

УДК 004.8

АВТОМАТИЗАЦИЯ ВЫЯВЛЕНИЯ ОШИБОК ПРИ СБОРКЕ ПРОГРАММНЫХ ПРОЕКТОВ С ПРИМЕНЕНИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Косарев В.Д. (ИТМО)

Научный руководитель – кандидат технических наук, доцент Зудилова Т.В.
(ИТМО)

Введение. Современные практики разработки программного обеспечения характеризуются большой сложностью проектов, множеством внешних библиотек и зависимостей, а также необходимостью частых релизов. На этапе сборки это приводит к появлению различных ошибок конфигурации, конфликтов версий и логических сбоев, которые сложно быстро обнаружить и устранить. Традиционные методы (статический анализ, тестирование, ручной разбор логов) нередко оказываются недостаточными, особенно в ситуациях, где ошибки проявляются лишь при редких сочетаниях условий.

В рамках выполненной научно-исследовательской и практической работы была поставлена цель изучить возможности использования искусственного интеллекта (ИИ) для автоматизации анализа ошибок при сборке проектов. Предполагается, что ИИ-подходы: машинное обучение, большие языковые модели – смогут не только своевременно выявлять сбои, но и формировать структурированные рекомендации по их устранению [1].

Основная часть. Наибольший интерес представляют вопросы сокращения времени и ресурсов, которые необходимы на поиск и диагностику ошибок при сборке, учитывая, что логи могут быть объёмными, а природа сбоев – разнообразной [2]. Традиционные инструменты оказываются недостаточно гибкими, ведь они либо фиксируют «факт» ошибки, либо требуют от разработчика вручную анализировать множество строк вывода.

Для решения этой проблемы был разработан прототип сервиса, написанный на языке Go, который запускает сборку и тестирование, перехватывая все логи. В случае обнаружения ключевых индикаторов сбоя сервис отправляет фрагмент логов к большой языковой модели, прося предоставить объяснение и краткий план устранения проблемы. Такой ответ возвращается в формате JSON, где описаны причина и возможные шаги исправления. Данный формат облегчает последующую интеграцию с таск-трекерами или другими инструментами CI/CD.

В ходе практической реализации была подтверждена универсальность такого подхода: сервис не требует обучения собственной модели, а использует существующий API, что сокращает издержки на настройку. Вместе с тем сам анализ логов с помощью ИИ показывает высокую эффективность для самых разных сценариев: от намеренно «упавшего» теста до синтаксической опечатки или неверно заданной переменной окружения [3]. Во всех случаях сервис выдаёт чёткое и понятное объяснение, позволяя разработчику быстро устранить сбой. Однако качество этих рекомендаций во многом зависит от того, насколько информативны сообщения об ошибке в логах.

Преимуществом разработанного решения является сокращение времени, которое уходит на исследование логов и поиск корневой причины проблем. Вместе с тем есть определённые ограничения: если логи слишком лаконичны или двусмысленны, даже ИИ-модель не всегда способна предоставить точные советы. Кроме того, при использовании внешнего API возможны лимиты на количество обращений, а при развёртывании локальной модели потребуются дополнительные аппаратные ресурсы. Тем не менее, сам прототип подтверждает, что автоматизация анализа логов сборки в реальном времени и генерация рекомендаций существенно повышают эффективность и надёжность процессов разработки.

Выводы. Разработанный сервис, интегрирующийся в сборочный конвейер и обращающийся к большим языковым моделям, позволяет минимизировать ручной труд при поиске и устранении ошибок. Тестовые сценарии, проверившие разные типы сбоев, показали,

что подход действительно снижает время реакции и даёт чёткие рекомендации, упорядоченные в JSON-формате. В перспективе можно углублять систему за счёт предиктивной аналитики и расширенных классификаторов типов ошибок. Таким образом, применение искусственного интеллекта на этапе сборки открывает новые горизонты в сфере DevOps, делая разработку ПО более гибкой и надёжной.

Список использованных источников:

1. Kodezi Blog. AI Code Fixer: Streamlining Code Analysis and Error Resolution [Электронный ресурс]. – URL: <https://blog.kodezi.com/ai-code-fixer-streamlining-code-analysis-and-error-resolution/> (дата обращения: 20.12.2024).
2. ContextQA. AI-driven bug detection: reshaping software quality [Электронный ресурс]. – URL: <https://contextqa.com/ai-driven-bug-detection-reshaping-software-quality/> (дата обращения: 20.12.2024).
3. Habr. AI в разработке микросервисов [Электронный ресурс]. – URL: <https://habr.com/ru/companies/bothub/articles/842816/> (дата обращения: 20.12.2024).