

УДК 004.4

В.П. Бугай, І.О. Боднарчук, канд. техн.наук, доц.

Тернопільський національний технічний університет імені Івана Пулюя

ОЦІНЮВАННЯ ПРОГРАМНОЇ АРХІТЕКТУРИ ПРИ ГНУЧКИХ МЕТОДАХ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ

V. Buhai, O. Bodnarchuk, Ph.D., Assoc. Prof.

SOFTWARE SYSTEMS ARCHITECTURE ASSESSMENT FOR AGILE TECHNIQUE OF DEVELOPMENT

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

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, з яких слід відмітити такі:

- оскільки вимоги в гнучких методах є «полегшеними» і керованими локально в межах кожної ітерації, 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.

В традиційних методах проектування задача забезпечення якості ПС вирішується проведенням архітектурного проектування на початкових етапах розробки ПС. На етапі аналізу вимог до ПС формулюються вимоги до архітектури. Далі будуються варіанти архітектур. На останньому етапі проводиться оцінювання варіантів архітектур по множині критеріїв якості та вибір кращої з них, яка реалізується на наступних етапах проектування ПС. Одним з підходів до забезпечення якості ПС, розроблюваних в ASD, є поєднання гнучких та архітектурно орієнтованих методів проектування. З цього приводу є як позитивні оцінки, так і негативні [3], [4].

Для оцінювання такої можливості були проведені дослідження по застосуванню архітектурних технік в ASD командами розробників, які використовують SCRUM [2]. Результати показали, що команди розробників використовують в основному чотири принципи архітектурного проектування. Це big-up-front design, sprint zero, in sprints, separate architecture team. В практиці big-up-front design аналіз, синтез і оцінювання виконуються архітекторами до виконання спринтів, а не командою розробників. У практиці sprint zero проектування архітектури виконується у першому спринті командою розробників. Архітектура в подальшому реорганізується у межах спринтів. Оцінювання тут як правило не проводиться, тому що внесення необхідних змін в готовий продукт вимагає великих затрат. Застосування цієї практики можливе лише висококваліфікованою і досвідченою командою.

В separate architecture team підході існує окрема команда, яка розробляє архітектуру. В команду можуть входити представники різних команд розробників, в тому числі і архітектор. Архітектурна оцінка проводиться командою при випуску нової версії.

Як виходить з наведеного аналізу, елементи архітектурного проектування певним чином використовуються а ASD. Однак, для більш ефективного його застосування, з метою забезпечення якості, необхідно підвищити роль архітектора в ASD. Також потребують певної модифікації і архітектурні практики, для впровадження їх в ASD. Для цього в [4] пропонується проектувати архітектуру адаптованою до змін для полегшення внесення в неї цих змін у випадку уточнення вимог.

Поєднанню процесів архітектурного проектування з процесами гнучкої розробки присвячені також роботи [5], [6]. 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, причетних до розробки ПС, на запитання контрольного списку. Але внесення змін в архітектуру в ASD і оцінювання відповідності її якості вимогам відбуваються шляхом аналізу відповідей експертів на запитання "контрольного списку", що є досить суб'єктивним і не точним. Також в них відсутня математична формалізація процесів оптимізації вибору архітектури та оперативного коригування критеріїв якості зміненої архітектури.

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

Література

1. Ioannis Stamelos, Panagiotis Stetsos editors / Agile Software Development Quality Assurance./ Information Science Reference, USA – 2007, – 257 p.
2. Мое, N.B., Dingsøyr, T., and Dybå, T., A teamwork model for understanding an agile team: A case study of a Scrum project, Information and Software Technology 52 (2010).
3. Abrahamsson P., Babar A.M., Krutchen P. Agile and Architecture- can the co- exist? IEEE Software 2010, 27(2): p. 16–22.
4. Mistrik I., Tang A., Bahsoon R. Software Architecture practices in Agile enterprise. Hershey: IGI Global, 2012. P. 230–49.
5. Babar A.M., Brown A., Mistrik I. Agile Software Architecture: aligning agile process and Software Architecture. Morgan Kaufman Elsevier Inc. 2014: – 410 p.
6. J. Coplien, G. Bjarving. Lean Architecture: for agile software development. // J Willey and Sons ltd. U.K., 2010, – 351 p.