Секція: ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
Голови: проф. М. Приймак, проф С. Лупенко, доц. О. Мацюк, проф. О. Пастух.
Вчений секретар: ас. Г. Шимчук

УДК 004.4

І. Боднарчук, канд. тех. наук, О. Харченко, канд. тех. наук, доц., Б. Хоміцький, Г. Шимчук
Тернопільський національний технічний університет імені Івана Пулюя

ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНИХ СИСТЕМ В ПРОЕКТАХ З ГНУЧКИМИ МЕТОДАМИ УПРАВЛІННЯ

I. Bodnarchuk, O. Kharchenko, B. Khomitskyi, G. Shymchuk

SOFTWARE SYSTEMS ARCHITECTURE DESIGN IN THE PROJECTS WITH FLEXIBLE MANAGEMENT TECHNIQUES

У зв’язку з широким впровадженням за останнє десятиліття гнучких методів проектування програмних систем (ПС), стали достатньо актуальними проблеми забезпечення їх якості. Особливістю цих методів є те, що проект ділиться на частини (спринти), в кожному з яких реалізується частина функціональності і вони розробляються окремо від формулювання вимог до розгортання. Причому вимоги до ПС можуть змінюватись як в процесі розробки, так і після розгортання, що потребує внесення змін в розроблюваний компонент. Дослідження і врахування впливу цих змін на якість ПС досить складне завдання і, як правило, воно не виконується в Agile Software Development (ASD) [1]. Використання технології TDD, та рефакторинг дозволяють лише виявити помилки та дефекти, виправляти їх, а не контролювати якість [2].

Основними перевагами гнучких методів є те що вони пристосовані до врахування постійної зміни вимог, які аналізуються і уточнюються на початку кожної ітерації. За рахунок цього, а також проведенням неперервного тестування і інтеграції, зменшуються ризики проекту, збільшується його ефективність [1], [2].

Однак, гнучкі методи мають і ряд недоліків, серед яких слід відмітити такі:

- оскільки вимоги в гнучких методах є «полегшеними» і керованими локально в межах кожної ітерації, то можуть з'явитися проблеми з їх стабілізації та узгодження при інтеграції;
- можливі також проблеми з необхідністю створення «гнучкої» архітектури, яку потрібно постійно корегувати для врахування зміни вимог.

Поєднанню процесів архітектурного проектування з процесами гнучкої розробки присвячені роботи [3], [4]. В них виділяються два етапи розробки, на першому з яких виконується архітектурне проектування і вибір базової архітектури, а на другому – реалізується процедура «гнучкої» зміни архітектури. Для забезпечення змінюваності архітектури пропонується на етапі

архітектурного аналізу виявляти «точки змін», та будувати «профілі змін» на основі відповідей всіх членів команди, причетних до розробки ПС, на запитання контрольного списку. Але внесення змін в архітектуру в ASD і оцінювання відповідності її вимогам якості відбуваються шляхом аналізу відповідей експертів на запитання «контрольного списку», що є досить суб'єктивним і неточним.

В даній роботі пропонується для поєднання архітектурного проектування і гнучких методів розробки контролювати якість зміненої архітектури обчисленням цільової функції, побудованої методом групового урахування аргументів (МГУА). Це дозволить оперативно отримати оцінки якості і прийняти рішення про продовження процедури ASD або перехід на процедуру перепроектування архітектури. Особливістю алгоритмів МГУА полягає в тому, що вид базисної функції, клас рівнянь і структура моделей встановлюється об'єктивним способом за допомогою перебору варіантів та вибору найкращого варіанту за обраними критеріями.

Оцінювання та вибір архітектури з множини альтернатив на першому етапі виконується методом аналізу ієрархій (MAI), який менш трудомісткий порівняно зі сценарними методами, добре формалізований і дозволяє автоматизувати процедури оцінювання.

На другому етапі в ітераціях ASD проводиться оптимізація змін архітектури методом «заміщення-компенсації», який дозволяє узгодити зміни архітектури зі зміною критеріїв її якості. При внесенні суттєвих змін в архітектуру виконується оцінювання і перевірка відповідності якості зміненої архітектури вимогам. При незадовільній якості відбувається перехід на операції першого етапу, перепроектування і заміна базової архітектури.

Процес поєднання архітектурного проектування з гнучкими методами створення ПС включає декілька процедур, які об'єднані в дві групи і виконуються в два етапи. На першому етапі формуються вимоги до програмної системи у відповідності зі стандартом ISO/IEC 25030. В подальшому ці вимоги комунікуються в вимоги до архітектури з використанням методу QFD, або методом базових протоколів [5, 6], визначаються критерії якості, які є оцінками виконання певних вимог.

З врахуванням вимог створюються альтернативні архітектури за технологією, яка залежить від прийнятого стилю документування (view) [7]. Для забезпечення варіативності архітектури зручним є стиль шарів в якому кожній альтернативі відповідає певний набір патернів, розміщених у відповідних шарах. Цей стиль широко використовується при розробці ПС, наприклад Microsoft, і є досить продуктивним, оскільки на даний час створено велику кількість патернів [7]. Для внесення змін в архітектуру досить змінити певний патерн або замінити його на альтернативний з існуючого репозиторію.

Для оцінювання альтернатив обчислюються їх відносні оцінки по кожному з критеріїв якості з використанням модифікованого (MAI). Вибір кращого варіанта архітектури виконується методом аналізу компромісів або по значенню інтегрального показника якості [6], [7], [8].

На другому етапі вибраний варіант архітектури впроваджується в гнучкому методі розробки. В разі виявлення нових вимог до ПС, або зміни існуючих, вносяться відповідні зміни в архітектуру. Зміни виконуються шляхом корегування коду відповідного патерну або його заміною на альтернативний. Для дослідження, та оптимізації впливу цих змін на якість проводиться корекція значень критеріїв якості. Процедура корекції розроблена на основі застосування методу «заміщення – компенсації», в якому збільшення деяких критеріїв якості відбувається за рахунок зменшення інших, що дозволяє оптимізувати ці зміни і не виходити за межі бюджету проекту.

Для оперативної оцінки зміненої архітектури можна використати попередньо побудовану оціночну функцію, яку можна було б попередньо побудувати , і застосовувати в ітераціях ASD.

Література

1. Ioannis Stamelos, Panagiotis Stetsos editors / Agile Software Development Quality Assurance / Information Science Reference ,USA, 2007, 257 p.

2. Mistrik I., Tang A., Bahsoon R. Software Architecture practices in Agile enterprise. Hershey: IGI Global, 2012, – pp. 230 – 249.

3. Babar A.M., Brown A., Mistrik I. Agile Software Architecture: aligning agile process and Software Architecture. Morgan Kaufman Elsevier inc. 2014, 410 p.

4. J. Coplien, G.Bjorving. Lean Architecture: for agile software development. // J Willey and Sons ltd.UK, 2010, 351p.

5. Kharchenko A. The method for comparative evaluation of software architecture with accounting of trade-offs // Alexandr Harchenko, Ihor Bodnarchuk, Vasyl Yatcyshyn //American Journal of Information Systems. 2(1) (2014) 20 – 25. Available online at http://pubs.sciepub.com/ajis/2/1/5.

6. Харченко О. Г. Проектування архітектури web-застосувань на основі моделі якості / О. Г. Харченко, І. О. Галай, І. О. Боднарчук, В. В. Яцишин // Інженерія програмного забезпечення. – 2010. – № 4. – сс. 26 – 34.

7. Kharchenko A. An Optimal Trade-off Solusion of the Software Architecture Choice Problem // Kharchenko A., Bodnarchuk I., Halay I., Yatcyshyn V. // Journal of Information and Computing Science Vol.11, No.4, 2016, – pp. 281 – 290.

8. Харченко О. Г. Метод багатокритеріальної оптимізації програмної архітектури на основі аналізу компромісів / Харченко О. Г., Боднарчук І. О., Галай І. О. // Інженерія програмного забезпечення. – 2012. – № 3–4 (11–12). – сс. 5 – 12.