АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ И ПРОИЗВОДСТВАМИ

УДК 004.94:681.51 Научная статья
DOI: 10.35330/1991-6639-2024-26-3-42-54
EDN: JHNTJW

Эффективность подхода «Архитектура как код» в управлении ИТ-архитектурой предприятия

М. А. Шмонин

Государственный университет управления
109542, Россия, Москва, Рязанский проспект, 99

Аннотация. Потребность в архитектуре предприятия увеличивается по мере масштабирования бизнеса, а при текущем интересе к экосистемам является одним из наиболее востребованных вопросов для изучения. В статье осуществлен сравнительно-сопоставительный анализ существующих подходов в управлении архитектурой информационных технологий (ИТ-архитектура) предприятия. В результате исследования был синтезирован ряд проблем, характерных для рассматриваемых подходов, одной из которых является изолированность ИТ-архитектуры от корпоративной. В рамках цели по митигированию обнаруженных рисков автор предлагает использовать инновационный подход «Architecture as a Code», который лишен выявленных уязвимостей. В статье особое внимание уделяется исследованию возможностей данного принципа и тому, каким образом он делает процесс управления ИТ-архитектурой эффективнее. Автором был разработан набор критериев, однозначно определяющих подход «Архитектура как код». Для этого также было проведено исследование концепции «Diagram as a Code», показавшее фундаментальную разницу понятий. Детально были изучены особенности виртуализации и повторного использования программного кода, являющиеся основополагающими принципами подхода. На основе принципов и критериев автором были сформулированы и классифицированы требования для успешной интеграции нового процесса в организационную структуру предприятия.

Ключевые слова: проектирование архитектуры, «Architecture as a Code», системная аналитика, рефакторинг архитектуры, TOGAF, Archimate

Поступила 04.04.2024, одобрена после рецензирования 02.05.2024, принята к публикации 17.05.2024

Для цитирования. Шмонин М. А. Эффективность подхода «Архитектура как код» в управлении ИТ-архитектурой предприятия // Известия Кабардино-Балкарского научного центра РАН. 2024. Т. 26. № 3. С. 42–54. DOI: 10.35330/1991-6639-2024-26-3-42-54

MSC: 68N30 Original article

Effectiveness of the “Architecture as code” approach in management of IT-architecture of an enterprise

M.A. Shmonin

State University of Management,
109542, Russia, Moscow, Ryazansky prospect, 99

Abstract. The need for enterprise architecture increases as business scales, and with the current interest in ecosystems, it is one of the most in-demand issues for study. The article provides a comparative analysis of existing approaches to management of IT-architecture of an enterprise. As a result of the study, several

problems specific for the approaches under consideration were synthesized, one of which is the isolation of the IT-architecture from the corporate one. As part of the goal of mitigating the identified risks, the author proposes to use an innovative “Architecture as a Code” approach, which is devoid of identified vulnerabilities. The article pays special attention to exploring the capabilities of this principle, and how it makes the process of managing IT architecture more efficient. The author has developed a set of criteria that clearly define the “Architecture as a Code” approach. For this purpose, a study of the “Diagram as a Code” concept was conducted, which showed the fundamental difference between the concepts. The features of virtualization and reuse of program code, which are the fundamental principles of the approach, were studied in detail. Based on the principles and criteria, the author formulated and classified the requirements for the successful integration of a new process into the organizational structure of the enterprise.

Keywords: architecture design, “Architecture as a Code”, system analytics, architecture refactoring, TOGAF, Archimate

Submitted 04.04.2024, approved after reviewing 02.05.2024, accepted for publication 17.05.2024

For citation. Shmonin M.A. Effectiveness of the “Architecture as code” approach in management of IT-architecture of an enterprise. *News of the Kabardino-Balkarian Scientific Center of RAS.* 2024. Vol. 26. No. 3. Pp. 42–54. DOI: 10.35330/1991-6639-2024-26-3-42-54

Введение

Технологический прогресс в сфере ИТ оказывает сильное влияние на бизнес. Для сохранения конкурентоспособности современным предприятиям приходится постоянно адаптироваться и изменяться. Преобразования затрагивают как бизнес-преобразования компании, так и технологическое совершенствование. Обеспечение высокой скорости этих преобразований требует прозрачности структуры предприятия на всех уровнях. «Прозрачность» является одним из постулатов концепции цифрового двойника предприятия, которая предполагает интеграцию всех участников производственного процесса в единый непрерывный конвейер. В рамках данной концепции системные аналитики и архитекторы получают инструменты генерации интуитивно понятных и обоснованных требований, что кратно снижает длительность процесса интерпретации.

Однако существующие подходы к управлению и проектированию архитектуры информационных технологий (далее – ИТ-архитектуры) предприятия обладают рядом проблем, ресурсозатратность на решение которых пропорциональна сложности самого предприятия. Одной из таких проблем является эффект «черного ящика» при управлении архитектурой, в рамках которого архитектор может повлиять на решение только после его непосредственного внедрения.

Основной целью данного исследования является разработка рекомендаций по внедрению подхода «Architecture as a Code» для решения проблем, с которыми сталкиваются архитекторы в современных фреймворках управления и проектирования ИТ-архитектуры предприятия. Задачами исследования являются:

1. Обосновать различия между понятиями корпоративной архитектуры и ИТ-архитектуры.
2. Выявить сильные и слабые стороны подмножества архитектурных подходов и точек зрения, используемые с целью описания корпоративной архитектуры в современном мире.
3. Проанализировать существующие проблемы проектирования архитектуры и доказать их отрицательное влияние на эффективность процесса.
4. Исследовать новые методы управления архитектурой и проверить возможности решения с их помощью существующих проблем.

Методология

Тема корпоративной архитектуры настолько же древняя, как и тема ИТ. Первое упоминание о корпоративном архитекторе было зафиксировано в 1951 г. в журнале Ассоциации информационных систем в статье «The Legacy of LEO: Lessons learned from an English Tea and Cakes Company: Pioneering Efforts in Information Systems», где термин «Maestro of Technology» применяли к специалисту с хорошим пониманием бизнеса, разбирающемуся в информационных технологиях и умеющему найти общий язык со всеми стейкхолдерами. В 1962 г. в журнале «Harvard Business Review» выходит статья «Master plan for Information Systems», в которой процесс проектирования архитектуры делится на пять последовательных этапов. Первый – заложить долгосрочные цели и разработать базовую функциональность системы, второй – проанализировать и определить используемые информационные системы, третий – в кратчайшие сроки исправить несоответствия, четвертый – распределить ответственности в долгосрочных целях и пятый – реализовать план. Каждый из перечисленных пунктов нашел отражение в постулатах проектирования корпоративной архитектуры практически всех современных фреймворков [1].

Развивая тему фреймворков, стоит заметить, что на данный момент рынок перенасыщен различными подходами. В сети интернет можно найти такие названия, как TOGAF, FEAF, DoDAF, IAF, Zachman и многие другие. Такое изобилие обусловлено тем, что каждая крупная консалтинговая компания в свое время выпускала свой собственный фреймворк проектирования, который впоследствии пыталась монетизировать. Однако несмотря на их количество, противоречивость и отличия, цель каждого из них – связать бизнес-модель предприятия с его ИТ-архитектурой [2]. Между терминами «ИТ-архитектура» и «корпоративная архитектура» есть тонкая грань, которую компании, публикующие новые версии стандартов, активно пытаются размыть. Более того, совершается это умышленно с целью охвата большой аудитории и увеличения числа продаж. Показательным примером выступает история популярности фреймворка Zachman. Сам по себе Джон Захман был бизнес-консультантом в компании IBM, общаясь с руководством консультируемых компаний, он прибегал к формулировке «корпоративная архитектура», позиционируя свой фреймворк как инструмент не только для информационных систем, но и для всего предприятия в целом. Но стоит смоделировать и обратную ситуацию, при которой, например, директор запрашивает архитектуру своего предприятия. При этом стопка бумажных артефактов, описывающих устройство систем, не даст директору никакого понимания того, как его предприятие работает. Корпоративная архитектура – это фундаментальная организация системы или того, как компоненты самого высокого уровня соединяются друг с другом. Однако в такой неоднозначной формулировке главная сложность – разграничить, что есть «фундаментальное» и что есть «высокий уровень» [3]. Поэтому в течение времени профессиональное сообщество конкретизировало определение. Корпоративная архитектура – это описание элементов организации, их предназначения, роли и подхода к выполнению функций, отвечающее целям бизнеса и однозначно декодируемое для всех заинтересованных сторон. Объективно корпоративная архитектура призвана воплотить идею применения системного подхода в управлении организацией [4]. Стоит также понимать, что корпоративная архитектура реализуется через методы системного анализа и системной инженерии, поэтому она тесно связана с ИТ-архитектурой.

Мартин Фаулер, известный программист, автор ряда статей по архитектуре ПО, рефакторингу и предметно-ориентированным языкам программирования, в своей книге «Кто нуждается в архитекторе?» пишет, что в результате общения с Ральфом Джонсоном он пришел к следующему: «В наиболее успешных проектах программного обеспечения

эксперты-разработчики, работающие над этим проектом, имеют общее понимание конструкции системы. Это общее понимание называется «архитектурой». Также оно включает в себя то, как система делится на компоненты и как компоненты взаимодействуют через интерфейсы. Эти компоненты обычно состоят из более мелких компонентов, но архитектура включает только те компоненты и интерфейсы, которые понятны всем разработчикам» [5]. Таким образом, автор говорит о том, что ИТ-архитектура зависит от программных решений и от коллегиальной согласованности их важности. На основе этой идеи было сформулировано следующее определение: ИТ-архитектура – это совокупность взаимосвязанных программно-аппаратных решений и компонентов, считаемых важными по групповому консенсусу и обеспечивающих эффективное функционирование бизнеса. Однако в рамках данной статьи будет использовано определение, описанное в стандарте ISO/IEC 42010:2011 и также используемое в фреймворке TOGAF: «Архитектура (системы) – это фундаментальная организация систем, воплощенная в ее компонентах, их отношением друг с другом и окружающей средой, а также совокупность руководящих принципов проектирования и эволюции» [6].

Сама по себе архитектура представляет собой некоторую абстракцию, состоящую из понятий и свойств, которая наилучшим образом воспринимается через множественные представления. Затрагивая тему областей представления, ранее упомянутый TOGAF предлагает использовать концепцию доменов архитектуры, которая изображена на рисунке 1. Сама концепция сильно напоминает модель представления архитектуры 4+1, состоящую из логического, процессного, физического представлений, а также представления разработки и описания вариантов использования. Но в отличие от модели 4+1, которая создавалась преимущественно для описания программных систем, TOGAF не ограничивается вариантами использования, добавляя полноценный слой бизнес-архитектуры [7]. Для корректного понимания фундаментальных различий необходимо рассмотреть каждый слой корпоративной архитектуры по TOGAF подробнее:

1. Бизнес-архитектура – это описание модели бизнеса, того, как компания зарабатывает деньги. Для оценки своего бизнеса компания выставляет бизнес-цели, для достижения которых ей необходимы определенные компетенции. В основе компетенций лежат процессы, которые создают ценности как для внешних, так и для внутренних клиентов в целом. В свою очередь процессы представляются через призму продуктов, потребительских сегментов и отношений и фиксируются в цифровом виде.

2. Архитектура данных – это часть ИТ-архитектуры, описывающая структуру данных. У объекта системы может быть разный бизнес-контекст, однако для автоматизированной системы это всего лишь набор атрибутов, собранных в сущность и объединенных определённым признаком. Как раз организация хранения, доступа и администрирования этих данных и является основной задачей слоя.

3. Архитектура приложений отвечает за создание карты автоматизированных систем, на которой отображается связь между заявленными бизнес-компетенциями и разработанными приложениями. Данный слой отвечает за «упаковку» бизнес-процесса в набор программных продуктов и сервисов с учетом разработанных в организации архитектурных стандартов (например, учет требований к защите персональных данных).

4. Технологическая архитектура описывает возможности аппаратного обеспечения и инфраструктуры в целом, те мощности, ресурсы и технологии, которые может выделить предприятие в рамках доступного бюджета для автоматизации заявленных бизнес-процессов.

Стоит заметить, что сам по себе фреймворк не является предписывающим сводом правил. TOGAF – это, скорее, инструмент, хранящий набор практик, выбранных авторами

библиотеки. Эти практики могут быть адаптированы под нужды конкретного предприятия. Самый яркий пример – выделение крупными компаниями пятого домена – интеграционная архитектура [8].

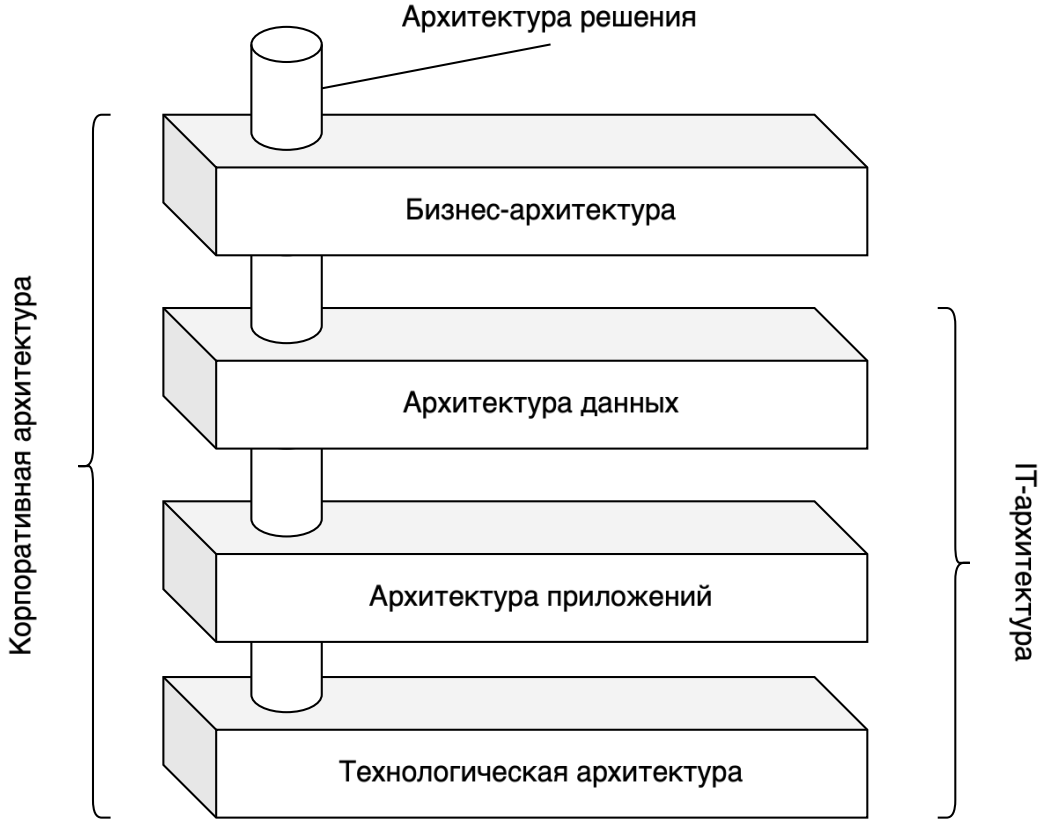

Рис. 1. *Корпоративная архитектура по TOGAF. Источник [3]*

Fig. 1. *Enterprise architecture according to TOGAF. Source [3]*

Результаты исследования

Современный подход к проектированию архитектуры сталкивается с рядом проблем. Одна из них заключается в существенных затратах по времени и ресурсам на исследование архитектурного ландшафта. Важно отметить, что архитектор как роль появляется в компании только при необходимости внесения целевых изменений в некоторую архитектуру [9]. Любое внесение изменений в архитектуру реализуется в рамках некоторого проекта, который контролируется, как правило, каскадной методологией управления. Основные этапы такого проекта включают в себя: погружение в предметную область, изучение бизнес-контекста и целей, описание предприятия в модели AS-IS, проведение GAP-анализа, разработка целевой архитектуры в модели TO-BE, прохождение этапа согласования со всеми заинтересованными сторонами. И только после всех этих шагов архитектора допускают до подготовки плана реализации [10]. В таких проектах всегда четко определены сроки и ресурсы, изменить которые после подписания договора практически невозможно. Так появляется первая составная часть проблемы – архитектор зачастую неспособен познать исследуемый архитектурный ландшафт организации в установленный срок. В свое время некоторые зоны предприятия выходят за пределы осознания или за рамки доступности архитектора. И, наконец, третья уязвимость заключается в том, что архитектор работает со «снимком» модели предприятия на конкретную дату. Пока он тратит колоссальные ресурсы на

познание этого «снимка», архитектурный ландшафт меняется. Соответственно, по окончании проекта описание архитектуры заведомо устаревшее, выбранные архитектурные паттерны для реализации на предприятии текущего «снимка» неактуальны, а полученные артефакты не несут никакой ценности. Для выхода из такого рода ситуаций архитекторам приходится упрощать предметную область, принимая во внимание все перечисленные сложности [11]. Так команда проектирования архитектуры старается упростить процесс познания ландшафта, даже с учетом того факта, что его сложность от этого не изменится.

Отчасти в решении этой проблемы может помочь сегментный подход к проектированию. Такой подход сосредотачивается на главных отраслях для бизнеса, подразумевает постепенное введение понятия архитектуры в компанию и позволяет добиться быстрой отдачи от проекта. Но это не решает следующей проблемы – отсутствие надежных средств донесения архитектурного замысла. Здесь стоит подробнее остановиться на еще одном инструменте компании The Open Group. ArchiMate – это открытый и независимый язык моделирования архитектуры предприятия для поддержки описания, анализа и визуализации архитектуры внутри и за пределами бизнес-процессов. ArchiMate – это самостоятельная концепция, однако наиболее точно она соответствует архитектурному подходу TOGAF. Разрабатывая ArchiMate, авторы планировали создать унифицированный описательный инструмент, не требующий глубоких познаний в языке, интуитивно понятный и применимый для всего. Действительно, в отличие от того же ARIS, который состоит из 80 различных элементов, ArchiMate насчитывает всего 15, однако едва ли ArchiMate можно назвать естественным языком для восприятия. Возникает критическая проблема интерпретации, которую невозможно решить издательством новой нотации. Проблема затрагивает ситуацию, при которой архитектор сформировал представление о функционировании системы, воплощенное в ряде артефактов, а разработчик не смог корректно интерпретировать заявленные требования [12]. Странно, что разработчик, который является потребителем этих схем, написанных на естественном для восприятия языке картинок, не может их однозначно декодировать или даже понять.

Стоит заметить, что данная проблема не является признаком некомпетентности разработчика, потому как современные стандарты насчитывают огромное количество моделей представления, которые он вправе не знать. Как итог реализация идет вразрез с задумкой, что является в первую очередь проблемой для самого архитектора. Данный инцидент запускает целую цепочку событий, среди которых: доработка артефактов с учетом обратной связи, обновление презентационных материалов, разработка новых моделей для формирования комплексного видения конечного результата у исполнителя, привлечение заинтересованных сторон, а также многократные встречи с целью донесения концепции. Этот инцидент имеет цикличный характер и тратит колоссальное количество человеко-часов всех вовлеченных на разбор блокирующих ситуаций [13]. Проблема затратности процесса донесения замысла – лишь составная часть комплексной проблемы. К сожалению, все потраченные ресурсы никак не гарантируют «попадание» финальной реализации в планируемый вариант. По сути, по результатам разработки архитектор получит «черный ящик», так как на данный момент современный подход к проектированию архитектуры не предусматривает контроль реализации решения и не дает надежных средств контроля финального результата.

Последняя рассматриваемая проблема относится больше к организации процесса проектирования архитектуры, чем к подходу, однако она имеет прямое влияние на эффективность самого процесса. Оснований для проблемы два: 1) неправильное понимание роли архитектора, 2) отсутствие стандартов управления проектированием. Выше уже затрагивалась ситуация, когда компаниям действительно нужен архитектор. Стоит дополнить, что на практике реального архитектора можно застать только в крупных компаниях уровня enter-

prise. В остальных организациях под архитектором, как правило, понимают опытного senior-разработчика, который имеет хорошее представление о функционировании системы на самых низких уровнях. Результатом деятельности такого специалиста обычно являются схемы, производные от диаграмм компонентов. Вся документация, полученная в рамках деятельности, разбивается между множеством проектных репозиториев и вики-систем. Получить целостное описание архитектуры или неразрывную логическую связь с полным набором нефункциональных требований практически невозможно [14].

При таком подходе хранения артефактов архитектор становится уникальным и эксклюзивным носителем знаний, ведь полная структурированная архитектура существует только в представлении архитектора и нигде не зафиксирована. С одной стороны, это неудобно, ведь получить какое-либо структурированное представление можно только в словесной форме после ряда уточняющих консультаций. С другой стороны, это критический риск, так как данный архитектор обладает исключительным знанием, и уход этого специалиста из компании повлечет за собой серьезные расходы. В заключение это образует тенденцию, при которой потребители архитектуры обращаются напрямую к архитектору, даже не пытаясь самостоятельно разобраться с требованиями.

Обсуждение результатов

Некоторое время назад активно набирал популярность термин ArchOps, который отражает возможный путь развития концепции DevOps. По сути, ArchOps подразумевает добавление нового узла в методику CI/CD (рис. 2), модернизирующее существующий производственный процесс разработки функциональности. Основная задача подхода – не просто дополнить существующий процесс, а решить глобальную проблему вовлеченности заинтересованных сторон. Эту задачу можно решить объединением всех стейкхолдеров в так называемый производственный комбинат – некий единый и непрерывный механизм управления проектированием, реализацией и внедрением.

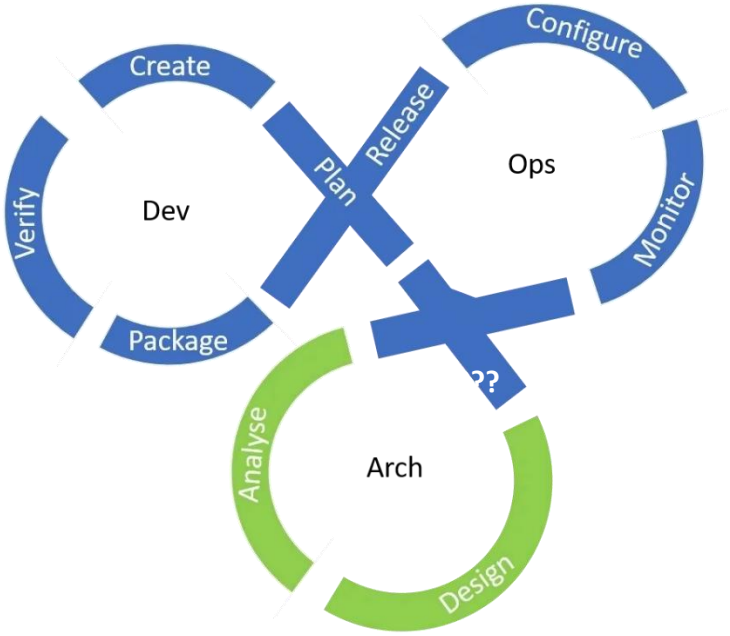

Рис. 2. *Диаграмма ArchOps.*
Составлено автором по результатам исследования

Fig. 2. *ArchOps diagram.*
Compiled by the author based on research results

Однако на сегодняшний день интерес к подходу ArchOps сильно упал. Все потому, что исследование в этой области столкнулось с проблемой поиска канала передачи информации от узла Operations к узлу Architecture. В существующей модели архитекторы и их артефакты – это обособленная сущность за рамками процесса разработки. Она получает информацию на вход и в некоторый момент времени отдает артефакты на выход. Складывается вполне прозрачная цель исследования – найти такой канал и изучить особенности его применения, тем самым преодолев ключевой барьер внедрения подхода ArchOps.

В рамках исследования был найден фреймворк «Architecture as a Code», который лучше всего подходит на роль связующего канала. Для обоснования целесообразности использования фреймворка следует вспомнить историю подхода «Infrastructure as a code», который еще некоторое время назад ставился под серьезное сомнение и считался зависимым от возможностей уникальных специалистов. Опыт современных ИТ-компаний разного уровня показал, что IaaC – это рутина. Беря во внимание факт успешного применения практики переиспользования кодовых баз с последующей связью и тестовым покрытием, ничего не мешает реализовать это по отношению к архитектуре. Таким образом, можно создавать некоторые архитектурные паттерны, трансформировать это в код, размещать в едином репозитории, а затем по мере необходимости подключать в нужные проекты.

В процессе изучения выяснилось, что для успешного применения концепции необходимо провести следующие подготовительные мероприятия: организация общего пространства и пересмотр матрицы компетенций современного архитектора. Переход на концепцию «Architecture as a Code» требует обучить специалистов обмениваться знаниями через систему единого репозитория, чтобы грамотно интегрировать результаты своей деятельности и социализировать предлагаемые решения. Это позволит не только планомерно развивать архитектуру, но и стимулировать инновации. Достичь эту цель представляется возможным только через организацию единого цифрового пространства, где не только архитекторы, но и любые люди, принимающие участие в архитектурных изменениях, могли бы договориться об изменениях на некотором универсальном языке.

На самом деле для реализации такого единого репозитория можно использовать существующие практики. Прародителями такого пространства являются репозитории программного кода и средства виртуализации. Ядром практической реализации является система контроля версий вместе с инструментами переиспользования программного кода. Успешность применения Git в opensource-продуктах доказывает возможность организации социального взаимодействия вокруг кода с использованием технологии. Таким образом, все необходимые технологии уже существуют и эффективно применяются, дело осталось за малым – описать архитектуру при помощи кода.

Стоит обратить внимание, что PlantUML или Structurizr хоть и выглядят как код архитектуры, но на самом деле им не являются. Можно сказать, что эти языки реализуют концепцию «Diagram as a Code», то есть описывают диаграммы. Сами по себе диаграммы – это не архитектура, а лишь визуализация, так называемый architecture viewpoint. Соответственно, работая с псевдокодом PlantUML, человек работает с картинками, но на ином принципе. Конечно, этот механизм упрощает работу с архитектурными артефактами, но он наследует все проблемы стандартного архитектурного подхода. При возникновении необходимости сравнения двух версий одной диаграммы в написанных на PlantUML, которые при этом будут расположены в разных местах, необходимо будет обратиться к архитектору, зарендерить диаграмму, осознать ее и только после этого можно получить однозначный вывод.

Пример кода архитектуры:

Components: {
System.gateway:
Title: API шлюз
Entity: component
Links: [
{ Id: system.backend
Title: Бизнес API
Direction: <-->
Contract: https://editor.swagger.io/ },
{ Id: system auth
Title auth API
Direction: <-->
Contract: example }]
Technologies: HTTP; OAuth }

Для того чтобы идентифицировать код архитектуры, нужно описать его признаки. В первую очередь код архитектуры должен быть машиноанализируемым, соответственно, архитектура должна быть представлена в формате данных. Все созданные артефакты должны быть анализируемые, чтобы к ним можно было обратиться и сделать определенные выводы. Как и любой программный код, код архитектуры должен быть однозначно интерпретируемым как человеком, так и машиной, чтобы из этого кода можно было сгенерировать не только понятные для программиста схемы, но и конфигурационные файлы, пригодные для автоматизированного развертывания. Помимо этого, код архитектуры должен быть генерируемым, чтобы была возможность собрать шаблон архитектуры приложения на основе проанализированных цифровых следов организации. Наконец, такой код должен поддерживать модульность или сегментируемость, то есть отвечать идеям концепции Domain Driven Design, на чем стоит остановиться подробнее.

Система – это конструкт, в основании которого лежит невозможность воспринять реальность в полном виде, как она есть. Из-за этого человеческий мозг упрощает эту полную картину до нескольких систем. В теории систем существует тезис, что управлять системой может только более сложная система. Ни одна система управления архитектурой не может подчинить себе реальную архитектуру ввиду своей упрощенности. Поэтому необходимо признать, что архитектура в компании – это не однородная субстанция, которая имеет определенные самостоятельные домены, каждый из которых управляется изолированно. Так появился подход «Federated Architecture», основанный на принципе «разделяй и властвуй», благодаря которому в действительности появилась более сложная система, способная покрыть задачи по управлению и проектированию архитектуры [15]. Принцип простой: сначала определяются домены управления архитектуры согласно определенным критериям, затем фиксируются контракты с этими доменами, при которых управление доменом отдается специалистам в конкретной функциональной области. Домен поставляет информацию, которая будет использоваться в мастер-системе, в обмен на необходимые для него ресурсы. Внутри домен может требовать управление архитектурой разного качества, а также выделять и управлять другими доменами внутри себя.

При сегментировании архитектуры и выделении независимых доменов необходимо учитывать поток увеличения объема информации. Для управления этим потоком можно применить концепцию архитектурного DataLake с целью накопления данных управления архи-

тектурой, чтобы впоследствии управлять ее качеством [16]. Унифицированный код позволяет накапливать структурированные данные доменов и консолидировать их. Системы контроля версий дают возможность работать с версиями данных для проведения ретроспективы. А запросы к данным позволяют проводить анализ. Таким образом, появляется методика управления архитектурой саморазвивающихся систем, изображенная на рисунке 3, при которой контролируется только контракт и отсутствует необходимость тотального описания моделей AS-IS.

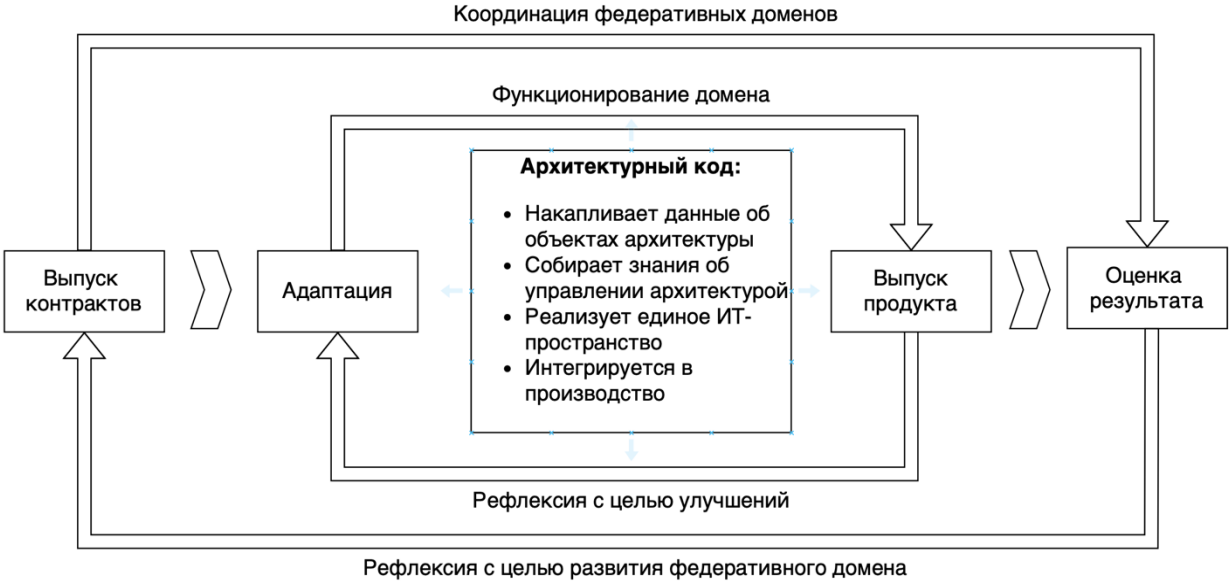

Рис. 3. *Процесс управления федеративной архитектурой. Составлено автором по результатам исследования*

Fig. 3. *Federated architecture management process. Compiled by the author based on research results*

Заключение

Архитектура – это всеобъемлющее понятие, обязательным элементом которого являются требования. Хорошая архитектура возникает на хорошо исследованном ландшафте, так как он становится базой для формулирования действительно значимых решений на основе точных требований. В данном исследовании были выявлены основные проблемы существующих подходов управления архитектурой, а именно: барьер интерпретации, недостаточность функции контроля и обособленность от производственного процесса. Еще недавно каталоги, матрицы и диаграммы считались единственным объективным способом донесения архитектурного замысла, однако восприятие мира изменилось, и теперь программный код интерпретировать стало куда проще и выгоднее. В статье описана локализация архитектурного компонента подхода «Architecture as a Code», который легко читать, воспринимать, размещать, анализировать и обсуждать. Сам по себе программный код легко поддерживать в актуальном состоянии, потому что он может быть как автоматически сгенерирован, так и написан человеком. Для упрощения процесса интеграции нового подхода в управлении ИТ-архитектурой предприятия был разработан авторский план внедрения, основанный на активно применяемых инструментах, таких как: системы контроля версий, механизмы повторного использования кодовых баз, управление само-

стоятельными сегментами системы (доменами). Дополненный новыми практиками предложенный подход «Architecture as a Code» сможет управлять архитектурой саморазвивающихся систем, при этом кратно повышая эффективность предприятия за счет ускорения lead time разработки и уменьшения количества требуемых заказчиком доработок при выводе функционала в промышленную эксплуатацию.

СПИСОК ЛИТЕРАТУРЫ

1. *Андросюк А. Б.* Принципы моделирования архитектуры предприятия // E-Scio. 2019. № 7(34). С. 581–584. EDN: ERTCDB

2. *Погонышева Д. А.* Инновации в управлении бизнес-процессами организации на основе использования информационных технологий // Вестник Брянского государственного университета. 2013. № 3. С. 148–151. EDN: RRYNAD

3. *Джонсон Р.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. М.: Питер, 2013. 368 с. ISBN: 978-5-459-01720-5

4. *Тельнов Ю. Ф., Казаков В. М., Трембач В. М.* Разработка системы, основанной на знаниях, для проектирования инновационных процессов создания продукции сетевых предприятий // Бизнес-информатика. 2020. Т. 14. № 3. С. 35–53. DOI: 10.17323/2587-814X.2020.3.35.53

5. *Фаулер М.* Шаблоны архитектуры корпоративных приложений. Бостон: Addison-Wesley Professional, 2012. 560 с. ISBN: 978-0321127426

6. Харрисон Р. Учебное пособие по основам TOGAF 9. Нидерланды: Van Haren Publishing, 2018. 532 с. ISBN: 978-9087537418. URL: https://publications.opengroup.org/i182 (дата обращения: 23.03.2024)

7. *Штейнгарт Е. А.* Бизнес-процесс как главный компонент архитектуры предприятия // Межвузовский сборник научных трудов «Актуальные проблемы социологии и управления». СПб.: Санкт-Петербургский государственный экономический университет, 2017. С. 90–100

8. *Тельнов Ю. Ф.* Развитие архитектуры цифровых предприятий // Научные труды Вольного экономического общества России. 2021. № 4. С. 230–235. DOI: 10.38197/2072-2060-2021-230-4-230-235

9. *Ильин И. В., Лёвина А. И., Дубгорн А. С.* Цифровая трансформация как фактор формирования архитектуры и ИТ-архитектуры предприятия // Научный журнал НИУ ИТМО. Серия: Экономика и экологический менеджмент. 2019. С. 50–55. DOI: 10.17586/2310-1172-2019-12-3-50-55

10. *Цебренко К. Н.* Анализ архитектуры предприятия с использованием визуальных средств моделирования // Международный журнал гуманитарных и естественных наук. 2021. № 8-1. С. 115–118. DOI: 10.24412/2500-1000-2021-8-1-115-118

11. *Цебренко К. Н.* Совершенствование архитектуры предприятия на основе функционально-структурного моделирования // Международный журнал гуманитарных и естественных наук. 2022. № 7-2. С. 211–214. DOI: 10.24412/2500-1000-2022-7-2-211-214

12. *Салакова Г., Хатджиева О., Сейиткулиев Б., Тангрыбердиева С.* Архитектура однопользовательских и многопользовательских информационных систем малых и корпоративных предприятий // Инновационная наука. 2024. № 1-1. С. 27–28. EDN: FPUZKO

13. *Донец Н. Ю., Клешнева Л. И.* Управление архитектурой предприятий агропромышленного комплекса // Финансовые рынки и банки. 2022. № 12. С. 52–55. EDN: TRRHYX

14. *Свищёв А. В., Попов Г. П.* Исследование и анализ значимости грамотного проектирования ИТ-инфраструктуры и архитектуры предприятия // Международный журнал гуманитарных и естественных наук. 2022. № 11–2. С. 121–124. DOI: 10.24412/2500-1000-2022-11-2-121-124

15. *Блокдик Г.* Федеративная архитектура: полное руководство. Торонто: 5STARCooks, 2021. 305 с. ISBN: 978-0655944164

16. *Курилова А. А., Савенков Д. Л.* К вопросу о диагностике архитектуры предприятия // Азимут научных исследований: экономика и управление. 2022. № 2(39). С. 45–48. DOI: 10.57145/27128482_2022_11_02_09

REFERENCES

1. Androsyuk A.B. Principles of enterprise architecture modeling. *E-Scio*. 2019. No. 7(34). Pp. 581–584. EDN: ERTCDB. (In Russian)

2. Pogonysheva D.A. Innovations in managing business processes of an organization based on the use of information technologies. *Bulletin of Bryansk State University*. 2013. No. 3. Pp. 148–151. EDN: RRYNAD. (In Russian)

3. Johnson R. Techniques of object-oriented design. Design patterns. Moscow: St. Petersburg, 2013. 368 p. ISBN: 978-5-459-01720-5. (In Russian)

4. Telnov Y.F., Kazakov V.M., Trembach V.M. Development of a knowledge-based system for designing innovative processes for creating products of network enterprises. *Biznes-informatika*. Vol. 14. No. 3. 2020. Pp. 35–53. DOI: 10.17323/2587-814X.2020.3.35.53. (In Russian)

5. Fowler M. Patterns of corporate application architecture. Boston: Addison-Wesley Professional, 2012. 560 p. ISBN: 978-0321127426

6. Harrison R. TOGAF 9 Basics Tutorial. Netherlands: Van Haren Publishing, 2018. 532 p. URL: https://publications.opengroup.org/i182 (access date: 03/23/2024). ISBN: 978-9087537418

7. Shteyngart E.A. Business process as the main component of enterprise architecture. *Interuniversity collection of scientific works "Current problems of sociology and management"*. St. Petersburg: St. Petersburg State Economic University, 2017. Pp. 90–100. (In Russian)

8. Telnov Yu.F. Development of architectures of digital enterprises. *Scientific works of the free economic society of Russia*. 2021. No. 4. Pp. 230–235. DOI: 10.38197/2072-2060-2021-230-4-230-235. (In Russian)

9. Ilyin I.V., Levina A.I., Dubgorn A.S. Digital transformation as a factor in the formation of architecture and IT architecture of an enterprise. *Nauchnyj zhurnal NIU ITMO. Seriya: ekonomika i ekologicheskij menedzhment* [Scientific journal of NRU ITMO. Series: economics and environmental management]. 2019. Pp. 50–55. DOI: 10.17586/2310-1172-2019-12-3-50-55. (In Russian)

10. Tsebrenko K.N. Analysis of enterprise architecture using visual modeling tools. *International journal of humanities and natural sciences*. 2021. No. 8-1 (59). Pp. 115–118. DOI: 10.24412/2500-1000-2021-8-1-115-118. (In Russian)

11. Tsebrenko K.N. Improving enterprise architecture based on functional-structural modeling. *International journal of humanities and natural sciences*. 2022. No. 7-2. Pp. 211–214. DOI: 10.24412/2500-1000-2022-7-2-211-214. (In Russian)

12. Salakova G., Khatdzhieva O., Seyitkuliev B., Tangriberdieva S. Architecture of single-user and multi-user information systems of small and corporate enterprises. *Innovative Science*. 2024. No. 1-1. Pp. 27–28. EDN: FPUZKO. (In Russian)

13. Donets N.Y., Kleshneva L.I. Architecture management of agricultural industrial comple. *Financial markets and banks*. 2022. No. 12. Pp. 52–55. EDN: TRRHYX. (In Russian)

14. Svishchev A.V., Popov G.P. Research and analysis of the significance of compotent design of IT infrastructure and enterprise architecture. *International journal of humanities and natural sciences*. 2022. No. 11–2. Pp. 121–124. DOI: 10.24412/2500-1000-2022-11-2-121-124. (In Russian)

15. Blokdik G. Federated architecture: A complete guide. Toronto: 5STARCooks, 2021. 305 p. ISBN: 978-0655944164

16. Kurilova A.A., Savenkov D.L. On the question of diagnostics of enterprise architecture. *Azimuth of scientific research: economics and administration*. 2022. No. 2(39). Pp. 45–48. DOI: 10.57145/27128482_2022_11_02_09. (In Russian)

Финансирование. Исследование проведено без спонсорской поддержки.

Funding. The study was performed without external funding.

Информация об авторе

Шмонин Михаил Андреевич, студент института информационных систем, Государственный университет управления;
109542, Россия, Москва, Рязанский проспект, 99;
mike.shmonin@mail.ru, ORCID: https://orcid.org/0009-0003-4756-1534

Information about the author

Mikhail A. Shmonin, Student of the Institute of Information Systems, State University of Management;
109542, Russia, Moscow, Ryazansky prospect, 99;
mike.shmonin@mail.ru, ORCID: https://orcid.org/0009-0003-4756-1534