристик життєздатності, включаючи економічні показники життєздатності, та механізми, що забезпечують якість життєздатності;
- модель внутрісистемної та міжсистемної спільності, мінливості та залежностей (обмежень) на основі інваріантів і

механізми внесення змін (механізми мінливості) та підтримки еволюції;

- модель прийняття рішень та механізми інтелектуалізації і зворотного зв'язку.

Модель життєздатності сімейства ПС підтримує розробку:

- еталонної моделі домену, яка визначає архітектуру життєздатних програмних систем сімейства;
- базований на архітектурі UML – профіль сімейства життєздатних ПС, який розширює мета-модель UML і розглядається як предметно-орієнтована мова (DSL);
- родової моделі якості із підтримкою мінливості на основі шаблонів проектування та узагальнених сценаріїв.

Розробка здійснюється на основі моделі прийняття рішень щодо відповідності моделі життєздатності доменній моделі.

У рамках взаємодії технології породжуючого програмування та технології забезпечення життєздатності доменів визначаються інваріанти сімейства систем, які відповідають певній моделі життєздатності. Поняття предметної області, які ставляться їм у відповідність можна розглядати як інваріантні. Вони мають бути згруповані в рамках обмеження, яке відповідає організації системи (інакше – певному архітектурному стилю). Архітектурний стиль забезпечує необхідну якість сімейства систем у цілому та окремих членів сімейства на основі родової моделі якості (generic quality model) та моделі простежування змін, а також підтримує модель (інакше – узгоджується з моделлю) мінливості (variability model) сімейства систем відповідно до змінюваних параметрів (variation points). Останні визначають місця, де відбувається мінливість.

Відповідно до моделювання життєздатної ПС сімейства можна виділити наступні категорії інваріантних понять:

- організаційно-структурні поняття, які визначають структуру системи та її організацію відповідно до певної моделі життєздатної системи або відповідно до контексту предметної області;

- композиційні поняття, які забезпечують розвиток системи (еволюцію) на основі певних механізмів (рекурсія, наслідування тощо);

- інтероперабельні поняття, які забезпечать гомеостатичну рівновагу щодо взаємодії системи із зовнішнім середовищем (зовнішніми системами).

Технологічна схема забезпечення життєздатності застосувань. У рамках цієї технології застосовується модель життєздатності ПС, яка базується на моделі життєздатності сімейства ПС. Ця технологія підтримується засобами прикладної інженерії (інженерії ПС). Члени сімейства створюються на основі базованого на архітектурі UML – профілю сімейства життєздатних ПС, а модель якості, яка застосовується для оцінки життєздатності ПС відповідає родовій моделі якості на основі шаблонів проектування, сценаріїв та моделі простежування змін.

Модель життєздатності ПС сімейства підтримує розробку:

- моделі життєздатної програмної системи сімейства (UML - модель);
- узагальнених сценаріїв функціонування ПС;
- моделей простежування змін для основних сценаріїв функціонування ПС;
- показників життєздатності ПС;
- метрик та методів оцінки показників життєздатності ПС.

Розробка здійснюється на основі моделі прийняття рішень щодо відповідності моделі системи моделі життєздатності.

Висновки

Досліджено технологічні аспекти підходу до створення життєздатних програмних систем на основі інтегрованої моделі забезпечення життєздатності доменів та застосувань.

Підхід базується та доповнює метод генеруючого програмування, який в даний час є ключовим у реалізації одного із основних принципів програмної інженерії, а саме забезпечення її розвитку в напрямку індустріалізації створення сімейства ПС і який нині розробляється в Інституті програмних систем НАН України.

1. *Разумовський О.С., Хазов М.Ю.* Проблема жизнеспособности систем // Гуманитарные науки в Сибири. – 1998. – № 1. – С. 3 – 7.
2. *Miller J.G.* Living Systems. Niwot, Colorado: University Press of Colorado, 1995. – 1102 p.
3. *Бур С.* Мозг фирмы. – М.: Радио и связь, 1993. – 218 с.
4. *Эшби У.Р.* Введение в кибернетику. – М.: Наука, 1975. – 427 с.
5. *Herring C.* Viable software. The intelligent control paradigm for adaptable and adaptive architecture. – 2002. – 343 p. <http://charles-herring.com/public/Viable Software.pdf>
6. *Ігнатенко П.П., Неумоїн В.М., Бистров В.М.* Про забезпечення ефективного реінжинірингу прикладних програмних систем // Проблеми програмування. – 2001. – № 1-2. – С. 42–52.
7. *Підхід до забезпечення реінжинірингу об'єктно-орієнтованих програмних систем / П.П. Ігнатенко, В.М. Бистров, І.О. Заєць, О.П. Ігнатенко // Там же. – 2002. – № 1-2. – С. 98–108.*
8. *Ігнатенко П.П., Стрелов І.А., Ткаченко В.М., Дуднік Р.О.* Концепція створення моделі прикладної програмної системи з розвинутою функцією життєздатності // Проблеми програмування. – 2004. – № 2-3. – С. 163–172.
9. *Ігнатенко П.П., Бистров В.М., Ігнатенко О.П., Ткаченко В.М.* Задачі та засоби моделювання і оцінювання життєздатних програмних систем // Проблеми програмування. – 2003. – № 3. – С. 59–70.
10. *Ігнатенко П.П.* Проблеми забезпечення життєздатності програмних систем та підходи до їх вирішення // Проблеми програмування. – 2002. – № 3-4. – С. 58–73.
11. *Ігнатенко П.П., Ткаченко В.Н., Стрелов І.А., Дуднік Р.А.* Підход к моделированию и проектированию CRM-систем // УСиМ. – 2005. – № 2. – С. 57 – 65.
12. *Ігнатенко П.П.* Життєздатні програмні системи. Концептуалізація підходу до автоматизації систем організаційного керування // Проблеми програмування. – 2006. – № 3. – С. 33–44.
13. *Лаврищева К.М.* Методы программирования, теория, инженерия, практика // К.: Наук. думка, 2006. – 451 с.
14. *Ігнатенко П.П., Бистров В.М.* Особливості забезпечення життєздатності програмних систем в умовах генеруючого програмування // Проблеми програмування – 2008. – № 2-3. – С. 270–278.

15. *Software Reengineering* // Ed. Roberts, S. Amold // IEEE Comp. Soc. Press. – 1994. – 676 p.
16. Чернецки К., Айзенкер У. Порождающее программирование // Методы, инструменты, применение. – М., С-Пет., Изд. дом «Питер», 2005. – 730 с.
17. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем. – М.: Мир, 1973. – 344 с.
18. Неумойн В.М., Игнатенко П.П. Аспекты адаптивности программных систем // Пр. Першої міжнар. наук.-практ. конф. УкрПРОГ'98. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України, 1998. – С. 542–546.
19. Игнатенко П.П., Неумойн В.М., Бистров В.М. Аспекти реінжинірингу програмних систем // Проблеми програмування. – 2000. – № 1-2. – С. 367–375.
20. Ткаченко В.М. Методика оцінювання деяких характеристик якості прикладних програмних систем // Проблеми програмування. – 2006. – № 4. – С. 16–27.
21. Стрелов І.А., Игнатенко П.П. Підхід до оцінювання економічних характеристик проектних рішень при розробці, модифікації та реінжинірингу програмних систем // Проблеми програмування. – 2004. – № 1. – С. 38–51.
22. Стрелов І.А., Игнатенко П.П. Підхід до прогнозування основних характеристик економічної ефективності ППС // Проблеми програмування. – 2007. – № 4. – С. 13–20
23. CSE, 1999 – Center for Software Engineering, "COCOMO II Reference Manual," Computer Science Department, USC Center for Software Engineering, 1999. – 86 p.
24. COSMIC, 2003 – The Common Software Measurement International Consortium, The COSMIC FFP Measurement Manual. Version 2.2. – 81p. // www.cos-micon.com.
25. Стрелов І.А., Игнатенко П.П. Ідентифікація та відображення функціональних елементів FPA – методу в UML – моделі створюваної системи для оцінювання її економічних характеристик // Проблеми програмування. – 2005. – № 3. – С. 67–76.
26. *Aspect-oriented Domain-specific modelling: A Generative Approach, using a Metaweaver framework.* – 2004.

Про авторів:

Игнатенко Петро Петрович,
кандидат технічних наук,
завідувач відділу,

Бистров Віктор Михайлович,
кандидат фізико-математичних наук,
старший науковий співробітник.

Місце роботи авторів:

Інститут програмних систем
НАН України,
03187, Київ - 187,
Проспект Академіка Глушкова, 40.
Тел.: 526 6025.
e-mail: o.ignatenko@gmail.com

Отримано 03.03. 2010