

Побудова архітектури програмного засобу для освітньої системи за допомогою методології Захмана

Я.О. Тупало

Інститут кібернетики імені В.М. Глушкова НАН України, 03187, м. Київ,
проспект Академіка Глушкова, 40, tupaloyaroslav91@gmail.com

Ya. Tupalo

BUILDING A SOFTWARE ARCHITECTURE FOR THE EDUCATION SYSTEM USING THE ZACHMAN METHODOLOGY

Abstract. *The main purpose of this article is to demonstrate a modern approach to constructing a complex architectural architecture using the Zachman method. The impetus for the writing of this article was the misunderstanding of many technical specialists of the holistic picture of the development of complex software architecture from the stage of translating the idea into human language or building the system on the server or computer. The Zachman methodology gives a mechanism to see all stages of software development and to present these steps in the form of a matrix of criteria. The article also discusses other methods of building the architecture of the software complex, the author tried to cluster these methodologies and to conduct a brief analysis of these methodologies. The author also provided a general example of using the Zachman methodology, this example can be used as a template to build your own system. Enterprise Architecture is a discipline which has evolved to structure the business and its alignment with the IT systems. The Zachman Framework is an enterprise ontology and is a fundamental structure for Enterprise Architecture which provides a way of viewing an enterprise and its information systems from different perspectives, and showing how the components of the enterprise are related. In today's complex business environments, many large organizations have great difficulty responding to changes. Part of this difficulty is due to a lack of internal understanding of the complex structure and components in different areas of the organization, where legacy information about the business is locked away in the minds of specific employees or business units, without being made explicit. The Zachman framework provides a means of classifying an organization's architecture. It is a proactive business tool, which can be used to model an organization's existing functions, elements and processes - and help manage business change. The framework draws on Zachman's experience of how change is managed in complex products such as airplanes and buildings. This article will be interesting not only for technicians but also for business analysts and executives of software development companies.*

Key words: *ST, IT, FEAF6 DoDAF.*

Анотація. *Основна мета цієї статті – продемонструвати сучасний підхід до побудови складної архітектурної архітектури методом Захмана. Поштовхом до написання цієї статті стало нерозуміння багатьма технічними фахівцями цілісної картини розвитку складної архітектури програмного забезпечення від етапу перекладу ідеї на людську мову до побудови системи на сервері чи комп'ютері. Методологія Захмана дає механізм бачити всі етапи розробки програмного забезпечення та представляти ці кроки у вигляді матриці критеріїв. У статті також обговорюються інші методи побудови архітектури програмного комплексу, автор намагався згрупувати ці методології та провести короткий аналіз цих методологій. Автор також подав загальний приклад використання методології Захмана, цей приклад може бути використаний як шаблон для побудови власної системи. Архітектура підприємства – це дисципліна, яка розвинулася для структури бізнесу та його узгодження з ІТ-системами. Підхід Захмана є онтологією підприємства і є фундаментальною структурою для архітектури, яка забезпечує спосіб перегляду підприємства та його інформаційних систем з різних точок зору, а також показує, як пов'язані компоненти підприємства. У сучасних складних бізнес-середовищах багато великих організацій мають великі труднощі з реагуванням на зміни. Частина цих труднощів пов'язана з відсутністю внутрішнього розуміння складної структури та компонентів у різних областях організації, де застаріла інформація про бізнес замикається у свідомості конкретних службовців або бізнес-підрозділів, не роблячи явного. Підхід Захмана забезпечує засіб класифікації архітектури організації. Це проактивний бізнес-інструмент, який можна використовувати для моделювання існуючих функцій, елементів і процесів організації та допомагати в управлінні змінами бізнесу. Підхід спирається на досвід Захмана щодо управління змінами у складних продуктах, таких як літаки та будівлі. Ця стаття буде цікавою не тільки для техніків, а й для бізнес-аналітиків та керівників компанії, що займаються розробкою програмного забезпечення.*

Ключові слова: *ST, IT, FEAF6 DoDAF.*

Анотация. *Цель этой статьи – продемонстрировать современный подход к построению сложной архитектурной архитектуры методом Захмана. Методология Захмана дает механизм видеть все этапы разработки*

© Я.О. ТУПАЛО, 2019

Вступ. Архітектура програмного забезпечення (ПЗ) [1] – це його будова як її видно (або має бути видно) зовні, тобто уявлення ПЗ як системи, що складається з деякої сукупності взаємодіючих підсистем. В якості таких підсистем виступають зазвичай окремі програми. Розробка архітектури є першим етапом боротьби зі складністю ПЗ, на якому реалізується принцип виділення відносно незалежних компонентів. Основні завдання розробки архітектури ПЗ: виділення програмних підсистем і відображення на них зовнішніх функцій (заданих в зовнішньому описі) ПЗ; визначення способів взаємодії між виділеними програмними підсистемами. З урахуванням прийнятих рішень проводиться подальша конкретизація і функціональна специфікація. Розрізняють такі основні класи архітектури програмних засобів [2]: цілісна програма; комплекс автономно виконуваних програм; багат шарова програмна система; комплекс паралельно виконуваних програм. Цілісна програма являє вироджений випадок архітектури ПЗ: до складу ПЗ входить тільки одна програма. Таку архітектуру вибирають зазвичай у тому випадку, коли ПЗ має виконувати одну яку-небудь яскраво виражену функцію і її реалізація не видається занадто складною. Природно, що така архітектура не потребує будь-якого опису (крім фіксації класу архітектури), так як відображення зовнішніх функцій на цю програму тривіальна, а визначити спосіб взаємодії не потрібно (в силу відсутності будь-якої зовнішньої взаємодії програми, крім як взаємодії її з користувачем, а останнє описується в документації щодо застосування ПЗ). Комплекс автономно виконуваних програм складається з набору програм, таких, що: будь-яка з цих програм може бути активізована (запущена) користувачем; при виконанні активізованої програми інші програми цього набору не можуть бути активізовані до тих пір, поки не закінчить виконання активізована програма; всі програми цього набору застосовуються до одного інформаційного середовища. Таким чином, програми цього набору не взаємодіють один з одним – взаємодія між ними здійснюється тільки через загальне інформаційне середовище. Багат шарова програмна сис-

тема складається з деякої впорядкованої сукупності програмних підсистем, званих шарами, такий, що: на кожному шарі нічого не відомо про властивості (і навіть існування) подальших (вищих) шарів; кожен шар може взаємодіяти (звертатися до компонентів) з безпосередньо попереднім (більш низьким) шаром через задалегідь визначений інтерфейс, нічого не знаючи про внутрішню будову всіх попередніх шарів; кожен шар може користуватися певними ресурсами, які він або приховує від інших верств, або надає безпосередньо подальшому шару (через вказаний інтерфейс) деякі абстракції. Таким чином, в багат шаровій програмній системі кожен шар може реалізувати деяку абстракцію даних. Зв'язки між шарами обмежені передачею значень параметрів звернення кожного шару до суміжного знизу шару і обміном результатів цього звернення від нижнього шару верхнього. Неприпустимо використання глобальних даних декількома шарами.

Загальна частина. Розробка ІТ-архітектури підприємства включає у себе компоненти, пов'язані з функціональною архітектурою, інформаційними технологіями (ІТ) і управлінням архітектурним процесом, які засновані на стратегії розвитку підприємства. Таким чином, ІТ-архітектура підприємства є цілісним описом ключових стратегій організації, пов'язаних з інформацією, прикладними системами і технологіями, а також їх впливом на функції і бізнес-процеси організації. Розробка ІТ-архітектури підприємства ведеться у відповідному контексті існуючих в організації структур управління і взаємодії.

Як же насправді відбувається проектування підприємства і які артефакти при цьому виникають? Перш ніж проектувати підприємство, будується модель вимог до нього. Модель вимог формується на основі вимог, які пред'явлені до цього підприємства з боку всіх його учасників, контрагентів і стейкхолдерів. Аналог в ІТ – вимоги до програмного продукту. Далі на основі цих вимог будується модель процесів підприємства з необхідним ступенем деталізації. Аналогом в ІТ буде перелік функцій програмного продукту. Далі будується модель функціональних об'єктів, або, кажучи спеціалізованою мовою, технічних місць, які повинні брати участь у перерахованих раніше процесах. Аналогом в ІТ буде опис процедур, і пояснення

які процедури в яких функціях беруть участь. Далі підбираються ті одиниці обладнання, які можуть виконувати ролі перерахованих технічних місць. Аналог в ІТ – це програмний код.

Підприємство – це функціональний об'єкт, який створений для задоволення певним вимогам. У цьому сенсі підприємство нічим не відрізняється від такого об'єкта, як годинник, або виробнича лінія. Часто замість терміна функціональний об'єкт можна почути термін технічне місце. Технічне місце відрізняється від одиниці обладнання тим, що одиниця устаткування виконує роль технічного місця. Наприклад, трансформатор виконує роль перетворювача напруги, при цьому в різний час різні трансформатори можуть виконувати роль одного і того ж перетворювача. Ще одним прикладом технічного місця є посада, відділ, підрозділ, штат. Наприклад, токарь бере участь у функції виготовлення деталей. Це технічне місце, роль якого в різний час можуть виконувати різні одиниці обладнання (фізичні особи). При моделюванні технічних місць, ми описуємо процеси і учасників цих процесів. Зауважу, що саме учасників, а не виконавців, – трансформатор не може перетворювати напругу, тому що він не є живою істотою. Якщо все ж сказати, що трансформатор «перетворює» напругу, то це – метонімія, яка розкривається так: трансформатор, виконує роль перетворювача напруги, який (перетворювач) бере участь у процесі перетворення напруги. Про метонімію можна прочитати в книзі «Метафори, якими ми живемо», автори: Джордж Лакофф, Марк Джонсон [3]. Іншою поширеною метонімією буде вислів: «комп'ютер вирішує завдання». Ті ж, хто дійсно вважають, що трансформатор, або комп'ютер щось робить насправді, одухотворюють неживе, користуючись міфічною свідомістю.

Зауважимо, що до цього моменту ні слова не сказано про цілі, про виконавців і причинно-наслідкові зв'язки. Згадується лише про вимоги, про функції і учасників цих функцій – технічних місць. Цілі залишилися на етапі формування вимог до підприємства і далі вони не згадуються. В деяких випадках відомі дані цілі, а в деяких ні, для моделі підприємства це не має ніякого значення. Модель підприємства відповідає на питання: як ми задовольняємо вимоги, а не те, звідки взяли ці вимоги. Виконавців теж немає, тому що не треба користуватися теорією діяльності, щоб описати

учасників активності. Ми не будемо причинно-наслідкові зв'язки. Якщо ж треба побудувати модель причинно-наслідкових зв'язків, то це ще одна додаткова модель. Це знання, якими користуються технологи при проектуванні підприємства. Це галузеві знання, і вчать їх в інститутах дуже багато років. Змоделювати, чому летить літак – нереально важко, і ніхто цього не робить. Просто моделюють політ літака.

Існують різні моделі і методики опису ІТ-архітектури. Ці методики задають класифікацію основних областей і єдині принципи для їх опису у взаємній ув'язці один з одним, опис використовуваних політик, стандартів, процесів, моделей для визначення різних елементів ІТ-архітектури:

- методики аналітичних компаній: Gartner, Giga, META і ін.;
- модель Захмана;
- методика TOGAF;
- методика POSIX 1003.23і й ін.

Для державних організацій існують спеціальні методики, такі як розроблені за підтримкою уряду США Федеральна Архітектура держорганізаціям (FEAF) або використовувана в Міністерстві Оборони США DoDAF (Department of Defence Architecture Framework).

Методика – це інструмент для створення широкого спектра різних архітектур. Вона, як правило, включає у себе опис методів проектування ІТ-архітектури в термінах використання певних "будівельних блоків", опис того, як ці "будівельні блоки" пов'язані між собою, набір інструментів для опису елементів архітектури, загальний словник використовуваних термінів. Методики також можуть містити список рекомендованих стандартів і сумісних продуктів, які можуть використовуватися для реалізації різних елементів архітектури. Важливо розуміти, що методики не тільки задають набір документів і планів, необхідних для опису підприємства, а й визначають, як всі ці елементи опису пов'язані між собою. Існують індустриальні стандарти для опису ІТ-архітектури підприємства, прийняті такими організаціями, як IEEE, ISO, описані в ITIL, COBIT і т. д. Але жоден з цих стандартів не займає домінуючого положення. Більш того, жоден з них, взятий окремо, не дає групам розробників ІТ-архітектури, всіх необхідних інструментів з методичної точки зору і з точки зору шаблонів, які використовуються для опису архітектури.

Однак цей накопичений арсенал методик і стандартів надає архітекторам широкі можливості вибору моделей, прикладів і досвіду різних індустрій.

Значний внесок у розвиток концепції архітектури підприємства був зроблений Дж. Захманом. З 1987 році, коли була запропонована перша версія цієї моделі, розширена згодом в роботах 1992–1996 рр. Вона була використана досить великою кількістю великих компаній, що входять в список 2000 найбільших корпорацій світу. Модель Захмана послужила основою для створення цілого ряду інших методик і моделей опису архітектури підприємства, таких як Федеральна архітектура США (FEAF), методика опису архітектури Open Group (TOGAF), методика опису архітектури міністерства оборони США (DoDAF).

Модель Захмана заснована на дисципліні класичної архітектури і забезпечує загальний словник і набір перспектив або структур, для опису сучасних складних корпоративних систем. У своїй роботі Дж. Захман визначив архітектуру підприємства як "набір описових моделей, які застосовуються для опису підприємства відповідно до вимог управлінського персоналу і які можуть розвиватися протягом певного періоду". Термін "архітектура" тут не випадкова, вона підкреслює існуючу аналогію між внутрішньою структурою абстрактного об'єкта – підприємства, і складного штучного об'єкта, такого як будівля або космічна станція.

Історично модель Захмана вперше була створена саме для ІТ-систем. Цей підхід в подальшій роботі був узагальнений для розгляду не тільки ІТ-систем, але і для опису підприємства в цілому, так що запропонована модель, взагалі кажучи, може використовуватися як засіб для опису архітектур складних виробничих систем будь-якого типу.

Метод Захмана є однією з ранніх спроб пов'язати характеристики інформаційної системи з бізнес-завданнями підприємства. Жодна сучасна організація не працює без системи або систем будь-якого роду, за допомогою яких досягаються цілі функціонування цієї організації. Інформаційна система – це комбінація «ручних» і комп'ютерних процесів, які вирішують поставлені завдання, чітко і логічно взаємодіючих між собою. В умовах сучасної конкурентної економіки використання розви-

нутих інформаційних систем допомагає організаціям займати лідируючі позиції у бізнесі.

Основна ідея полягає у тому, щоб забезпечити можливість послідовного опису кожного окремого аспекту системи в координації з усіма іншими. Для будь-якої досить складної системи загальне число зв'язків, умов і правил зазвичай перевершує можливості для одночасного розгляду. Водночас окреме, у відриві від інших, розгляд кожного аспекту системи найчастіше призводить до неоптимальних рішень як у плані продуктивності, так і вартості реалізації. Цей метод добре відомий у світовій практиці. Суть його зводиться до формалізованого представлення моделі підприємства у вигляді матриці. У рядках цієї матриці відображаються різні категорії фахівців, певним чином пов'язаних з діяльністю підприємства (планувальник, власник підприємства, проектувальник, розробник і субпідрядник), а в стовпчиках – основні аспекти виробничої діяльності (об'єкти = що, дії = як, розташування = де, люди = хто, час = коли і мотиви = чому). Власне модель представляється у вигляді таблиці, що має п'ять рядків і шість стовпців, які приведені в табл. 1.

ТАБЛИЦЯ 1. Матриця Захмана

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34
35	36	37	38	39	40	41

- 1 – дані ЩО?
- 2 – функції ЯК?
- 3 – дислокація ДЕ?
- 4 – люди ХТО?
- 5 – час КОЛИ?
- 6 – мотивація ЧОМУ?
- 7 – планувальник;
- 8 – список важливих понять і об'єктів;
- 9 – список основних бізнес-процесів;
- 10 – територіальне розташування;
- 11 – ключові організації;
- 12 – найважливіші події;

- 13 – бізнес-цілі і стратегії;
- 14 – власник, менеджер;
- 15 – концептуальна модель даних;
- 16 – модель бізнес процесів;
- 17 – схема логістики;
- 18 – модель потоку робіт;
- 19 – майстер-план реалізації;
- 20 – бізнес-план;
- 21 – конструктор, архітектор;
- 22 – логічні моделі даних;
- 23 – архітектура додатку;
- 24 – модель розділеної архітектури;
- 25 – архітектура інтерфейсу користувача;
- 26 – структури керування;
- 27 – ролі і моделі бізнес-правил;
- 28 – проєктант;
- 29 – фізична модель даних;
- 30 – системний проєкт;
- 31 – технологічна архітектура;
- 32 – архітектура презентації;
- 33 – структури управління;
- 34 – опис бізнес-правил;
- 35 – розробник;
- 36 – опис структури даних;
- 37 – програмний код;
- 38 – мережева модель;
- 39 – архітектура безпеки;
- 40 – визначення;
- 41 – реалізація бізнес логіки.

Два верхні рядки відповідають найбільш загальним уявленням і досить широко описують існуюче оточення, плани і цілі. Якщо проводити аналогію з будівництвом, то ці рівні містять відомості про місцезнаходження і призначення будівлі, а також плани і зображення, які архітектор обговорює з господарем майбутнього будинку. Наступний рівень "логічної моделі" вже є більш конкретним, але все одно ще досить абстрактним. Це схеми, які архітектор будинку повинен показувати підрядникам. Аналогічно, в застосуванні до діяльності підприємства верхній рядок "Контекст" відповідає рівню інтересів вищого керівництва і зборів акціонерів. Другий рівень відповідає інтересам менеджерів і власників процесів. Третій рівень – той, на якому менеджери, аналітики та ІТ-менеджери повинні працювати разом. Рівні з четвертого і далі описують деталі, які становлять інтерес для ІТ-менеджерів, проєктувальників, розробників. На кожному з цих рівнів учасники розглядають одні й ті ж категорії питань, відповідних

стовпців у табл. 1, але з різним рівнем абстракції і деталізації. У зміст цих колонок входять:

- задіяні дані (що?);
- процеси і функції (як?);
- місця виконання цих процесів (де?);
- організації та персоналії-учасники (хто?);
- керуючі події (коли?);
- цілі і обмеження, що визначають роботу системи (навіщо?).

Основні правила заповнення таблиці наступні:

- кожна клітина таблиці незалежна від інших, разом вони утворюють функціонально повний простір для опису системи ("базис");
- порядок проходження колонок є несуттєвим;
- кожна клітка містить відповідний опис аспекта реалізації системи у вигляді певної моделі або простого опису;
- базові моделі для кожної з колонок – унікальні;
- відповідні моделі в клітинах кожного ряду в сукупності утворюють повний опис системи з обраної перспективи;
- заповнення клітин має проводитися послідовно "зверху вниз".

Перший рядок відповідає рівню планування бізнесу в цілому (бізнес-модель). На цьому рівні вводяться досить загальні основні поняття, що визначають бізнес (продукти, послуги, клієнти), а також формулюється бізнес-стратегія. Фактично, даний рядок визначає контекст усіх наступних рядків. Другий рядок (концептуальна модель) призначена для визначення в термінах бізнесу структури організації, ключових і допоміжних бізнес-процесів. Третій рівень (логічна модель) відповідає розгляду з точки зору системного архітектора. Тут бізнес-процеси описуються вже в термінах інформаційних систем, включаючи різні типи даних, правила їх перетворення і обробки для виконання їх на рівні бізнес-функцій. На четвертому рівні технологічної або фізичної моделі здійснюється прив'язка даних і операцій над ними до обраних технологій реалізації. Наприклад, тут може бути визначений вибір реляційної СУБД, або засоби роботи з неструктурованими даними, або об'єктно-орієнтованого середовища. П'ятий рівень відповідає детальній реалізації системи, включаючи конкретні моделі обладнання, топологію мережі, виробника і версію СУБД, засоби

розробки і власне готовий програмний код. Багато з робіт на даному рівні часто виконуються субпідрядниками. Шостий рівень описує працюючу систему. На цьому рівні можуть бути введені такі об'єкти, як інструкції для роботи з системою, фактичні бази даних, робота служби HelpDesk і т. д. Треба зауважити, що у вихідній роботі Захмана зміст цього рівня не деталізується.

Розглянемо, як здійснюється послідовна деталізація окремих аспектів опису системи, для чого звернемо увагу на різні колонки таблиці. Так, перша колонка відповідає на питання "ЩО?" і визначає використовувані в системі дані. На верхньому рівні достатнім буде просте перерахування основних об'єктів, використовуваних у бізнесі. На другому рівні дані об'єкти об'єднуються у семантичну модель високого рівня і зазвичай описуються у вигляді діаграми "сутність-зв'язок" з відображенням основних зв'язків і найбільш істотних обмежень. На третьому рівні ця модель приводиться до нормалізованої форми, визначаються всі атрибути і ключі. Четвертий рівень – це фізична модель даних у системі. Наступний рівень містить опис моделі на мові управління даними для формування таблиць. Останній рівень може описувати фактичні набори даних, в тому числі такі характеристики, як журнали доступу, розміри реально займаного дискового простору, статистику звернень і т. п.

Колонка функцій (відповідь на питання "ЯК?") призначена для послідовної деталізації опису того, як місія підприємства реалізується на рівні окремих операцій. Зокрема, на першому рівні достатнім буде просте перерахування процесів. Другий рівень буде містити модель процесів, яка згодом деталізується в операції над даними та архітектуру додатків (рівень 3), методи класів (рівень 4), програмний код (рівень 5) і, нарешті, виконувані модулі. При цьому, починаючи з 4-го рівня, розгляд ведеться вже не в рамках підприємства в цілому, а за окремими підсистемами або додатками.

Наступна колонка (питання "ДЕ?") визначає просторовий розподіл компонентів системи і мережеву організацію. На рівні планування бізнесу тут досить визначити розташування всіх виробничих об'єктів. На наступному рівні ці об'єкти об'єднуються в модель із зв'язками, що характеризують взаємодію між собою. На

третьому рівні системної архітектури здійснюється прив'язка компонент інформаційної системи до вузлів мережі. Четвертий рівень служить для визначення фізичної реалізації у термінах апаратних платформ, системного програмного забезпечення, а також засобів проміжного рівня, використовуваних для інтеграції різних компонент інформаційної системи між собою. На п'ятому рівні визначаються використовувані протоколи і специфікації каналів зв'язку. Останній рівень описує функціонування реалізованої мережі.

Колонка таблиці, що відповідає на питання "ХТО?", визначає учасників процесу. На рівні планування бізнесу тут представлений список підрозділів підприємства і виконувані ними функції. На наступному рівні наводиться повна організаційна діаграма, а також повинні бути визначені загальні вимоги до інформаційної безпеки. Далі послідовно визначаються учасники бізнес-процесів і їх ролі, вимоги до інтерфейсів користувача і правила доступу до окремих об'єктів, фізична їх реалізація на рівні коду або операторів визначення доступу до таблиць в СУБД. Останній рівень описує навчених користувачів системи.

Остання колонка ("ЧОМУ?" або "НА-ВЩО?") служить для визначення мотивації і задає порядок переходу від завдань бізнесу до вимог і елементів інформаційних систем. Вихідною точкою є бізнес-стратегія, яка потім послідовно транслюється в бізнес-план, потім у правила і обмеження для реалізації бізнес-процесів, а на рівні 4 – відповідними програмами, необхідних для включення до складу інформаційних систем і, в подальшому, в їх фізичну реалізацію. Така модель опису корисна для ідентифікації можливих обмежень. Ці обмеження можуть поширюватися як від верхніх рівнів до нижніх (наприклад, вказівка керівництва компанії про вибір тих чи інших засобів, продуктів або принципів роботи), так і в зворотньому напрямку (наприклад, можливості існуючих технологій бездротового зв'язку в значній мірі визначають спектр пропонованих послуг і організацію бізнес-процесів у провайдерів цих послуг).

Тепер розглянемо застосування моделі Захмана для побудови програмного комплексу для освітньої системи. У табл. 2 опишемо перший шар.

ТАБЛИЦЯ 2. Перший шар

Власник компанії	
Дані ЩО?	Готовий продукт – інформаційна система
Функції ЯК?	Пошук замовника; обговорення проекту на концептуальному рівні; підписання договорів з замовником; призначення керівника на проект; продаж, впровадження і підтримка інформаційної системи
Дислокація ДЕ?	Відділ продажу; відділ аналізу; відділ розробки; відділ тестування; відділ впровадження і підтримки
Люди ХТО?	Зовнішній замовник, користувач системи; керівник проекту; аналітик; розробник; тестировщик; супорт
Час КОЛИ?	Часовий період вказаний в контракті
Мотивація ЧОМУ?	Отримання прибутку від продажу системи; отримання стабільного прибутку від підтримки системи; продаж готової інформаційної системи стороннім замовникам

ТАБЛИЦЯ 3. Другий шар

Керівник проекту	
Дані ЩО?	Основні об'єкти і сутності інформаційної системи – концептуальна модель системи
Функції ЯК?	Взаємодія з замовником; постановка задачі аналітикам проекту
Дислокація ДЕ?	Окреме робоче місце керівника проекту
Люди ХТО?	Замовник; аналітики проекту; відділ розробки (керівник відділу розробки)
Час КОЛИ?	Календарний план впровадження робіт, створений на основі ТЗ і контракту з замовником
Мотивація ЧОМУ?	Збільшення прибутку за рахунок вдалої задачі проекту

ТАБЛИЦЯ 4. Третій шар

Бізнес і системні аналітики проекту	
Дані ЩО?	Поглиблена модель інформаційної системи; інформаційна система, розбита на окремі моделі (блоки); потреби замовника; вимоги до системи
Функції ЯК?	Виявлення вимог замовника; визначення і керування вимог до системи; оформлення бізнес і системних постановок; постановка задачі керівнику відділу розробки; нотатки поправок в бізнес логіку системи на основі зауважень замовника і розробника
Дислокація ДЕ?	Відділ аналітики
Люди ХТО?	Замовник; керівник проекту; бізнес аналітики проекту; системні аналітики проекту
Час КОЛИ?	Часові проміжки, які виділені на розробку окремих модулів (блоків) системи
Мотивація ЧОМУ?	Ефективне керування вимог системи (мінімальні поправки); максимально точний і простий опис системи (за рахунок системного підходу)

ТАБЛИЦЯ 5. Четвертий шар

Керівник відділу розробки	
1	2
Дані ЩО?	Фізичне представлення моделі даних інформаційної системи
Функції ЯК?	Розробка фізичної моделі даних на основі системних постановок; управління відділом розробки (постановка задач розробникам, ефективний розподіл навантажень між розробниками); дотримання термінів виконуваних робіт
Дислокація ДЕ?	Окреме робоче місце керівника розробки
Люди ХТО?	Системні аналітики проекту; керівник відділу розробки

1	2
Час КОЛИ?	Часові проміжки виконуваних робіт які поставлені аналітиком
Мотивація ЧОМУ?	Ефективне керування відділом розробки; максимально просте і прозоре представлення системи, яке легко змінюється

ТАБЛИЦЯ 6. П'ятий шар

Розробники	
Дані ЦОУ?	Готові таблиці БД; програмні засоби для розробки системи
Функції ЯК?	Виконання завдань, які поставлені керівником; виправлення помилок, які були виявлені на етапі тестування
Дислокація ДЕ?	Відділ розробки
Люди ХТО?	Керівник розробки; системні аналітики проекту; розробники
Час КОЛИ?	Часові проміжки виконання завдань, які були поставлені керівником
Мотивація ЧОМУ?	Виконання задач вчасно; виконання завдань з мінімальним набором помилок

Основними характеристиками даної моделі Захмана є:

- простота для розуміння як технічними, так і нетехнічними фахівцями;
- цілісність стосовно підприємства, тобто кожна проблема може бути співвіднесена з підприємством в цілому;
- підтримка обговорень складних питань з використанням щодо невеликої кількості нетехнічних понять;
- можливість застосування для планування, що дозволяє краще приймати рішення за рахунок того, що рішення ніколи не буде виноситися "в порожнечі" (у відриві від інших аспектів діяльності підприємства);
- застосовність для вирішення завдань, тобто можливість працювати з абстракціями і сутностями, виділяючи і ізолюючи окремі параметри системи без втрати сприйняття підприємства як цілого;

• нейтральність, тобто незалежність від будь-яких інструментів; завдяки цьому кожен інструмент і методологія можуть бути відображені на дану модель і можуть явно показати, що вони роблять і чого вони не роблять.

Розглянемо, як може використовуватися такий підхід на практиці. По-перше, цю модель зручно застосовувати для класифікації всієї інформації, яка описує підприємство та інформаційні системи цього підприємства, виявлення "білих плям" і координації робіт. По-друге, цю модель можна використовувати на метарівні, для порівняння різних реалізацій створення архітектур підприємства. Нарешті, вона може бути зручним засобом для використання в окремих проектах. Не можна, звичайно, вважати, що дана модель позбавлена недоліків. Один з них полягає у тому, що при застосуванні її на практиці виникають певні труднощі, пов'язані з відсутністю "вбудованого механізму" поширення змін між елементами таблиці. Іншим обмеженням моделі є відсутність розгляду системи в динаміці. З побудованої матриці можна помітити, що бачення системи з точки зору розробників, не зовсім відноситься до самої системи. Кожен розробник зацікавлений лише у виконанні конкретного завдання, поставленого йому. Вони не зацікавлені у виконанні завдання таким чином, щоб вона максимально безболісно інтегрувалася з іншими завданнями, виконуваними на проекті, оскільки розробник не закріплений за конкретним проектом. У зв'язку з цим, на технічному рівні немає цілісного бачення системи (кінцевий результат), як між собою взаємодіють окремі процеси, звідки потім і виникають нестиківки, які потребують великих доопрацювань. У такій ситуації на технологічному рівні керівник відділу розробки – єдина і при цьому відповідальна людина щодо всіх проектів компанії. Виходить, що на цьому рівні архітектури системи лише начальник відділу зацікавлений в ефективній роботі системи, а його підлеглі (розробники), як виявляється, просто відпо- відають за її реалізацію. Слідуючи таким шляхом, важко добитися успіхів у проекті. Це може привести до підривання довіри з боку Замовника. Щоб уникнути такої ситуації, автором пропонується наступна модель проектної діяльності (модель «Тобе»). Пропонується під кожен окремий проект формувати групу. До складу проектної групи входять: керівник

проекту, бізнес і системні аналітики, розробники і тестувальники. Таким чином, за кожним проектом буде закріплено коло осіб, яке бере участь від початку і до кінця проекту. У всіх учасників групи буде складатися цілісне розуміння системи (буде видно кінцевий результат), і кожен буде прагнути до якісного виконання роботи, тому що за виконану роботу буде нестися відпові- дальність.

Це нагадує модель «Asis» (в частині реалізації проекту), але як уже було сказано вище, в ній відділ розробки був розформований на окремі групи. Також у кожній проектній групі можна призначити керівника розробки, важливо лише те, що ця людина у кожній групі повина бути своя.

Важливо зауважити, що сам Захман не запропонував конкретного набору моделей, які мають бути використані для опису кожного зрізу, і, тим більше, того як вони мають бути пов'язані між собою. Та це й неможливо, тому що у кожній компанії своя термінологія для опису своєї діяльності. Вибір конкретних методів залежить, у тому числі, від наявності фахівців-аналітиків тієї чи іншої кваліфікації.

Створена за вищенаведеними правилами модель архітектури конкретної організації покликана служити простим, але потужним інструментом підтримки системності планування і здійснення робіт з створення і розвитку корпоративної інформаційної системи і стикування її додатків. Архітектура дозволяє концентруватися на окремих аспектах системи і у водночас не втрачати відчуття загального контексту або

цілісної (холістичної) перспективи. Саме втрата такої перспективи, зокрема, розробка інформаційної системи субпідрядниками, які перебувають «поза контекстом», становить основну причину появи не інтегрованої системи.

Висновки. В статті наведено опис проблематики та шляхи вирішення даної проблеми завдяки методології Захмана. Дослідження, яке було проведено, показує, що будь яку комплексну систему можливо представити у вигляді матриці критеріїв, при чому кожен рівень даної матриці вирішує певну проблему побудови комплексної програмної архітектури. Дане дослідження представляє механізм переведення комплексної архітектури в простий і зрозумілий для всієї вертикалі компанії, яка займається розробкою програмного забезпечення.

СПИСОК ЛІТЕРАТУРИ

1. Майерс Г. Надійність програмного засобу. М.: Дружба, 1980. С. 78–91.
2. Дійкстра Є.В. Структура багатопрограмного програмування. *Communications of the ACM*. 1968. 11(5). С. 341–346.
3. Лакофф Д. Метафори, якими ми живемо. 2004. 256 с.

REFERENCES

1. G. Myers. Reliability of the software - M. : Druzhba 1980. P. 78–91
2. E.W. Dijkstra. The Structure of the THE Multiprogramming. *Communications of the ACM*. 1968, 11(5). P. 341–346.
3. D. Lakoff. The metaphors we live by. 2004. 256 p.

Одержано 03.09.2019