

УДК 65.012.123

О. Є. ФЕДОРОВИЧ¹, О. В. БАБИЧ²¹ *Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ», Україна*² *Полтавський політехнічний коледж НТУ«ХПІ», Україна*

МЕТОД БАГАТОРІВНЕВОГО КОМПОНЕНТНОГО ПРОЕКТУВАННЯ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ СТВОРЮВАНОЇ ПРОГРАМНОЇ СИСТЕМИ

Ставиться та вирішується завдання забезпечення якості під час створення складної програмної системи (ПС) шляхом розвитку компонентного підходу на основі формування багаторівневої компонентної архітектури ПС. Виділяються типи компонент для забезпечення реалізованості проекту та якості ПС, що розробляється. Особливу увагу приділено компонентам повторного використання, які дозволяють знизити ризики проектування та підвищити якість з метою забезпечення конкурентоспроможності ПС. Запропоновано системне представлення та модель багаторівневого компонентного проектування ПС. Обґрунтовано використання високорівневої технології проектування на основі P-Modeling Framework (P-FM) для забезпечення якості створення ПС.

Ключові слова: багаторівневе компонентне проектування, забезпечення якості програмної системи, базова архітектура, компоненти повторного використання.

Вступ

Сучасні технології створення програмного забезпечення (ПО) спрямовані на забезпечення якості проєктованих програмних систем (ПС). Формування та вимір показників якості пов'язано з вимогами кваліметрії, в якій якість розглядається через вимоги користувача до створюваного «продукту». В залежності від вимог користувачів до «продукту» (ПС) можна використовувати різні типи показників якості (надійність, рівень якості і т. д.), а також види їх вимірювань й оцінки (кількісна шкала, якісна шкала, безвимірна шкала і т. д.). Найчастіше показники якості мають суперечливий характер (рівень якості, витрати на забезпечення якості і т. д.), що необхідно враховувати для забезпечення якості при розробці ПС.

Проблема якості вирішується шляхом впровадження нових технологій проектування, у яких керування проектом здійснюється з урахуванням архітектури створюваної ПС. Складність архітектури привела до необхідності використання компонентного підходу, в якому використовується позитивний досвід минулих розробок ПС, що підвищує реалізованість проекту, якість розробки ПС і скорочення строків проектування [1]. Тому тема запропонованої публікації, у якій розглядається розвиток компонентного підходу шляхом створення багаторівневої компонентної архітектури ПС і використання методів проектування, що забезпечують якість ПС, є актуальною.

Постановка завдання дослідження

Сучасні ПС є складними багаторівневими системами. Метою проекту зі створення ПС є розробка складного продукту з архітектурою у вигляді ієрархії структурних елементів, в числі яких можна виділити як порівняно прості, так і складні компоненти. Прості компоненти є елементами нижнього рівня деталізації ПС (наприклад, утиліти – статичні класи або бібліотеки), а складні – елементами середніх і верхнього рівнів (системи, підсистеми, фреймворки). Тож зазвичай всі управлінські рішення проекту зі створення ПС зосереджені на отриманні багаторівневої компонентної архітектури.

Тож життєвий цикл (ЖЦ) проекту зі створення ПС на основі багаторівневої компонентної архітектури можна представити наступним чином. В ході виявлення вимог замовника формується технічне завдання (ТЗ) або у версії Microsoft Solutions Framework (MSF) – документ який описує бачення системи, в якому викладені основні вимоги і характеристики створюваної програмної системи [2]. В тому числі, документ містить інформацію, необхідну для формування компонентної архітектури ПС. На цьому завершується перший етап моделі процесів MSF – формування концепції системи. Використовуючи отримані в результаті цього етапу дані, команда проекту переходить до етапу управління процесом створення ПС (внутрішнє проектування), або у відповідності до MSF – етапу планування.

Зазвичай на цьому етапі використовується методика планування виконання проекту «зверху вниз», або структурна декомпозиція завдань WBS [3]. Вона полягає у формуванні високорівневої компонентної архітектури, яка поступово деталізується, з подальшим плануванням термінів виконання шляхом зворотного відліку від кінцевого терміну завершення проекту. Це дає змогу оцінити терміни завершення мікропроектів – робіт, пов'язаних з окремими компонентами проекту, з урахуванням витрат часу на розробку та тестування (стабілізацію проекту за MSF) та з огляду на ієрархічну компонентну архітектуру створюваної системи.

Етап стабілізації компонент ПС є одним з найвідповідальніших, оскільки в ході цього етапу виконуються функціональне, інтеграційне та інші види тестування, що може виявити можливу невідповідність до вимог, сформульованих на етапі формування концепції ПС (порушення якості), з чим пов'язана значна доля проектних ризиків. Додаткові ризики можуть бути оцінені шляхом відстеження міжкомпонентних зв'язків на ранніх етапах проекту.

Завершальним етапом проекту є розгортання системи – пілотне впровадження з метою верифікації та приймального тестування багатоконпонентної ПС в умовах інфраструктури замовника, по завершенні якого підписується документ про завершення проекту. Результати проекту передаються замовнику, здійснюється постпроектний аналіз, життєвий цикл проекту завершується. Звісно, завжди є можливість наступних ітерацій проекту, які мають таку ж структуру і виконуються як новий проект.

Вирішення завдання дослідження

Компоненти, які використовуються під час створення архітектури складної ПС можна поділити на три основних типи:

- повторно використовувані компоненти (ПВК), які було створені командою у ході попередніх проектів, або компоненти, які входять до складу стандартних бібліотек та фреймворків;
- нові компоненти (НК), які створюються для задоволення вимог конкретного проекту та можуть забезпечувати його інноваційність та привабливість;
- складні компоненти (СК) або підсистеми чи фреймворки, отримані шляхом групування інших компонент (в тому числі ПВК та НК).

Аналіз існуючих рішень свідчить про те, що сучасна ПС може містити 50% або навіть більше ПВК. З одного боку це полегшує процес проектування, зменшує працевитрати, фінансові витрати, дозволяє виключити певні ризики за рахунок використання відлагоджених, якісних та надійних компонентів, скорочує терміни проектування. Проте, використати лише наявні рішення, не завжди мож-

ливо, оскільки нова ПС може вимагати вирішення нових функціональних завдань, які з допомогою ПВК можна вирішити далеко не завжди. Тож зазвичай проектна команда стикається з проблемою створення нових компонент, що може призвести до продовження циклу розробки, вимагає додаткових витрат і збільшує ризики.

З іншого боку, нові компоненти здатні забезпечити конкурентоспроможність проекту та можливість розширення кола замовників і споживачів на ринку ПС за рахунок їх повторного використання у майбутніх проектах. Нові компоненти повинні пройти повний цикл розробки, перш ніж потрапити до складу створюваної ПС, хоч зазвичай їх створення відбувається паралельно. Досить часто вони містять інноваційні рішення, але разом з тим, додають нові ризики щодо можливості їх розробки та задоволення вимог з якості, що впливає на можливість реалізації проекту в цілому.

Використання повторно використовуваних компонент дозволяє застосувати в проекті готові, перевірені практикою минулих проектів, проектні рішення, які можна використовувати для формування платформи для створення ПС.

Для складної компоненти ПС ризики, пов'язані з її створенням, зменшуються шляхом використання в її складі ПВК. Тому для складної багаторівневої архітектури ПС запропонований підхід дозволяє скоротити тривалість етапів проекту від планування компоненти до інтегрування її до складу ПС, що призводить до скорочення терміну виконання проекту в цілому і задоволення вимог якості в ході розробки складної ПС.

Використання компонентного підходу передбачає спеціалізацію компонент на всіх рівнях архітектури ПС, певну реорганізацію процесу проектування, що, в поєднанні з методиками оцінки та забезпечення якості проектування, на кшталт запропонованої методики P-Modeling Framework, застосовної як в процесі проектування окремих компонент, так і для проекту в цілому, в решті-решт забезпечує високу якість результатів проектування [4].

Уявімо архітектуру складної ПС у вигляді багаторівневої ієрархічної компонентної структури, кожен з елементів якої може являти собою ПВК, НК, СК. Аналіз можливості реалізації і оцінку якості такого проекту можна здійснити, використавши згортання горизонтальної та вертикальної проєкції компонентної архітектури ПС.

Горизонтальний напрям згортання є згортанням за етапами життєвого циклу (ЖЦ) компонент і всього проекту зі створення ПС в цілому. Згортання може здійснюватись на основі досвіду команди з обов'язковим урахуванням оцінки якості для кожного етапу життєвого циклу. Надалі ця оцінка уточню-

ється в процесі здійснення кожного конкретного етапу ЖЦ створюваної програмної системи.

Вертикальний напрям згортання, пов'язаний з деталізацією складної багаторівневої компонентної ПС, призначено для оцінки можливості реалізації проекту з урахуванням групування компонент за рівнями архітектури ПС.

Вимоги до створення нової ПС можна укрупнено розбити на три групи: економічні, організаційні та технологічні. Економічні вимоги пов'язані з оптимізацією витрат під час розробки нової ПС з урахуванням активного використання багаторівневого компонентного підходу (пошук оптимального за витратами багаторівневого складу компонентної архітектури ПС).

Організаційні вимоги пов'язані зі створенням нової організаційної структури управління проектом, до складу якої входить команда проектувальників, які формують базову архітектуру (платформу) ПС, а також проектувальників, які адаптують базову багатоконцентну архітектуру до конкретного проекту зі створення ПС.

Технологічні вимоги пов'язані зі створенням нової технології проектування, заснованої на багаторівневому компонентному підході і інтеграційній моделі проектування ПС. Для організації процесу проектування складної ПС та контролю якості проектування на всіх його етапах пропонується використовувати методологію P-Modeling Framework (P-MF), яка підтримує організаційне управління і технологічний процес проектування із забезпеченням якості створюваних програмних продуктів. В ході створення нової ПС множина повторно використовуваних компонентів може поповнюватись новими елементами, які виникають внаслідок задоволення нових вимог. Основна увага при цьому концентрується на створенні багаторівневої компонентної архітектури та виділенні множини повторно використовуваних компонентів. Таким чином, поява нової багаторівневої компонентної моделі розробки докорінно змінює процес створення складної ПС. Тож з метою його ефективної реалізації необхідно інтегрувати, в рамках багаторівневої архітектури, високорівневу організаційну модель управління проектом, технологічний процес розробки на базі P-MF, і забезпечення якості створюваної ПС. Запропоновану модель проектування можна представити у вигляді трьох складових, тісно пов'язаних між собою: організаційна структура управління проектом, компонентна архітектура і технологія проектування P-MF для забезпечення якості розробки ПС (рис. 1). На перетині цих трьох складових отримуємо зміни у складі проектної команди, необхідність виконання зворотного семантичного трасування (ЗСТ) для забезпечення якості на кожному етапі

проектування [4, 5], а отже й ітеративний характер процесу.

Процес забезпечення якості в ході створення багаторівневої компонентної ПС доцільно представити у вигляді системного куба з виділенням наступних проєкцій (рис. 2):

- рівні якості;
- рівні компонентної архітектури ПС;
- склад множини компонент ПС.

В рамках існуючих традиційних організаційних схем керування проектом зі створення нової ПС, проектні завдання зазвичай виконуються без здійснення глибокого аналізу багаторівневої архітектури ПС і виділення компонент. Досвід минулих розробок може використовуватись інтуїтивно, без явного використання ПВК. Одну з традиційних схем (каскадна модель) проектування представлено на рис. 3.



Рис. 1. Модель багаторівневого компонентного проектування ПС

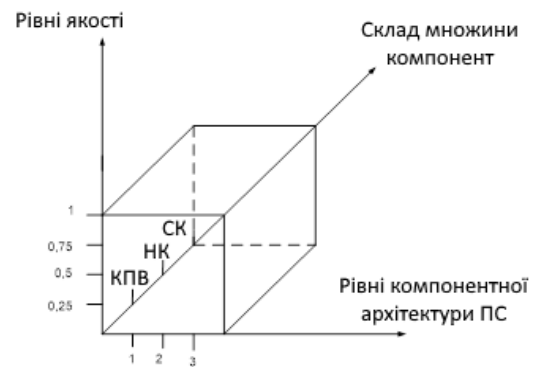


Рис. 2. Системний куб створення багаторівневої компонентної ПС

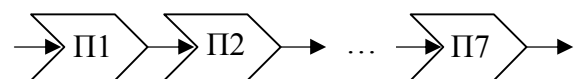


Рис. 3. Каскадна модель проектування ПС

У складі цієї моделі можна виділити наступні етапи проектування:

- П1 – формування вимог до системи;
- П2 – системне проектування та формування

архітектури ПС;

П3 – розробка нових елементів, використання або модифікація існуючих;

П4 – інтеграція елементів ПС;

П5 – стабілізація (тестування) ПС;

П6 – пілотне впровадження;

П7 – розгортання (впровадження).

Недоліком такого підходу є відсутність явного виділення старих, нових і складних елементів ПС у вигляді закінчених компонент та інтуїтивні процеси створення, часто – проектування «з нуля».

В рамках запропонованого багаторівневого компонентного підходу спочатку здійснюється вивчення та аналіз вимог до ПС, потім – формування множини компонент, які будуть використані на всіх рівнях архітектури майбутньої ПС та які можна використати в новому проекті задля забезпечення необхідного рівня функціональності, визначення нових компонент і формування їх ЖЦ, організаційне керування процесом інтеграції на базі технології P-MF задля створення компонентної багаторівневої архітектури ПС.

Діючи у відповідності до такого підходу, необхідно заздалегідь сформулювати базову архітектуру ПС, а потім, шляхом адаптації ПВК і розробки нових компонент, переходити до задоволення конкретних вимог замовника. Вибір і формування множини ПВК здійснюється командою проектувальників, до складу якої, окрім архітекторів програмних систем, входять також досвідчені розробники, спеціалісти з тестування та впровадження, які виступають в якості експертів на системному етапі проектування ПС.

На рис. 4 показано запроповану схему створення складної ПС, засновану на багаторівневому компонентному підході і технології P-MF, яка забезпечує контроль та підтримку належного рівня якості проектування. Тут виділено два процеси проектування:

- процес створення базової архітектури ПС (верхній рівень);
- процес реалізації конкретних вимог до ПС (нижній рівень).

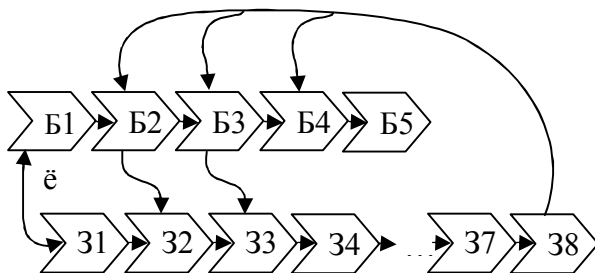


Рис. 4. Багаторівневий компонентний підхід до проектування складної ПС

На схемі представлено такі окремі завдання, виконувані в процесі проектування:

Б1 – аналіз вимог до ПС;

Б2 – аналіз і узагальнення досвіду минулих проектів з розробки ПС;

Б3 – створення базової багаторівневої компонентної архітектури ПС;

Б4 – формування множини програмних компонент на всіх рівнях архітектури ПС;

Б5 – вдосконалення багаторівневої компонентної архітектури з урахуванням нових вимог;

31 – формування вимог до конкретного проекту;

32 – адаптація базової багаторівневої компонентної архітектури ПС для задоволення конкретних вимог;

33 – адаптація множини ПВК базової архітектури ПС для конкретних вимог;

34 – створення множини нових компонент (НК) для конкретних вимог з використанням технології P-MF для контролю та забезпечення якості проектування окремих компонент;

35 – групування та інтеграція компонент в архітектуру створюваної ПС з використанням технології P-MF для контролю та забезпечення якості процесу в цілому;

36 – стабілізація ПС;

37 – пілотне впровадження;

38 – розгортання створеної ПС.

Уявімо архітектуру складної ПС у вигляді ієрархічно пов'язаних між собою рівнів, на кожному з яких знаходиться множина компонент L (ПВК, НК, СК):

$$L = \{L_1, L_2, \dots, L_n\},$$

де L_1 – відповідає верхньому, системному, рівню ПС;

L_n – відповідає найнижчому рівню (рівень платформи).

Для i -го рівня архітектури, в процесі проектування формується множина компонент (ПВК, НК, СК), які групуються та інтегруються до складу створюваної ПС:

$$P_i = \{P_{i_1}, P_{i_2}, \dots, P_{i_m}\}.$$

Системну модель створення складної багаторівневої компонентної ПС, з задоволенням вимог щодо забезпечення належного рівня якості, можна представити у вигляді кортежу:

$$S = \{T, KPI, NK, SK, B, MKT, IP, VT, SV, Q\},$$

де T – вимоги замовника;

KPI – множина ПВК на всіх рівнях архітектури

ПС;

НК – множина нових (створюваних) компонент на всіх рівнях ПС;

СК – множина складних компонент ПС;

В – базова архітектура ПС;

МКТ – компонентна технологія створення ПС;

ІР – інтеграційний процес формування ПС з врахуванням групування компонент;

VT – високорівневе технологічне проектування на базі P-MF задля забезпечення якості розробки ПС;

SV – організаційна структура керування проектом;

Q – задоволення вимог щодо забезпечення належного рівня якості на кожному етапі розробки ПС.

Побудова подібних моделей підвищує рівень абстракції опису процесу створення складної багаторівневої компонентної системи, що дозволяє не лише формалізувати цей процес, але й полегшити тестування складних ПС[6]. Наступним кроком є побудова формальної моделі якості створюваної системи у вигляді вектору, складові якого корелюють і є підмножиною характеристик якості за стандартом ISO/IEC 25010 [7], формулювання й вирішення завдань оптимізації як окремих показників, так і у комплексі, а також подальша інтеграція методології забезпечення якості P-Modeling Framework в процес проектування багаторівневої архітектури ПС.

Висновок

Запропонований підхід за допомогою багаторівневого компонентного проектування ПО є розвитком компонентного підходу. Використання компонентного підходу дозволяє виділити та формалізувати досвід минулих розробок, що зарекомендували себе, у вигляді компонент повторного використання (КПВ) в складі ПС, які априорі задовольняють вимогам якості і тим самим забезпечують мінімізацію ризиків при проектуванні ПС, а також позитивно впливають на якість всього проекту зі створення ПС. Все це дозволяє підвищити реалізованість проекту, забезпечити якість при створенні ПС за рахунок використання P-Modeling Framework на всіх етапах проектування ПС, скоротити строки розробки й зробити програмний продукт конкурентоспроможним на ринку споживачів ПЗ.

Література

1. Verhoef, C. *Evolutionary software engineering: a component-based approach [Text]* / C. Verhoef, P. Klint // *IFIP TC2 WG2.4 Working Conference on Systems Implementation 2000: Languages, methods and tools 23–26 February 1998, Berlin, Germany.* – Springer, 2016. – P. 1-18.

2. Бабич, А. В. *Использование элементов Microsoft Solutions Framework в курсовом проектировании [Текст]* / *International Forum of Educational Technology & Society // Educational Technology & Society by IEEE.* – ISSN: 1436-4522. – 2005. – № 8(3). – С. 367-379.

3. *Is There Still a Room For Programmers' Productivity Improvement? [Text]* / L. Pavlov, S. Busygin, N. Boyko, A. Babich // *International Software And Productivity Engineering Institute, University of Florida.* – Yerevan : 5th annual IEEE East-West Design and Test Symposium, 2007. – P. 146-151.

4. *Using Reverse Semantic Traceability for Quality Control in Agile MSF-based Projects [Electronic resource]* / K. Zhereb, V. Pavlov, A. Doroshenko, V. Sergienko // *International Software & Productivity Engineering Institute (INTSPEI).* – Moscow, Russia: 4th Software Engineering Conference, October 23, 2008: Proceedings. – Mode of access: WWW.URL: http://2008.cee-secr.org/ru/etc/secr2008_konstantin_zhereb_intspei_using_rst.pdf. – Last access: March 11, 2016. – Title from the screen.

5. *Applying Pantomime and Reverse Engineering Techniques in Software Engineering Education [Text]* / V. L. Pavlov, N. Boyko, A. Babich, O. Kuchaiev, S. Busygin // *37th ASEE/IEEE Frontiers in Education Conference.* – Milwaukee, Wisconsin, USA, 2007. – P. T1E-1-T1E-5.

6. *Testing the Untestable: Model Testing of Complex Software-Intensive Systems [Electronic resource]* / L. Briand, S. Nejati, M. Sabetzadeh, D. Bianculli // *38th International Conference on Software Engineering (ICSE 2016 by ACM).* – Austin, Texas, USA, May 14-22 2016: Proceedings. – Mode of access: WWW.URL: <http://orbit.uni.lu/handle/10993/23840>. – Last access: March 11, 2016. – Title from the screen.

7. *ISO/IEC 25010 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (2010) by ISO/IEC.* – March 1, 2011. – Mode of access: WWW.URL: <http://iso25000.com/index.php/en/iso-25000-standards/iso-25010> – Last access: March 10, 2016. – Title from the screen.

References

1. Verhoef, C., Klint, P. Evolutionary software engineering: a component-based approach. *IFIP TC2 WG2.4 Working Conference on Systems Implementation 2000: Languages, methods and tools*, 23–26 February 1998, Berlin, Germany, Springer Publ., 2016, pp. 1-18.

2. Babich, A. V. *Ispol'zovanie jelementov Microsoft Solutions Framework v kursovom proekti-rovanii [Using the Microsoft Solutions Framework elements in a term projects]. International Forum of Educational Technology & Society. Educational Technology & Society (IEEE)*, 2005, no. 8(3), pp. 367-379. (In Russian).

3. Pavlov, V. L., Busygin, S., Boyko, N., Babich, A. *Is There Still a Room For Programmers' Productivity Improvement? 5th annual IEEE East-West Design and Test Symposium, Yerevan, Armenia*, 2007, pp. 146-151.

4. Zhereb, K., Pavlov, V., Doroshenko, A. and Sergienko, V., (2008). Using Reverse Semantic Traceability for Quality Control in Agile MSF-based Projects. *4th Software Engineering Conference*. [online], Moscow, Russia. Available at: http://2008.ccc-secr.org/ru/etc/secr2008_konstantin_zhereb_intspei_using_rst.pdf [Accessed 11 Mar. 2016].

5. Pavlov, V. L., Boyko, N., Babich, A., Kuchaiev, O., Busygin, S. Applying Pantomime and Reverse Engineering Techniques in Software Engineering Education. *37th ASEE/IEEE Frontiers in Education Conference*, Milwaukee, Wisconsin, USA, 2007, P. T1E-1-T1E-5.

6. Briand, L., Nejati, S., Sabetzadeh, M.,

Bianculli, D. (2016). Testing the Untestable: Model Testing of Complex Software-Intensive Systems. *38th International Conference on Software Engineering (ICSE 2016 by ACM)*. [online], Austin, Texas, USA. Available at: <http://orbilu.uni.lu/handle/10993/23840> [Accessed 11 Mar. 2016].

7. ISO/IEC 25010 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (2010) by ISO/IEC. [online], March 1, 2011. Available at: <http://iso25000.com/index.php/en/iso-25000-standards/iso-25010> [Accessed 10 Mar. 2016].

Поступила в редакцію 28.01.2016, рассмотрена на редколлегии 18.02.2016

МЕТОД МНОГОУРОВНЕВОГО КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ДЛЯ ОБЕСПЕЧЕНИЯ КАЧЕСТВА СОЗДАВАЕМОЙ ПРОГРАММНОЙ СИСТЕМЫ

О. Е. Федорович, А. В. Бабич

Ставится и решается задача обеспечения качества при создании сложной программной системы (ПС) путем развития компонентного подхода на основе формирования многоуровневой компонентной архитектуры ПС. Выделяются типы компонент для обеспечения реализуемости проекта и качества разрабатываемой ПС. Особое внимание уделено компонентам повторного использования, которые позволяют снизить риски проектирования и повысить качество для обеспечения конкурентоспособности ПС. Предложено системное представление и модель многоуровневого компонентного проектирования ПС. Обосновано использование высокоуровневой технологии проектирования на основе P-Modeling Framework (P-FM) для обеспечения качества создания ПС.

Ключевые слова: многоуровневое компонентное проектирование, обеспечение качества программной системы, базовая архитектура, компоненты повторного использования.

METHOD OF MULTILEVEL COMPONENT DESIGN TO ENSURE THE QUALITY OF CREATED SOFTWARE SYSTEM

O. Ye. Fedorovich, A. V. Babich

Posed and solved the problem of quality assurance during creating a complex software system (SS) through the development of component-based approach of forming a multi-level component architecture of the SS. Component types are refined to ensure the feasibility of the project and to ensure the quality of the SS. Particular attention is paid to reused components that can reduce the risks and improve the quality of design to ensure the competitiveness of the SS. Proposed the system view and multilevel model of the component design of the SS. Substantiated the usage of high-level design technology based on the P-Modeling Framework (P-FM) to ensure the quality of created SS.

Keywords: multilevel component design, quality assurance of the software system, the basic architecture, reusable components.

Федорович Олег Евгеньевич – д-р техн. наук, проф., зав. каф. информационных управляющих систем, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: kafius@ukr.net.

Бабич Александр Викторович – преподаватель Полтавского политехнического колледжа Национального технического университета «Харьковский политехнический институт», МСТ/OCUP Advanced, Полтава, Украина, e-mail: alexander.taurus@gmail.com.

Fedorovich Oleg Yevgenevich - Doctor of Technical Sciences, Prof., Head of Information Management Systems Department, National Aerospace University named after N. Ye. Zhukovsky "Kharkiv Aviation Institute", Kharkov, Ukraine, e-mail: kafius@ukr.net.

Babich Aleksandr Viktorovich - a Teacher of Poltava Polytechnic College by National Technical University "Kharkiv Polytechnic Institute", МСТ/OCUP Advanced, Poltava, Ukraine, e-mail: alexander.taurus@gmail.com.