

рынке, а эффективность их использования определяется ценностью конечных потребительских благ, которые могут быть произведены с их помощью. Аргумент Хайека, согласно Салерно, постулирует только сложность проблемы размещения ресурсов, а не невозможность рационально осуществлять данную процедуру. При этом стоит отметить, что дальнейшие работы Хайека о социализме, выходящие до 1980-х годов, игнорируются Салерно.

Перспективным направлением выглядит исследование и сравнение других работ, посвящённых критике антисоциалистических аргументов Хайека другими современными представителями австрийской школы экономики, среди которых Х.У. де Сото, Й.Г. Хюльсманн, Г.-Г. Хоппе.

Литература:

1 Мизес, Л. фон. Социализм: экономический и социологический анализ / Л. фон Мизес ; пер. с англ. Б.С. Пинскера ; науч. ред. Р. Левита. – Москва ; Челябинск : Социум, 2016. – 598 с.

2 Mises, L. von. Economic Calculation in the Socialist Commonwealth / L. von Mises ; Translated from German by S. Adler. – Ludwig von Mises Institute, 2008. – xviii + 69 p.

УДК 336.719+004

В.Е. Терентьев

Общество с ограниченной ответственностью «Скартел»

Санкт-Петербург, Россия

Белорусский государственный университет

Республика Беларусь, Минск

vadim.terentev.26s@gmail.com

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ЦИФРОВЫХ СЕРВИСОВ КАК ОБЯЗАТЕЛЬНЫЙ ЭЛЕМЕНТ ПОСТРОЕНИЯ ЭКОНОМИЧЕСКИ УСТОЙЧИВОЙ ЭКОСИСТЕМЫ СЕРВИСОВ

Terentyev V. E. Limited liability company «Scartel» St. Petersburg, Russia, Belarusian State University, Republic of Belarus, Minsk.

MICROSERVICES ARCHITECTURE OF DIGITAL SERVICES AS A REQUIRED ELEMENT FOR BUILDING ECONOMICALLY SUSTAINABLE ECOSYSTEM OF SERVICES. The paper describe a possibility to build more sustain-

able form of digital services that are residents of the High-Tech Park using microservices architecture as required aspects for building ecosystem of services. In paper there are comparing two model of architecture for building digital services in ecosystems context, base on this was form recommendations to architecture digital services startup projects.

KEYWORDS: Startup project, ecosystem of services, microservices architecture, digital economy, sustainable form of digital services.

В данной статье рассматривается возможность формирования более устойчивых форм цифровых продуктов и сервисов, являющихся резидентами Парка высоких технологий, с использованием микросервисной архитектуры, как необходимого аспекта при формировании экосистемы сервисов. В статье приведено сравнение двух моделей архитектуры для построения цифровых сервисов в контексте экосистем, на основе чего приведены рекомендации к архитектуре цифровых сервисов стартап-проектов.

КЛЮЧЕВЫЕ СЛОВА: стартап-проект; экосистема сервисов; микросервисная архитектура; цифровая экономика; устойчивые формы цифровых сервисов.

Стремительное развитие цифровой экономики Республики Беларусь, стимулированное Декретом №8 Президента Республики Беларусь от 21.12.2017 «О развитии цифровой экономики», приводит к увеличению числа стартап-проектов, направленных на создание цифровых продуктов и сервисов. Но по статистике 9 из 10 стартап-проектов терпят неудачу, по различным причинам. Одним из решением данной проблемы может стать объединение стартапов в экосистемы. Практика построения экосистем цифровых продуктов и сервисов уже активно внедряется крупными компаниями, как например Яндексом. Термин экосистема появился в биологии. Определение данное А. Тенсли: «Экосистема – это совокупность живых организмов (сообществ) и среды их обитания, образующих благодаря круговороту веществ, устойчивую систему жизни» [1].

Экосистема цифровых сервисов во многом схожа с биологической экосистемой. Основные составляющие экосистемы – это цифровые сервисы. То, что определяет будет ли существовать цифровой сервис или нет, это обеспечение актуальных пользовательских потребностей. Таким образом, можно сказать, что средой обитания цифровых сервисов являются потребности пользователей.

Экосистема также более устойчивая система жизни по сравнению с отдельным цифровым сервисом. Чтобы разобраться с основным компонентом цифрового сервиса, необходимо разделить существующие определения термина цифрового сервиса:

«Цифровой сервис – это сервис, который предоставляется клиенту посредством онлайн-каналов».

Чтобы цифровые сервисы воспринимались как единое целое, они должны быть выставлены для пользователя через один интерфейс: одно мобильное приложение или один сайт, с доступом ко всем сервисам.

С точки зрения ТРИЗ (теория решения изобретательских задач) ситуация будет идеальна, когда к сервисам можно будет получать доступ без специальных интерфейсов, только с использованием человеческих органов чувств без прослоек [2].

Для объединения сервисов под одним интерфейсом, необходимо более тесная интеграция, что влечет увеличение связей между компонентами на уровне приложения и инфраструктуры, а это увеличивает сложность образующейся системы. Для работы со сложными информационными системами используются архитектурные подходы. До недавнего времени существовало два подхода для построения архитектуры приложений: монолитная архитектура и сервис ориентированная архитектура. Монолитная архитектура – подход к построению программного приложения в виде большого связанного модуля, где все компоненты спроектированы так, чтобы работать вместе друг с другом, используя общую память и ресурсы. Сервис-ориентированная архитектура — модульный подход к разработке программного обеспечения, основанный на использовании распределенных, слабо связанных заменяемых компонентов, оснащенных стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.

В 2002 года компания Амазон стала переходить с монолитной архитектуры на набор слабосвязанных сервисов. И в 2012 году был описан новый подход к построению приложений в докладе Криса Ричардсона «Декомпозиция приложений для улучшения развертываемости и масштабируемости» [3]. Данный подход был назван микросервисной архитектурой.

Микросервисная архитектура была предложена для использования в рамках отдельно взятого цифрового сервиса – приложения. Так как при развитии продукта сложность его кодовой базы увели-

чивается, и если при начале работы над продуктом монолитная архитектура наиболее привлекательна, то с ростом приложения, разработка продукта сталкивается с серьезными проблемами. Крис Ричардсон описал их в своей книге «Микросервисы. Паттерны разработки и рефакторинга» [4]:

- высокая сложность кодовой базы пугает разработчиков;
- медленная разработка (большой объем кодовой базы замедляет работу IDE);
- длинный и тяжелый путь от сохранения изменений до их развертывания (так как приходится развертывать все приложения при обновлении любого модуля);
- трудности с масштабированием (некоторые модули требуют большой вычислительной мощности, другим требуется большой объем оперативной памяти);
- сложность добиться надежности монолитного приложения (все модули приложения взаимосвязаны и сбой в одном модули могут приводить к сбоям в других модулях, что трудно протестировать);
- зависимость от постепенно устаревающего стека технологий.

Таким образом, наступает момент, когда компании для продолжения развития своего цифрового сервиса требуется увеличивать затраты на штат ит-специалистов, так как требуются специалисты с более высокой квалификацией, которые смогут справиться с общей сложностью системы, хотя сложность бизнес-задач будет оставаться на прежнем уровне, но при этом скорость вывода новой функциональности на рынок все равно будет замедляться, что является неэффективным. Микросервисная архитектура помогает решить перечисленные проблемы монолитной архитектуры. Сложность внутреннего устройства сервиса также влияет на его интеграцию с другим. На этапе существования и функционирования экосистема должна постоянно изменяться, так как именно гибкость и адаптируемость элементов на изменения среды их обитания и изменения окружающих элементов определяют устойчивость экосистемы. Рассмотрим упрощенную архитектуру экосистемы, основанную на 3-ех монолитных приложениях на компонентной диаграмме (рисунок 1).

Вся сложность данной системы сосредоточена в компонентах приложений, которые выполняют всю техническую и бизнес логику цифрового сервиса. Для каждой из трех компонент экосистемы характерны проблемы, описанные для монолитного приложения.

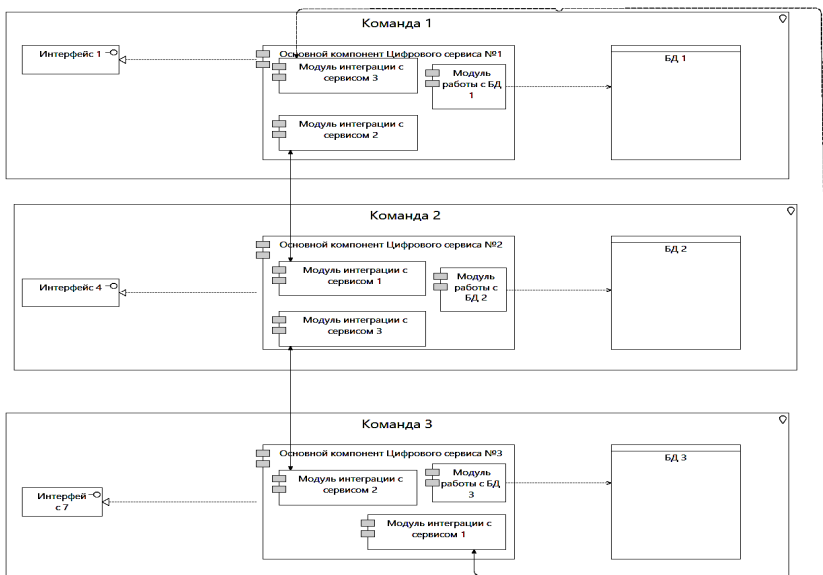


Рисунок 1 – Модель архитектуры, основанной на монолитной архитектуре приложений

Эти проблемы усугубляются тем, что изменения одного компонента приводят к изменениям другого, а с учетом минусов монолитных приложений изменение компонента является болезненным процессом для команды. Так как предполагается, что клиент один на все сервисы экосистемы, компоненты будут дублировать логику, связанной с идентификацией клиента, с хранением информации о клиенте и так далее.

Рассмотрим упрощенную архитектуру, основанную на микросервисном подходе (рисунке 2).

Микросервисная архитектура приводит к увеличению сложности системы на уровне архитектуры, для ее поддержки требуются работы системных архитекторов. Также увеличивается количество работы по установке сервисов на сервера, что увеличивает работу инженеров эксплуатации и сопровождения. Но если проблемы сложности и объема решить программными средствами пока не удастся, то решать однотипные задачи эксплуатации возможно с помощью специального программного обеспечения.

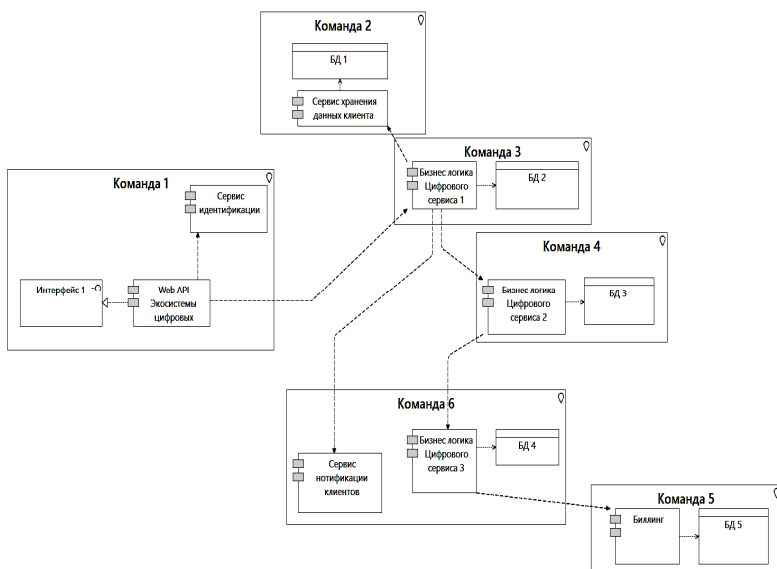


Рисунок 2 – Модель архитектуры, основанной на микросервисной архитектуре

Таким образом, мы приходим к тому, что при появлении сложных цифровых сервисов и объединение их в экосистему, требуется распределять сложность системы, уменьшая сложность кодовой базы, путем выделения функциональных блоков в отдельный компонент, таким образом, увеличивая сложность архитектуры решения.

Для стартап-проектов, которые разрабатывают свои цифровые сервисы, предложенные выводы могут стать определяющими для архитектуры их сервиса, а также определить дальнейший путь развития через интеграции с другими цифровыми сервисами стартап-проектов.

Литература:

1 Лисов, Н.Д. Общая биология: учеб. Пособие / Н.Д. Лисов, Л.В. Камлюк, Н. А. Лемеза – Мн.: Беларусь, 2002. – 279 с.

2 Альтшуллер, Г.С. Найти идею / Г. С. Альтшуллер. – Москва: Альпина, 2019. – 404 с.

3 SlideShare [Electronic resource] : Decomposing applications for deployability and scalability. – Mode of access: <https://www.slideshare.net/chris.e.richardson/decomposing-applications->

for-scalability-and-deployability-april-2012. – Date of access: 17.09.2019.

4 Ричардсон, К. Микросервисы. Паттерны разработки и рефакторинга: книга / К. Ричардсон. – СПб : Питер, 2019. – 544 с.

УДК 365.244

И.В. Шанюкевич

Белорусский национальный технический университет

Республика Беларусь, Минск

shaniukevich@gmail.com

ДОСТУПНОСТЬ ЖИЛЬЯ: ВЗГЛЯД ПОТРЕБИТЕЛЯ

Shaniukevich I.V. Belarusian National Technical University, Republic of Belarus, Minsk.

AFFORDABILITY OF HOUSING: A VIEW OF THE CONSUMER. In the article are presented the results of a survey on housing affordability for the population of the Republic of Belarus. It is analyzed official statistics on satisfaction with their housing conditions. Some conclusions and suggestions have made to increase the affordability of housing.

KEYWORDS: housing conditions, housing affordability, residential real estate market.

В статье представлены результаты проведенного опроса по доступности жилья для населения Республики Беларусь. Анализируются официальные статистические данные по удовлетворенности ими своими жилищными условиями. Сделаны ряд выводов и предложений по повышению доступности жилья.

КЛЮЧЕВЫЕ СЛОВА: жилищные условия, доступность жилья, рынок жилой недвижимости.

В национальных нормативно-правовых документах закреплены права человека на достойный уровень жизни. В соответствии со ст. 21 Конституции Республики Беларусь, каждый имеет право на достойный уровень жизни, включая достаточные питание, одежду, жилье и постоянное улучшение необходимых для этого условий. Для более полного понимания возникающих проблем в ходе реализации государственных программ по обеспечению качественным и доступным жильем необходимо, в том числе в каждом регионе