

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего  
образования «Самарский национальный исследовательский университет имени академика  
С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики

Кафедра информационных систем и технологий

## ОТЧЕТ ПО ПРАКТИКЕ

Вид практики производственная  
(учебная, производственная)

Тип практики научно-исследовательская работа  
(в соответствии с ОПОП ВО)

Сроки прохождения практики: с 01.09.2022 по 29.12.2022  
(в соответствии с календарным учебным графиком)

по направлению подготовки 09.03.01 Информатика и вычислительная техника  
(уровень бакалавриата)  
направленность (профиль) «Информационные системы»

Обучающийся группы № 6304-090301D И.И. Алеев

Руководитель практики,  
доцент кафедры информационных систем и технологий В.С. Сивков

Дата сдачи 29.12.2022  
Дата защиты 29.12.2022

Оценка \_\_\_\_\_

Самара 2022

## **СОДЕРЖАНИЕ**

1. Задание(я) для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований).
2. Описательная часть.
3. Список использованных источников.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение  
высшего образования «Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики

Кафедра информационных систем и технологий

**Задание(я) для выполнения определенных видов работ, связанных с  
будущей профессиональной деятельностью (сбор и анализ данных и  
материалов, проведение исследований)**

Обучающемуся Алееву Ибрагиму Ильясовичу  
группы 6304-090301D

Направление на практику оформлено приказом по университету  
от 29.08.2022 г. № 305-ПР  
на кафедру информационных систем и технологий  
(наименование профильной организации или структурного подразделения университета)

Тема НИР: «Разработка устройства интернета вещей на базе современных  
систем реального времени»

Планируемые результаты освоения образовательной программы (компетенции)	Планируемые результаты практики	Содержание задания
ПК-6 Способен осуществлять разработку требований и проектирование программного обеспечения ПК 6.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие	Знать: технологии разработки технических спецификаций программных компонентов и их взаимодействия. Уметь: обосновывать выбор технологии разработки технических спецификаций программных компонентов. Владеть: навыками разработки технических спецификаций программных компонентов.	Провести анализ имеющихся технологий разработки технических спецификаций программных компонентов и их взаимодействия.  Провести анализ методологий разработки информационно-логических

		<p>проектов программного обеспечения.</p> <p>Сделать обоснование выбора используемых методологий и технологий для информационной системы по разработке и функционированию устройства интернета вещей.</p>
--	--	---

Дата выдачи задания 01.09.2022.

Срок представления на кафедру отчета о практике 29.12.2022.

Руководитель практики,

доцент кафедры ИСТ \_\_\_\_\_

В.С. Сивков

(подпись)

Задание принял к исполнению

обучающийся группы № 6304-090301D \_\_\_\_\_

И.И. Алеев

(подпись)

## **Описательная часть**

### **Анализ имеющихся технологий разработки технических спецификаций программных компонентов и их взаимодействия.**

Система реального времени (СРВ) – система, которая должна реагировать на события во внешней, по отношению к системе, среде или воздействовать на среду в рамках требуемых временных ограничений. Оксфордский словарь английского языка говорит об СРВ как о системе, для которой важно время получения результата. Другими словами, обработка информации системой должна производиться за определённый конечный период времени, чтобы поддерживать постоянное и своевременное взаимодействие со средой. Естественным, что масштаб времени контролирующей системы и контролируемой ею среды должен совпадать.

Интернет вещей – концепция сети передачи данных между физическими объектами («вещами»), оснащёнными встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой. Предполагается, что организация таких сетей способна перестроить экономические и общественные процессы, исключить из части действий и операций необходимость участия человека.

Существует множество устройств реального времени, которые способны функционировать как устройства интернета вещей. В данный момент на рынке представлены микроконтроллеры которые были специально разработаны для работы с системами умного дома, что показывается в виде поддержки различных протоколов обмена «из коробки», предоставление производителем программного обеспечения и инструкций по интеграции своего устройства в уже готовую систему. Примером готовой платформы является микроконтроллер от компании ST серии STM32L4 – B-L475E-IOT01A, которое имеет возможность передачи данных с помощью технологий WI-FI, Bluetooth, NFC, радио, а также множество разнообразных датчиков.

Для того, чтобы информационная система была способна получать команды от пользователя существует два основных варианта архитектуры:

Централизованная ИС – множество устройств являются дочерними по отношению к какому-либо локальному устройству, которое считывает, хранит, обрабатывает данные полученные с узлов, а также управляет ими.

Децентрализованная ИС – множество устройств являются равноценными по отношению друг к другу, управление осуществляется с помощью удалённых сервисов.

	Централизованная ИС	Децентрализованная ИС
Достоинства	<ol style="list-style-type: none"> <li>1. Простота в проектировании</li> <li>2. Детерминированный порядок обработки</li> <li>3. Возможна отладка</li> </ol>	<ol style="list-style-type: none"> <li>1. Популярность сервисов</li> <li>2. Возможно локальное размещение</li> <li>3. Стабильность не зависит от состояния узлов</li> </ol>
Недостатки	<ol style="list-style-type: none"> <li>1. Стабильность системы всецело зависит от центрального узла</li> <li>2. Высокая нагрузка на канал центрального узла</li> <li>3. Необходимость отдельного размещения, что влечёт затраты</li> </ol>	<ol style="list-style-type: none"> <li>1. Отладка возможна только при локальном размещении</li> <li>2. Недетерминированный порядок обработки</li> </ol>

В обоих вариантах возможно использование различных протоколов и технологий, наиболее распространёнными являются варианты с использованием беспроводных сетей, так как они наиболее удобны при установке. Наиболее популярными протоколами являются MQTT, ZigBee и Wi-Fi.

1. ZigBee (IEEE 802.15.4) – технология созданная для сетей, в которых функционируют малопроизводительные устройства с ограниченным энергопотреблением, для таких целей как домашняя автоматизация,

общий сбор данных с медицинских датчиков и подобных, где в малых проектах необходима беспроводная связь.

2. MQTT (Message Queue Telemetry Transport) – открытый протокол обмена данными созданный для работы в условиях с ограниченной пропускной способностью канала и ограниченным количеством кода.
3. Wi-Fi – технология беспроводной локальной сети с устройствами на основе стандартов IEEE 802.11, то есть в беспроводной локальной сетевой зоне частотных диапазонов 0,9; 2,4; 3,6; 5 и 60 ГГц.

Так как протокол MQTT высокоуровневый и основывается на протоколе TCP/IP, то в качестве аппаратного модуля можно использовать, как модули ZigBee, так и Wi-Fi.

	ZigBee	Wi-Fi
Достоинства	<ol style="list-style-type: none"><li>1. Распространённость в готовых решениях</li><li>2. Множество готовых модулей</li><li>3. Высокая энергоэффективность</li></ol>	<ol style="list-style-type: none"><li>1. Наиболее простая и популярная технология</li><li>2. Широкий набор модулей</li><li>3. Механизм настройки известен и прост.</li><li>4. Низкая цена</li><li>5. Чаще всего достаточно иметь только устройство для работы, так как в рабочих условиях уже присутствует роутер или любое другое устройство с поддержкой режима «точка доступа Wi-Fi»</li></ol>

Недостатки	<ol style="list-style-type: none"> <li>1. Цена несколько выше, чем у модулей Wi-Fi</li> <li>2. Менее распространён, чем Wi-Fi</li> <li>3. Для взаимодействия будет недостаточно иметь модуль на готовом устройстве</li> </ol>	<ol style="list-style-type: none"> <li>1. Энергопотребление несколько выше, чем у ZigBee</li> <li>2. Необходимо настроить статический адрес для локальной сети</li> </ol>
------------	---	---

В результате для выполнения проекта была выбрана следующая конфигурация:

- Централизованная информационная система
- Беспроводная технология Wi-Fi



## **Анализ методологий разработки информационно-логических проектов программного обеспечения.**

От выбора методологии будет зависеть то, как разные этапы жизненного цикла будут связаны между собой и в какой последовательности реализованы. Чтобы правильно выбрать модель, нужно понимать плюсы и минусы каждой из них и суть своего проекта.

### **1. Waterfall**

Эта модель предполагает постепенное перемещение по этапам жизненного цикла. Сначала проводится анализ и составление задачи, затем проектирование, затем программирование и так далее. Каждый следующий этап стартует только тогда, когда закончен предыдущий. В этом кроется главное преимущество «водопада» и главный недостаток.

С одной стороны, проектом легко управлять, есть четкая последовательность действий, сроки выполнения и бюджет известен заранее. С другой – проекты с такой моделью не терпят правок, требующих возвращения к предыдущим этапам, а результат заказчик видит только на завершающих этапах разработки, когда приложение почти готово.

Достоинства	Недостатки
<ol style="list-style-type: none"><li>1. Тестирование могут проводить люди с более низкой квалификацией</li><li>2. Низкая вероятность ошибок в небольших проектах.</li><li>3. Стоимость и сроки известны на начальном этапе</li><li>4. Простое управление разработкой при наличии четко сформулированной документации.</li></ol>	<ol style="list-style-type: none"><li>1. Тестирование происходит на последних этапах.</li><li>2. Чем масштабнее проект, тем большая вероятность критических ошибок, исправление которых потребует значительного увеличения бюджета.</li><li>3. Заказчик видит готовый продукт лишь в конце разработки.</li><li>4. Написание и согласование подробной документации также может вызвать множество задержек.</li></ol>

## 2. TDD ( Test-driven development) или же V-model

TDD модель является модифицированной версией «водопада». V стоит в названии от двух главных принципов данной методологии — validation и verification. По сути, здесь процессы происходят друг за другом, однако на каждом этапе присутствует элемент тестирования. Продукт подвергается тщательным проверкам уже на начальных этапах разработки. Тестирование является основополагающим элементом всего процесса.

Достоинства	Недостатки
<ol style="list-style-type: none"><li>1. Тестирование проходит на всех этапах разработки.</li><li>2. Вероятность ошибок сводится к минимуму.</li></ol>	<ol style="list-style-type: none"><li>1. Требуется высокий уровень квалификации тестировщиков и/или их высокая занятость.</li><li>2. Если ошибка все же была допущена, то вернуться к предыдущему этапу будет даже дороже, чем при каскадной модели.</li></ol>

## 3. Инкрементная модель

Инкрементная модель в целом следует той же структуре, что и каскадная, однако, как можно понять из названия, все этапы проходят несколько раз в течение жизненного цикла ПО. Получается своеобразный «мультиводопад».

Достоинства	Недостатки
<ol style="list-style-type: none"><li>1. Есть возможность раннего выхода на рынок, чтобы посмотреть реакцию пользователей.</li><li>2. Базовая версия ПО стоит дешевле. Модули можно доделывать по мере появления денег, либо не делать вовсе за ненадобностью. Самые рискованные идеи можно отложить на потом.</li><li>3. Исправление ошибок обходится дешевле.</li></ol>	<ol style="list-style-type: none"><li>1. Требования к проекту на каждом этапе должны быть четко определены и понятны. Необходим хороший менеджмент.</li><li>2. Приложение может выйти слишком «сырым» и не дожить до появления всех функций.</li></ol>

## 4. Быстрая разработка

RAD Model (Rapid Application Development model) — это модель быстрой разработки приложений. Это своего рода ответвление инкрементной модели, так как процесс создания ПО происходит таким же образом с единственным

исключением — над проектом работает сразу несколько команд. То есть в один момент времени параллельно существует несколько мини-проектов в одном большом проекте, которые интегрируются в рабочий прототип по мере готовности.

Достоинства	Недостатки
<ol style="list-style-type: none"> <li>1. Есть возможность раннего выхода на рынок, чтобы посмотреть реакцию пользователей.</li> <li>2. Базовая версия ПО стоит дешевле. Модули можно доделывать по мере появления денег, либо не делать вовсе за ненадобностью. Самые рискованные идеи можно отложить на потом.</li> <li>3. Исправление ошибок обходится дешевле.</li> </ol>	<ol style="list-style-type: none"> <li>1. Требования к проекту на каждом этапе должны быть четко определены и понятны. Необходим хороший менеджмент.</li> <li>2. Приложение может выйти слишком «сырым» и не дожить до появления всех функций.</li> </ol>

### 5. Итеративная модель

По сути, итеративная модель — это также разновидность инкрементной модели, которая, однако, лучше показывает себя в больших проектах, где конечная цель заранее не определена либо планируется применение каких-либо инновационных подходов.

Достоинства	Недостатки
<ol style="list-style-type: none"> <li>1. Есть возможность раннего выхода на рынок, чтобы посмотреть реакцию пользователей.</li> <li>2. Возможность запустить проект, когда конечная цель до конца не определена.</li> <li>3. Добавлять новые функции и менять направление проекта можно с каждой новой итерацией в зависимости от бюджета.</li> </ol>	<ol style="list-style-type: none"> <li>1. Добавление заранее не оговоренных функций может привести к необходимости полного переделывания целых кусков проекта.</li> <li>2. Отсутствие фиксированного бюджета и сроков реализации.</li> <li>3. Приложение может выйти слишком «сырым» и не дожить до того, как станет функционально соответствовать задумке</li> </ol>

### 6. Спиральная модель

Эта модель — также «родственница» инкрементной и итеративной моделей, но с большим упором на анализ рисков и оценку выгоды проекта. Разработка идет по такому же принципу: реализация части проекта и вывод продукта на рынок поэтапно. Единственное отличие — разработка каждой новой версии

продукта начинается только в том случае, если заказчик уверен в ее необходимости, востребованности и потенциальной выгоде.

Все перечисленные методологии обладают своими преимуществами и недостатками, а также необходимыми условиями для применения, поэтому для применения одной из них необходимо составить требования к проекту, в частности указать формат поддержки, ограниченность в сроках разработки и эксплуатации, расширяемость набора функций.

### **Обоснование выбора используемых методологий и технологий.**

Среди перечисленных методологий для выполнения проекта была выбрана разработка через тестирования или TDD. Главной причиной данного выбора стало то, что в проекте задействуются не только абстракции и программное обеспечение, но и аппаратная платформа, поэтому малейшая ошибка в проектировании способна вывести аппарат из строя, например, при неправильном задании частоты тактового сигнала устройство будет неспособно с достаточной скоростью ответить на входящий запрос, в результате будут отсутствовать и исходящие данные о состоянии, и входящие не смогут быть корректно интерпретированы.

## **ЗАКЛЮЧЕНИЕ**

В процессе выполнения научно-исследовательской работы был освоен индикатор ПК-6.2 компетенции ПК-6, и решены все поставленные задачи:

- был проведен анализ имеющихся технологий разработки технических спецификаций программных компонентов и их взаимодействия;
- был проведен анализ методологий разработки информационно-логических проектов программного обеспечения;
- было сделано обоснование выбора используемых методологий и технологий для разработки информационной системы по разработке и функционированию устройства интернета вещей.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. [https://ru.wikipedia.org/wiki/Операционная\\_система\\_реального\\_времени](https://ru.wikipedia.org/wiki/Операционная_система_реального_времени)
2. [https://ru.wikipedia.org/wiki/Информационная\\_система](https://ru.wikipedia.org/wiki/Информационная_система)
3. <https://highload.today/metodologii-razrabotki/>