

Архітектура як опис підсистем цільової системи «програмний продукт»

Рач В. А.

*д. т. н, професор,
директор ННІ інформаційних та комунікаційних технологій,
ВНЗ «Університет економіки та права «КРОК», м. Київ, Україна,
e-mail: v_rach@krok.edu.ua,
ORCID: 0000-0002-6563-2354*

Гулько О. Р.

*студент кафедри інформаційних технологій,
ВНЗ «Університет економіки та права «КРОК», м. Київ, Україна,
e-mail: HunkoOR@krok.edu.ua*

Одним з етапів розробки будь-якого програмного продукту (ПП) є етап побудови його архітектури. В професійному ІТ середовищі поняття архітектури містить в собі дуже багато аспектів виробництва ПП, починаючи від визначення проблем у замовника, цілей автоматизації та закінчуючи етапом утилізації застарілого ПП, яке колись було розроблено в межах конкретного проєкту. Крім того, незважаючи на наявність стандартів (наприклад ISO/IEC/IEEE 42010 «Системи та розробка програмного забезпечення. Опис архітектури» [1]), книжок, які закликають «оновити свою архітектурну практику для нових викликів, середовищ і очікувань зацікавлених сторін» [2], кожна компанія має власне корпоративне бачення архітектури. Саме тому за даними, зібраними на веб-сайті Інституту програмної інженерії університету Карнегі-Меллон, вже нараховується більше 150-ти визначень цього терміну. Це дає нам підставу припустити, що однією з причин такого становища є відсутність методологічних основ організації діяльності, яка пов'язана з ІТ архітектурою. Логічним кроком у вирішенні цього питання може стати застосування сучасного системного підходу 2.0, його базової термінології, принципів та організації нового мислення [3].

З позиції методологічного рівня, доречно скористуватися наступним ємним визначенням архітектури: «архітектура-це про все важливе. Щоб це не було» [4]. А як визначити, що важливо і хто це повинен визначати? Згідно з системним мисленням, це стейкхолдери, які в проєкті виконують відповідні ролі. Для кожної ролі майбутньої ПП в системі діяльності замовника буде представлятись у своїх аспектах (системних поділах), кожний з яких є дисципліно-орієнтованим і має свою мову опису. Але при цьому при множині описів у системному мисленні ПП буде залишатися цілісною/холістичною/цілокупністю всіх системних поділів і розглядатись як холон, тобто систему яка одночасно є частиною і більшої надсистеми. Тоді таку цілісність треба розглядати як цільову систему, а ролі поділити на дві групи. Ролі першої групи відносяться до надсистеми і представляють інтереси систем в її оточенні, а ролі другої-інтереси підсистем, які складають у взаємосодії цільову систему.

Такий підхід не суперечить відомій схемі Захмана, яка на сьогодні є одним з повних архітектурних каркасів, що визначає загальні властивості інформаційних систем. Це рівень, коли властивості ще не залежать від парадигми проєктування, технологій і засобів розробки [5]. Її перевага полягає в тому, що вона систематизує знання про архітектуру системи і охоплює всі аспекти про все важливе, щоб це не було, яке визнали стейкхолдери. Наразі з позиції системного мислення рекомендовано виділяти не менше трьох груп опису: функціональний / ролевий / аналітичний; модульний / конструктивний /

синтетичний; просторовий.

Перевіримо запропоновану методологію бачення на прикладі методу Крухтен 4+1 перспектив [6]. Він передбачає чотири групи опису: логічний, функціональний, фізичний, розробки. Інтерфейсом між групами є сценарії. Логічний опис-це **що** система робить, який функціонал ПП надає для задоволення потреб клієнта. Функціональний опис-це **як** ПП робить, що відбувається всередині функціоналу, як саме система виконує його. Фізичний опис-це **де, коли, чим** ПП виконується, яке технічне обладнання, системне забезпечення. Опис розробки-це **ким та як** буде реалізовуватись ПП.

Розглянемо, які ролі до якого опису можна віднести. Логічний опис відноситься до першої групи стейкхолдерів, всі інші-до другої. Стейкхолдери першої групи виконують роль планувальників, які встановлюють межі, інтерфейси, основні функції ПП. До стейкхолдерів другої групи відносяться ролі проєктувальника, який визначає фізичні компоненти ПП, їх взаємодію, межі, інтерфейси, відноситься до фізичного опису; розробника-проєктувальника, який визначає ієрархічну структуру, технології, бібліотеки ПП, відноситься до опису розробки; менеджера проєкту, який визначає бюджет, терміни, ролі в команді розробників, відноситься до опису розробки; інтегратора, який визначає сценарії та потоки даних, графічний інтерфейс користувача, відноситься до функціонального опису; конструкторів, які поєднують описи між собою, створюючи загальну картину архітектури.

Перша група ще виконує роль моделювання для перевірки правильності й повноти розроблених описів вимогам, які були сформульовані на попередньому етапі. У разі позитивного результату перевірки друга група формує план щодо тестування етапу виготовлення (складання). В ньому визначають головні атрибути якості, для яких надалі будуть зроблені сценарії. Ця група також буде формувати команду розробників на основі розробленої архітектури.

Як бачимо, методу Крухтен 4+1 перспектив повністю відповідає запропонованій методології бачення, яка базується на сучасному системному мисленні 2.0.

Список використаних джерел

1. ISO/IEC/IEEE 42010. Systems and software engineering. Architecture description. URL: https://www.academia.edu/38568828/ISO_IEC_IEEE_42010_Systems_and_software_engineering_Architecture_description
2. Woods E., Erder M., Pureur P. (2022). *Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps*. Addison-Wesley Professional. 352 p.
3. Левенчук А. Системное мышление: Учебник . Бостон-Ульдинген-Киев. Проект «Баловство», Толман. 2019. 536 с.
4. Fowler M. (2003). *Who Needs an Architect?* IEEE Computer Society. URL: <https://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf>.
5. Мадорская Ю. М. *Схема Захмана при разработке требований к ИС. Практика проектирования систем*. 2015. URL: <http://reqcenter.pro/zachman-framework/>.
6. Software Architecture Course Notes. (2017). University of Alberta. URL: https://d3c33hcgivew3.cloudfront.net/_773d48108e2dda1c1a731bf69b06c3be_Software-Architecture_Course-Notes.pdf?Expires=1670025600&Signature=IEbjuAUrbdcMQs31d4sRb2~fLLajnAAmj4Ymw0cFni---e4mblhjxfYbi2n9m5CGcvdbhw5IViFve3wfcJ5BGDJzyRFqO3Z4ilUewrzwfEAlkzftkQIDeXv0T-MclPCSkmlfQ5BX6XWOB5ojXwBF7mGblvUSMhWz4VvCZHjtk8Y_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A