

РАЗРАБОТКА МОДЕЛЕЙ ЦИФРОВЫХ СЕРВИСОВ В МАШИНОСТРОЕНИИ С ИСПОЛЬЗОВАНИЕМ УНИВЕРСАЛЬНОГО АЛГОРИТМА

DEVELOPMENT OF MODELS OF DIGITAL SERVICES IN MECHANICAL ENGINEERING USING A UNIVERSAL ALGORITHM

**D. Kilyakov
V. Petrov**

Summary. Digital services offer many advantages over traditional solutions, including ease of use, high data processing speed and extensive scaling capabilities. Due to their versatility and functionality, they have become an integral part of the daily life of a modern person, providing access to information and providing various services at any time and place convenient for the user.

At the stage of designing a digital service model, it is extremely important to take into account possible limitations and difficulties that may arise at the stage of its scaling and integration with other information systems. This approach minimizes the risks and financial costs of eliminating potential problems in the future, thereby ensuring the successful implementation and operation of the service.

One of the key stages in the process of designing a digital service is the development of an algorithm that ensures quality control of the model of the service being designed. This algorithm should have high flexibility and adaptability, allowing it to take into account all possible changes and new requirements in the process of developing and using the service. This guarantees the successful implementation of the project and the satisfaction of the needs of all stakeholders.

Keywords: digitalization, digital services, modeling, standardization, mechanical engineering.

Киляков Данил Александрович

Аспирант, ФГБОУ ВО «Московский государственный технологический университет «Станкин»
den1997@rambler.ru

Петров Валерий Евгеньевич

к.т.н., доцент, ФГБОУ ВО «Московский государственный технологический университет «Станкин»
cu58@mail.ru

Аннотация. Цифровые сервисы предлагают множество преимуществ по сравнению с традиционными решениями, включая удобство использования, высокую скорость обработки данных и широкие возможности масштабирования. Благодаря своей универсальности и функциональности, они стали неотъемлемой частью повседневной жизни современного человека, обеспечивая доступ к информации и предоставляя различные услуги в любое удобное для пользователя время и месте.

На этапе проектирования модели цифрового сервиса крайне важно учесть возможные ограничения и сложности, которые могут возникнуть на этапе его масштабирования и интеграции с другими информационными системами. Такой подход позволяет минимизировать риски и финансовые затраты на устранение потенциальных проблем в будущем, обеспечивая тем самым успешное внедрение и эксплуатацию сервиса.

Одним из ключевых этапов в процессе проектирования цифрового сервиса является разработка алгоритма, обеспечивающего контроль качества модели проектируемого сервиса. Данный алгоритм должен обладать высокой гибкостью и адаптивностью, позволяющими ему учитывать все возможные изменения и новые требования в процессе разработки и использования сервиса. Это гарантирует успешную реализацию проекта и удовлетворение потребностей всех заинтересованных сторон.

Ключевые слова: цифровизация, цифровые сервисы, моделирование, стандартизация, машиностроение.

Введение

Цифровой сервис представляет собой интегрированную систему, основанную на использовании цифровых технологий, которая предназначена для значительного улучшения или ускорения основных процессов, обеспечивающих жизнедеятельность человека, включая организационные и бизнес-процессы, а также производственные процессы. Цифровые сервисы обладают рядом преимуществ перед классическими решениями, такими как удобство, более высокая скорость обслуживания и гибкость, позволяющая легко адаптировать систему под конкретные нужды.

Разработка цифровых сервисов предполагает предварительное проектирование модели, что позволяет

предотвратить возможные ограничения при масштабировании и интеграции с другими системами. Это особенно важно, поскольку устранение таких ограничений впоследствии может повлечь за собой значительные финансовые и временные затраты, которых можно было бы избежать, проведя тщательное планирование и проектирование на начальном этапе. Таким образом, грамотное проектирование является ключевым фактором, обеспечивающим эффективность и успешность цифрового сервиса в долгосрочной перспективе.

Последствия ошибок моделирования

Ошибки моделирования цифровых сервисов могут привести к различным негативным последствиям, таким как:

- Снижение эффективности бизнес-процессов: Неправильное моделирование цифрового сервиса может привести к тому, что он не будет выполнять свои функции должным образом, что в свою очередь может снизить эффективность работы компании.
- Увеличение затрат на разработку и внедрение: Ошибки в моделировании могут привести к необходимости переделки или доработки сервиса, что может потребовать дополнительных затрат.
- Ухудшение репутации компании: Если цифровой сервис не работает должным образом или вызывает проблемы у пользователей, это может негативно сказаться на репутации компании.

Чтобы избежать этих проблем, необходимо тщательно подходить к процессу моделирования цифровых сервисов, использовать качественные инструменты и методы моделирования, а также проводить тестирование и отладку моделей.

Сроки, сложность и стоимость разработки могут увеличиваться из-за ошибок в моделировании и постановке задачи, ниже приведены примеры таких ошибок:

1. Рассмотрим сервис, который отображает геолокацию товара или услуги на карте при помощи API Яндекс или Google карт. Архитектура сервиса априори завязана на сторонние решения. При возникновении специфических требований, которые сторонние сервисы не предоставляют, возникла необходимость в дополнительных действиях: нужно установить движок карты, сменить базу данных, сконвертировать данные, написать нужные формулы, адаптировать старый код под возможности модуля карт. Это повлечет за собой временные и денежные затраты.
2. Имеем неправильно смоделированную базу данных, необходимо добавить несколько полей и изменить программный код.
3. Неправильно составлено техническое задание: необходимо полное переписывание.
4. Неправильно выбраны технологии для разработки: при реализации многозадачного многопоточного приложения с использованием технологии, которая для этого не предназначена, в выбранном фреймворке нет необходимых возможностей.

Последствиями ошибок проектирования как правило становятся — увеличение расходов, вплоть до 100 % от изначального времени разработки и бюджета и даже, превышать 100 %. Для того, чтобы модель учитывала все нюансы, необходим алгоритм, который позволит гарантировать качество модели проектируемого сервиса.

Существующие решения для избежания потерь при моделировании

Различные специалисты и IT-компании пытаются решить задачи при возникновении таких проблем.

Одним из решений можно рассмотреть концепцию, используемую компанией AvitoTech — PaaS (платформа как сервис).

Этот подход предлагает следующее:

- Максимальная автоматизация;
- Построение продукта, который закрывает реальные потребности разработчиков;
- Низкий порог входа и максимальная концентрация на скорости доставки фич;
- Нулевой перерасход интеграции с инфраструктурой.

В PaaS для разработчиков от Avito, реализованы и прописаны следующие этапы:

1. Создание сервиса;
2. Разработка сервиса;
3. Тестирование;
4. Доставка сервиса;
5. Эксплуатирование сервиса.

Наиболее подробно предлагается рассмотреть проблематику этапа № 2. Несмотря на то, что подход, описанный выше, позволяет структурировать разработку сервисов и сделать их унифицированными, во многом он касается лишь части, связанной с разработчиками.

В итоге, получаются сервисы *от разработчиков для разработчиков*.

Но, остается проблема, как заранее оценить их трудоемкость, какой уровень проработки интерфейса у них должен быть и согласен ли с итоговой оценкой заказчик.

Отчасти, данная проблема решается прохождением по всем этапам абстракции, как это описано в книге Роберта Мартина «Чистая архитектура»[8]. Р. Мартин предлагает: ввести метрики, полностью основанные на том, как модули системы связаны между собой.

Цель этих метрик — определить, насколько масштабируема и удобна в обслуживании система.

Алгоритм моделирования цифровых сервисов

Алгоритм моделирования цифрового сервиса включает в себя несколько этапов, каждый из которых имеет свои особенности. Первым этапом является определение требований к конечному результату — цифровому сервису. На этом этапе необходимо понять, какие функции и возможности должен предоставлять сервис, какие задачи он должен решать и для каких пользователей он предназначен.

Далее следует этап разработки структуры цифрового сервиса. Этот этап включает в себя определение состава

ва микросервисов и их взаимодействия, определение ролей пользователей и схемы отказоустойчивости сервиса. Также на этом этапе разрабатывается предполагаемая метрика для оценки качества работы сервиса.

Наконец, последним этапом является оценка трудоемкости разработки цифрового сервиса. На основе детальной проработки составных частей сервиса определяется объем работы, необходимый для его создания, и оценивается стоимость разработки.

Преимущества алгоритма моделирования цифровых сервисов заключаются в его универсальности и полноте.

Предлагаемый алгоритм должен соответствовать следующим условиям:

1. Должен быть «Универсальным», применяться для любого вида цифрового сервиса;
2. Включает проектирование модели бизнес-процесса сервиса и ПО;
3. Итерационный, который будет позволять контролировать детализацию итоговой модели;
4. Итоговая модель, должна соответствовать, современным требованиям к ПО и бизнес-процессам;
5. Гарантировать качество итоговой модели, с контролируемым процентом (%) ошибок проектирования, в зависимости, от детализации итоговой модели.

Опираясь на предъявленные к алгоритму требования, был разработан алгоритм, описанный ниже:

Шаг 1

«Сформулировать бизнес-цель цифрового сервиса» — является самым важным и основным во всем процессе. Определение итоговой цели цифрового сервиса влияет на все аспекты моделирования. Результатом выполнения данного этапа должно являться четкое описание решаемой задачи.

Шаг 2

«Разбиение сервиса на микросервисы». Основой для данного этапа является «Принцип единственной ответственности» методологии SOLID [8]. Микросервисы, решающие единственную задачу, позволяют выполнять эту задачу максимально эффективно, а также не влиять на реализацию других. Благодаря этому принципу, итоговый цифровой сервис, будет масштабируемым и прост в отладке.

Шаг 3

«Оценка компетенции пользователя и анализ сценария его работы». На этом этапе осуществляет сбор информации о каждом пользователе, который будет взаимодействовать с сервисом, в разрезе каждого микросервиса. Важность данного этапа заключается

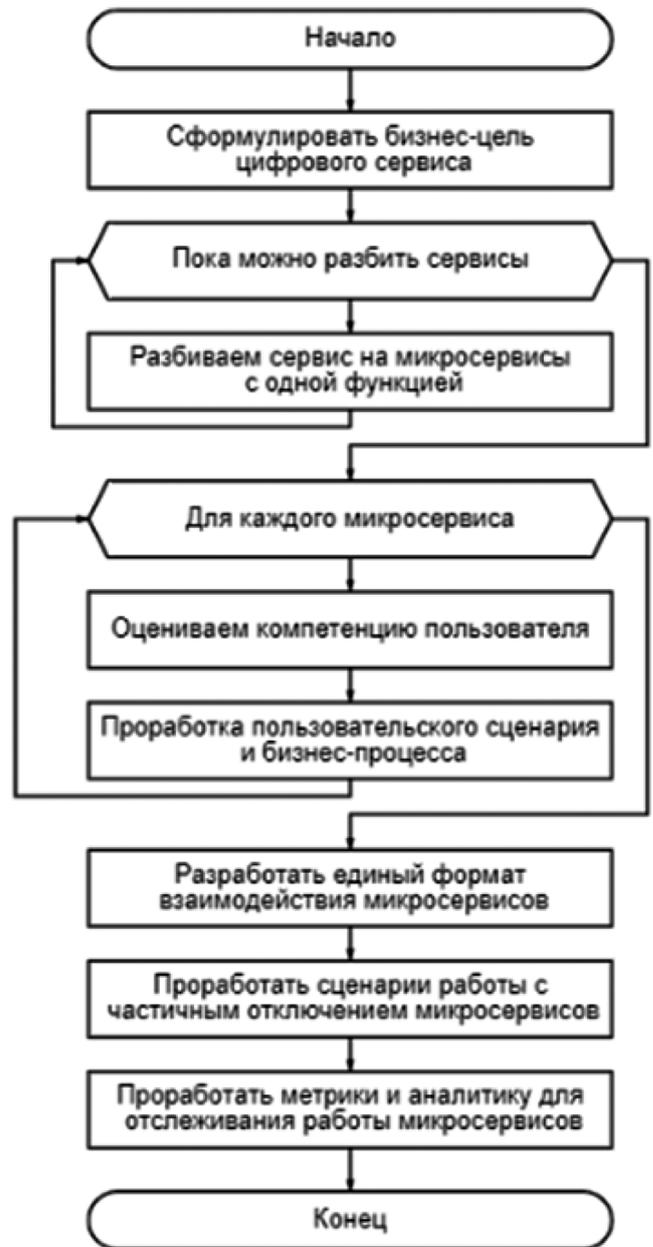


Рис. 1. Блок-схема алгоритма

в том, чтобы пользователь мог эффективно и полно использовать предполагаемый функционал, не возникало трудностей с пониманием интерфейсов и пониманием доступных возможностей. При моделировании сервиса без оценки пользователей, во многих случаях возникает сложность с использованием конечного сервиса или же отсутствием у пользователей необходимых компетенций. В случае, если для сервиса требуется компетенция, которая на текущий момент отсутствует у пользователей, необходимо заложить его обучение.

Шаг 4

«Разработать единый формат взаимодействия микросервисов». Ценность этапа заключается в унифика-

ции всех взаимодействий между микросервисами, что позволяет избежать расхождений и ошибок в обменах данными. В зависимости от сложности конечного сервиса, описание формата, может быть различным. Формат, может включать в себя несколько вариантов, в зависимости от требуемых задач, но порядок их использования, должен быть четко завязан на конкретные условия.

Шаг 5

«Проработать сценарии работы с частичным отключением микросервисов». Во время реальной эксплуатации, возможны сбои и отключения. При проектировании цифровых сервисов важно учитывать возможность частичного отключения отдельных модулей системы. На данном этапе, мы поочередно моделируем отключение каждого из микросервисов и описываем поведение остальных микросервисов. Если отключение одного из микросервисов невозможно заменить или перенаправить данные, то для такого микросервиса требуется обеспечить высокую степень отказоустойчивости и дублирование.

Шаг 6

«Проработать метрики и аналитику для мониторинга работы микросервисов». Для оценки потребления ресурсов, наличия ошибок и другой информации об итоговом сервисе, необходимо проработать полную метрику по каждому из микросервисов. Описание данной метрики, должно включать в себя, как показатели этой метрики, так и их крайние и нормальные значения. За основу этого этапа, взяты подходы SLI и SLO. [9]

Для проверки работы алгоритма, предлагается смоделировать работу цифрового сервиса сбора и хранения данных. Сервисы сбора и хранения, отвечают, за сбор производственных данных с локальных систем автоматизации, обработку потоков и пакетов данных, хранение временных рядов, событий и неструктурированных данных.

Для этого выполним пошагово алгоритм, последовательно с небольшим количеством итераций и рассмотрим итоговую модель сервиса и сравним ее с реализацией одного из существующих.

Шаг 1. Необходим сервис, который будет отвечать за сбор входящей информации с производственных линий, а также сохранять полученный данные в реляционную БД.

Шаг 2. Итерация 1. Сервисы: сбор данных, хранение данных.

Шаг 2. Итерация 2. Сервисы: сбор данных сборочных линий, сбор данных линий станков с ЧПУ, сбор данных

упаковочных линий, хранение входных данных линии, хранение выходных данных линии.

Для примера 2-ух итераций достаточно, но для более качественной модели цифрового сервиса количество итераций может быть больше.

Шаг 3. Микросервисы сбора данных сборочных линий, сбора данных линий станков с ЧПУ, сбора данных упаковочных линий:

Пользователь аналитик: умеет читать таблицы. Будет наблюдать за данными, проходящими через сервис для выявления узких мест

Пользователь тех. специалист: умеет читать таблицы, пользоваться консолью для включения/выключения сервиса и его настройки, настраивать интеграция сервиса и производственного оборудования. Технический специалист отвечает за обслуживание сервиса

Микросервисы хранения входных данных линии, хранения выходных данных линии.

Пользователь аналитик: умеет читать таблицы, работать с файлами для выгрузки таблиц для других систем. Отвечает за анализ накопленных данных, а также их выгрузку и обработку в других системах.

Пользователь технический специалист: умеет читать таблицы, пользоваться консолью для включения/выключения сервиса и его настройки, интеграцию загрузки данных из других микросервисов. Тех. Специалист отвечает за обслуживание сервиса

Шаг 4. Единым форматом обмена между сервиса будут выступать csv таблицы с данными.

Шаг 5. В случае частичного отключения микросервиса сбора данных, в сервисах хранения данных должна включаться возможность ручного ввода данных. В случае если перестает работать один из сервисов хранения, данные начинают получать второй сервис хранения до тех пор, пока не будет восстановлена работа первого.

Шаг 6. Все сервисы должны быть подключены к сервисам отслеживания количества входящих и исходящих записей сервисов. В случае если сервис не принимает или не отдает записи, это будет сигнализировать об ошибке.

Заключение

Рассмотренный алгоритм, позволил нам, смоделировать структуру технического сервиса, а также пользовательские возможности, требуемые для использования моделируемого сервиса. В итоге, получена полная мо-

дель, в которой присутствуют требования к конечному сервису и становится возможным оценить трудоемкость разработки, за счет детальной проработки его составных частей.

Если рассмотреть подробнее, мы имеем следующее:

- Список реализуемых микросервисов;
- Структуру взаимодействия микросервисов;
- Список предполагаемых ролей пользователей;
- Схему отказоустойчивости в случае сбоев;

— Предполагаемую метрику сервиса.

Таким образом, используя алгоритм, предложенный ранее, мы покрываем все аспекты моделирования цифрового сервиса, описанные выше.

Можно сделать вывод, что предложенный алгоритм, является полным, включающим в себя все этапы моделирования и в то же время, универсальным, для различных цифровых сервисов.

ЛИТЕРАТУРА

1. Кияков Д.А., Цифровые сервисы в машиностроении — Сборник научных трудов XXIII Международной научно-практической конференции НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ, ООО «1С-Пабблишинг» Москва, том 1, тезисы, 2023 г.
2. Ковригин Е.А., Интеграция современных цифровых технологий в систему менеджмента качества высокотехнологичных предприятия, Автореферат диссертации на соискание ученой степени кандидата технических наук, 2020 г.
3. Прохоров А., Коник Л. Цифровая трансформация. Анализ, тренды, мировой опыт. М.: Издательские решения, 2018. 460 с.
4. Цифровая трансформация: ожидания и реальность: докл. к XXIII Ясинской (Апрельской) междунар. науч. конф. по проблемам развития экономики и общества, Москва, 2022 г. [Текст] / Г.И. Абдрахманова, С.А. Васильковский, К.О. Вишневецкий, М.А. Гершман, Л.М. Гохберг и др.; рук. авт. кол. П.Б. Рудник; Нац. исслед. ун-т «Высшая школа экономики». — М.: Изд. дом Высшей школы экономики, 2022. — 221 с. — ISBN 978-5-7598—2658-3 (в обл.). — ISBN 978-5-7598-2468-8 (e-book)
5. Блог компании AvitoTech. Платформа как сервис в Авито: как это устроено. [Электронный ресурс]. URL: <https://habr.com/ru/companies/avito/articles/527400/> (Дата обращения: 01.05.2023)
6. Чистая архитектура на PHP. Как её измерять и контролировать? [Электронный ресурс]. URL: <https://habr.com/ru/articles/504590/> (Дата обращения: 04.04.2023)
7. Метрики для улучшения архитектуры ПО [Электронный ресурс]. URL: <https://medium.com/nuances-of-programming/метрики-для-улучшения-архитектуры-по-e27daa3e8634> (Дата обращения: 05.04.2023)
8. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. — СПб.: Питер, 2021. — 352 с.: ил. — (Серия «Библиотека программиста»).
9. Блог компании AvitoTech. Как измерить надёжность вашего приложения [Электронный ресурс]. URL: <https://habr.com/ru/companies/avito/articles/742960/> (Дата обращения: 13.09.2023)