

СОСТАВ И СТРУКТУРА ПРОГРАММНОГО КОМПЛЕКСА РАСЧЕТА СЛОЖНОСТИ И ТРУДОЕМКОСТИ IT-ПРОЕКТОВ

Ерохова Д. И., студент

Вайнилович Ю. В., к.т.н., доцент

Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет» г. Могилев, Беларусь

Аннотация. В статье описывается программная система для оценки сложности и размера программных продуктов на основе методов функциональных точек и модели COSOMO II, обращается внимание на проблемы, связанные с отсутствием единых стандартов и метрик, которые влияют на процессы разработки и конечный продукт.

Ключевые слова: метод функциональных точек, модель COSOMO II, программная система, сложность программного продукта, размер программного продукта

В разработке программного обеспечения на этапе оценки сложности и размера программных продуктов существует ряд проблем, которые существенно влияют не только на процессы разработки, но и на конечный продукт. Отсутствие единых стандартов и метрик создает трудности в сопоставлении результатов оценок сложности различных проектов, а существование субъективности оценки, основанной на опыте разработчика, может привести к недостаточной точности результата оценивания. Что в свою очередь, может привести к неверному планированию сроков реализации IT-проектов, неверному планированию потребности в ресурсах, неверному подбору состава и структуры проектной команды.

В связи с этим становится актуальной задача создания специализированного программного обеспечения, способного обеспечить объективную оценку сложности создаваемых программных продуктов.

Одним из решений является разработка программной системы в виде web-приложения, предназначенной для удобного и более точного подсчета сложности и размера программного продукта на основе требований к нему [1].

Контекстная диаграмма программной системы представлена на рисунке 1.



Рисунок 1 – Контекстная диаграмма системы расчета сложности и размера программных продуктов

В основе решения лежит комбинация двух ключевых методов оценки – метод функциональных точек и модель COSOMO II [2, 3], что позволяет учесть как размер кода, так и функциональность решения, а также множество факторов влияющих на сложность. За счет интеграции методов функциональных точек и COSOMO II обеспечивается более всесторонний анализ, учитываются факторы, которые могут влиять на сложность проекта, такие как опыт команды, сложность платформы разработки и разработка для повторного использования.

Предложенная программная система для оценки сложности и размера программных продуктов предоставляет широкий спектр функциональных возможностей:

- автоматизированный расчет функциональных точек на основе введенных требований, что обеспечивает количественную оценку сложности;
- поддержка предварительной оценки на начальных этапах проекта и детальной оценки после проработки архитектуры. Система учитывает различные факторы, такие как квалификация персонала, опыт, сложность и надежность продукта, оборудование и другие, предоставляя точные прогнозы времени на разработку;
- совмещение результатов методов функциональных точек и COSOMO II для получения более надежных оценок;
- использование анкет проектов для обеих методик, позволяющие провести более глубокий и всесторонний анализ требований проекта, а также структурировать и конкретизировать функциональные и нефункциональные характеристики проекта, что дает возможность учесть дополнительные аспекты, влияющие на сложность;

- предоставление отчетов результатов оценивания сложности и размера программного продукта для лучшего понимания пользователем;
- расширяемость системы, что позволит добавлять новые методы и метрики в будущем;
- экспорт результатов оценивания для последующего использования в проектной документации, отчетах или интеграции с другими инструментами управления проектами;
- автоматизация процесса оценки, что минимизирует влияние субъективности и человеческого фактора при расчетах, обеспечивая более точные результаты;
- учет особенностей различных языков программирования при оценивании количества строк кода, что улучшает точность оценок;
- интеграция справочной информации и обучающих материалов по методам оценки, обеспечивающая простое, быстрое и комфортное взаимодействие пользователя и системы.

Для хранения данных в программной системе выбран реляционный тип баз данных, так как в проекте данные структурированы и связи между ними играют важную роль.

В качестве системы управления базами данными выбрана система PostgreSQL – объектно-реляционная система управления базами данных с открытым исходным кодом. Выбор обусловлен несколькими факторами. В первую очередь, PostgreSQL обладает отличной производительностью и надежностью, что является критически важным для приложения, ориентированного на эффективное взаимодействие с данными [5].

В качестве серверной технологии был выбран NodeJS — асинхронная управляемая событиями исполнительная платформа JavaScript с мощной, но компактной стандартной библиотекой [6]. Этот выбор обусловлен высокой производительностью и эффективностью NodeJS в обработке большого количества одновременных соединений. Благодаря асинхронному и событийно-ориентированному характеру NodeJS, можно достичь высокой отзывчивости сервера. В сочетании с Express.js, фреймворком, обеспечивающим набор возможностей для построения одно- и многостраничных и гибридных веб-приложений на Node.js [7], обеспечивается быстрое развертывание серверной части проекта. Express.js предоставляет удобный механизм для обработки запросов, маршрутизации и поддержки средств безопасности.

Дополнительно, использование JavaScript как языка программирования для серверной и клиентской стороны обеспечивает единообразие кода и более удобную поддержку.

Для разработки клиентской части приложения использовалась библиотека React – React предоставляет декларативный подход к созданию пользовательских интерфейсов, что облегчает понимание и поддержку кода.

Для обеспечения эффективного взаимодействия между клиентской и серверной частями использовалась библиотека axios, позволяющая легко выполнять HTTP-запросы, отправлять данные на сервер и обрабатывать ответы.

Axios – мощный инструмент для обработки запросов и ответов на различных этапах их жизненного цикла.

Важным аспектом разработки приложения является обеспечение безопасности пользовательских данных, в частности, паролей. Для хеширования паролей выбрана библиотека `bcryptjs`, предоставляющая надежные алгоритмы хеширования, основанные на `bcrypt` и предотвращающая распространенные угрозы, таких как атаки по словарю и радужные таблицы, обеспечивая эффективную защиту учетных записей пользователей.

Для обеспечения безопасной аутентификации и авторизации пользователей в приложении применяется библиотека `jsonwebtoken`.

В заключение, система оценки сложности программных продуктов, объединяя методы функциональных точек и COCOMO II, предоставляет точные, объективные результаты и эффективно решает проблемы, связанные с оценкой на ранних этапах разработки. Гибкие настройки, поддержка различных языков программирования и уникальное сочетание методик делают ее мощным инструментом для более глубокого понимания требований проекта и оценки сложности и размера программного продукта[4].

Литература

1 Ерохова Д. И., Вайнилович Ю. В. Об актуальности разработки приложения для оценки сложности и размера программных продуктов // Энергетика, информатика, инновации – 2023 (Электроэнергетика, электротехника и теплоэнергетика, математическое моделирование и информационные технологии в производстве). Сб. трудов XIII Межд. науч.-техн. конф. В 3 т. Т 1. – 2023. – С 188-190.

2 Будыльский, А. В. Методы функциональных точек / А. В. Будыльский // Информационные технологии. Радиоэлектроника. Телекоммуникации. – 2012. – № 2-1. – С. 272-278.

3 Boehm B., et al. «Software cost estimation with COCOMO II». Englewood Cliffs, NJ: Prentice–Hall, 2000.

4 Вайнилович, Ю. В. Оценка сложности и трудоемкости разработки IT-проектов при проектно-ориентированном обучении / Ю. В. Вайнилович, Д. И. Ерохова, М. В. Башаримова // Цифровые, компьютерные и информационные технологии в науке и образовании : Сборник статей Межрегиональной научно-практической конференции с международным участием, Брянск, 01–02 ноября 2023 года. – Брянск: Брянский государственный университет имени академика И.Г. Петровского, 2023. – С. 86-88.

5 Джуба, С. Изучаем PostgreSQL 10 / С. Джуба, А. Волков // Пер. с англ. – А. А. Слинкина. – М.: ДМК Пресс, 2019. – 400 с.

6 Янг, А. Node.js в действии / А. Янг, Б. Мек, М. Кантелон // 2-е изд. — СПб.: Питер, 2018. — 432 с.

7 Браун, И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript / И. Браун // СПб.: Питер, 2017. — 336 с.