

ото дня. Однако сеть в Беларуси все еще находится в эпохе 4G, и сигнал не всегда стабилен, что сказывается на пространстве развития белорусской цифровой экономики. Многие рабочие места, которые могут быть заменены умными машинами, до сих пор не заменены, что приводит к трате ненужных человеческих ресурсов.

2. Безопасность сети не идеальна. В течение долгого времени система сетевой безопасности Беларуси не была идеальной, и существует множество проблем с безопасностью, например, таких, как мошенничество, незаконная экономическая деятельность через Интернет, некоторые финансовые учреждения сталкиваются с риском кражи данных. Хотя многие финансовые учреждения приняли превентивные меры, лазейки по-прежнему неизбежны, некоторые факторы вышли из-под контроля, и безопасность данных и информации не может быть гарантирована. Несовершенная система сетевой безопасности затрудняет эффективную защиту безопасности и целостности данных и информации.

Меры, которые будут способствовать развитию цифровой экономики в Беларуси:

1. Дальнейшее развитие цифровых технологий. Чтобы создать хорошую платформу для развития цифровой экономики, необходимо усилить стабильность построения сетевых сигналов и развивать сеть 5G, идущую в ногу со временем. В то же время необходимо максимально внедрить существующие цифровые технологии во все рабочие места и заменить некоторые простые ручные задачи интеллектуальными машинами, чтобы сэкономить больше ресурсов.

2. Улучшить правовую систему. Цифровая экономика относительно зависима от данных и информации, и защита данных и информации должна быть основным объектом надзора за экономическим законодательством. Чтобы лучше направлять развитие рынка цифровой экономики, необходимо проанализировать характеристики цифровой экономики и сформулировать специальные законы для ее регулирования, принять принцип пропорциональности, ввести динамический анализ и обратить внимание на процесс конкуренции участников рынка. Согласно закону перехода от необработанных данных к большим данным, следует оценивать потенциальное влияние конкуренции и правовых рисков, совершенствовать экономическую правовую систему и поддерживать порядок на рынке цифровой экономики.

3. Установить кредитную систему. Следуя принципу честности и надежности, необходимо внедрять механизм перспективного прогнозирования, классифицировать кредитоспособность, оценивать и определять кредитный рейтинг обеих сторон в рыночной сделке цифровой экономики. Следует создавать онлайн-профили кредитоспособности в цифровом виде, проводить кредитные рейтинги и отмечать граждан с низким кредитным рейтингом, чтобы снизить вероятность нарушений информационной безопасности у их источника.

У Беларуси есть все возможности для создания условий развития цифровой экономики и дальнейшего прогресса.

УДК 004.413

## **РАЗРАБОТКА ПРИЛОЖЕНИЯ ОТ ПРОЕКТИРОВАНИЯ ДО СБОРКИ С ПРИМЕНЕНИЕМ ИНСТРУМЕНТАРИЯ СЕРВИСА GITHUB**

*Шинкевич Д. О., Иванова М. Д., Давидовская М. И.*

*Белорусский государственный университет*

*e-mail: fpm.shinkevi@bsu.by*

**Summary.** *This article is devoted to the design of the application architecture, project documentation, team development of the application, build and continuous integration using the GitHub cloud service. In particular, the features and capabilities of the service are analyzed. The stages of the project life circle from the application design and creation of documentation to its development are shown.*

Для управления разработкой программных проектов применяются различные инструменты. Одним из них является облачный сервис GitHub. «GitHub – это платформа для

размещения кода для контроля версий и совместной работы» [1; 4, с. 33]. Кроме контроля версий программного кода GitHub предоставляет возможности для управления жизненным циклом приложения. Сервис содержит инструменты для управления кодом приложения, менеджмента и документирования проекта, непрерывной интеграции и доставки. Как следствие, разработчикам не нужно переключаться между множеством узкоспециализированных сервисов, что ускоряет и упрощает процесс разработки [4, с. 33]. Построение этапа разработки учебных и исследовательских проектов с использованием инструментария GitHub позволяет получить практические навыки командной разработки.

Выделим основные инструменты, которые предоставляет веб-сервис GitHub:

1. Сервис GitHub – это репозиторий и онлайн-сервис для командной разработки и контроля версий, который позволяет отслеживать любые изменения в коде и их автора, возвращаться к предыдущим версиям проекта, а также организовывать совместную работу членов команды, территориально удаленных друг от друга [4, с. 33].

2. Инструмент Projects предоставляет функционал для управления проектами на основе концепции Kanban, которая относится к гибким методологиям разработки (Agile development). Сервис Projects помогает отслеживать задачи, отображая их в виде карточек на Kanban-доске или списком. Задачи могут быть интегрированы с работами (issues) и запросами на публикацию [4, с. 33].

3. Для выполнения автоматической сборки проекта применяется инструмент GitHub Actions, который относится к классу программных продуктов для непрерывной интеграции и доставки. Его основой являются рабочие процессы workflows. Рабочий процесс описывается конфигурационным файлом с использованием языка разметки YAML. Инструкции, указанные в файле рабочего процесса, запускают по событию автоматическую сборку и тестирование. Таким событием является внесение изменений в репозиторий. В итоге команда разработки выполняет автоматическую сборку проекта, который тестируется и выполняется в окружении, отчужденном от окружения разработки.

4. Качественная документация – важный компонент любого проекта разработки программного обеспечения (ПО). Для документирования проекта сервис GitHub предлагает несколько инструментов. Краткая характеристика проекта добавляется в файл README. Детальная и многостраничная документация проекта формируется с помощью GitHub Pages и wiki. Каждый из инструментов документирования использует язык разметки Markdown. В отличие от инструмента wiki, который относится к функционалу репозитория, GitHub Pages является внешним сервисом по отношению к сервису GitHub и представляет собой небольшой веб-сайт, индексируемый поисковыми системами.

Для исследования возможностей GitHub было разработано консольное приложение на языке C++ для управления процессами автопарка, предоставляющее пользовательский интерфейс для выполнения CRUD-операций (**create** – создание, **read** – чтение, **update** – модификация, **delete** – удаление) над данными о водителях, автомобилях и перевозках. Функционал приложения обеспечивает формирование различных отчетов на основе данных, хранимых в базе данных (БД) [4, с. 33].

В проекте применяется компактная однофайловая встраиваемая база данных SQLite. Для подключения к базе данных автопарка используется API языка C++ для SQLite. В приложении реализован функционал разграничения прав доступа. Пользователи разделены на две категории с различными правами доступа: обычные пользователи и администраторы.

Сегодня многие приложения – результат работы команды специалистов, вклад каждого из которых управляется системой контроля версий. Для получения опыта совместной разработки проект был выполнен в команде из двух человек. С помощью GitHub работа над проектом была организована следующим образом: каждый участник добавляет свою часть изменений в отдельной ветке, а для слияния веток используются запросы на публикацию [4, с. 34].

Этап проектирования проекта включал разработку спецификации проекта с использованием унифицированного языка моделирования UML. Функциональные требования были детализированы с помощью диаграмм вариантов использования, а структура ПО – с помощью диаграммы файлов на основе диаграммы классов [3].

Библиотека модульного тестирования Google C++ Testing Framework (**gtest**) использовалась для разработки тестов, позволяющих эффективно проверить работу отдельных частей программы.

В рабочий процесс workflow проекта в GitHub Actions включены задания на автоматический запуск тестов, что позволило проверять приложение во время его разработки и моментально получать результаты. В результате получили возможность выявлять изменения в коде, приводящие к ошибкам. Конфигурация рабочего процесса включает события для сборки проекта, этап настройки среды для сборки на основе операционной системы Ubuntu, этап установки пакета **gtest** и необходимых библиотек и этап запуска тестов [2].

Дополнительным инструментом проверки качества приложения являются решения для исследования покрытия исходного кода тестами. В рассматриваемом проекте использовалась утилита **gcov**, анализирующая протестированные участки кода. Для визуализации полученных данных применялся интерфейс **lscov**, который собирает сведения о покрытии кода в комплект HTML файлов [3; 4, с. 37].

Кроме автоматического тестирования приложения была разработана конфигурация автоматической сборки в виде файла Makefile. Файл содержит целевые связки, описывающие этапы сборки. Сборка выполняется с помощью утилиты **make**. Во внешней репозитории в GitHub Actions рабочий процесс workflow содержит команды для выполнения сборки.

На основе проведенного исследования возможностей GitHub можем сделать вывод, что данный веб-сервис предоставляет разнообразные инструменты для совместной разработки приложений. Он отлично подходит для всех стадий создания приложения, начиная от разработки и заканчивая сборкой и тестированием проекта. Практические результаты подтверждаются разработанным проектом.

#### **Список использованных источников**

1. GitHub Docs. 2022. – Electronic resource. URL: <https://docs.github.com/en>. – (Date of access: 29.07.2022). – Title from screen.
2. Репозиторий проекта приложения «Автопарк». 2022 г. – Электрон. данные. URL: <https://github.com/fpmi-tp2022/labrabota5t1-scorpion>. – Дата обращения: 07.08.2022. – Заглавие с экрана.
3. Сайт проекта «Автопарк». 2022 г. – Электрон. данные. URL: <https://fpmi-tp2022.github.io/labrabota5t1-scorpion/>. – Дата обращения: 07.08.2022. – Заглавие с экрана.
4. Шинкевич Д. О., Иванова М. Д., Давидовская М. И. Проектирование, реализация, тестирование и сборка приложения с применением инструментария сервиса GitHub // Научные исследования XXI века: сетевое издание. – 2022. – № 4 (18). – С. 32–37. – Электрон. данные. – URL: <http://scientific-research.ru/files/JOURNAL--4--18-.pdf>. – Дата обращения: 30.09.2022.