

ИНСТРУМЕНТАЛЬНЫЕ И МАТЕМАТИЧЕСКИЕ МЕТОДЫ В ПРОЦЕССАХ УПРАВЛЕНИЯ

ПРИМЕНЕНИЕ ШКАЛЫ УРОВНЕЙ АВТОМАТИЗАЦИИ ШЕРИДАНА И ВЕПЛАНКА ДЛЯ ОБЪЕКТИВНОЙ ОЦЕНКИ ИНСТРУМЕНТОВ БИЗНЕС-АНАЛИТИКИ

Получено: 29.09.2025

Доработано: 24.11.2025

Принято: 01.12.2025

УДК 65.011.56, 004.5

JEL C88

DOI <https://doi.org/10.26425/2658-3445-2025-8-4-22-34>

Клюев Роман Владимирович

Д-р техн. наук, гл. науч. сотр.; Государственный университет управления, г. Москва, Российская Федерация
ORCID: 0000-0003-3777-7203

E-mail: rv_klyuev@guu.ru

Макаров Владимир Сергеевич

Д-р техн. наук, ст. науч. сотр.; Государственный университет управления, г. Москва, Российская Федерация
ORCID: 0000-0002-4423-5042

E-mail: vs_makarov@guu.ru

Стефановский Дмитрий Владимирович

Канд. техн. наук, зав. каф. информационных систем; Государственный университет управления, г. Москва, Российская Федерация
ORCID: 0000-0002-8261-5951

E-mail: dv_stefanovskiy@guu.ru

Шпонарский Илья Владимирович

Ассист. информационных систем; Государственный университет управления, г. Москва, Российская Федерация
ORCID: 0009-0008-2097-1441

E-mail: iv_shponarskiy@guu.ru

Барина Полина Андреевна

Инженер научной лаборатории «Перспективные информационные технологии» Центра цифровых технологий управления; Государственный университет управления, г. Москва, Российская Федерация
ORCID: 0009-0007-8791-4938

E-mail: pa_barinova@guu.ru

АННОТАЦИЯ

В условиях стремительного роста разнообразия инструментов бизнес-аналитики выбор между визуальными Low-Code платформами и инструментами command line interface, работающими в сочетании с большими языковыми моделями, все чаще носит субъективный характер, что повышает риски возникновения ошибок при внедрении решений. Настоящее исследование предлагает применение шкалы уровней автоматизации Шеридана и Вепланка как объективной метрики для сравнения инструментов бизнес-аналитики с адаптацией методики оценки уровня автоматизации операций для производственных предприятий И.Г. Коралева к задачам построения дашбордов и использования программных конвейеров, взаимодействующих с большими языковыми моделями. Представлены материалы и методы исследования на базе эксперимента, в ходе которого рассмотрены процессы разработки дашбордов и диаграмм процессов на платформах Apache Superset и Draw.io соответственно, а также с применением программного конвейера на базе инструментов JupyterLab, Dash, Plotly, PlantUML с использованием больших языковых моделей. По итогам проведения эксперимента описаны результаты использования количественной оценки уровней автоматизации по шкале Шеридана и Вепланка, проведен анализ эргодичности в части «устойчивости» business intelligence инструментов за пределами идеальных условий. Выводы демонстрируют, что использование шкалы уровней автоматизации Шеридана и Вепланка может способствовать устранению субъективности выбора инструментов бизнес-аналитики и послужить основанием для выработки управленческих решений в сфере информационных технологий.

КЛЮЧЕВЫЕ СЛОВА

Автоматизация, бизнес-аналитика, бизнес-процесс, оценка, инструмент, операция, эргономика, шкала

БЛАГОДАРНОСТИ

Статья выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации, Соглашение № 075-15-2024-542.

ДЛЯ ЦИТИРОВАНИЯ

Клюев Р.В., Макаров В.С., Стефановский Д.В., Шпонарский И.В., Барина П.А. Применение шкалы уровней автоматизации Шеридана и Вепланка для объективной оценки инструментов бизнес-аналитики // E-Management. 2025. Т. 8, № 4. С. 22–34.

© Клюев Р.В., Макаров В.С., Стефановский Д.В., Шпонарский И.В., Барина П.А., 2025.

Статья доступна по лицензии Creative Commons “Attribution” («Атрибуция») 4.0. всемирная (<http://creativecommons.org/licenses/by/4.0/>).



INSTRUMENTAL AND MATHEMATICAL METHODS IN MANAGEMENT PROCESSES

APPLICATION OF THE SHERIDAN AND VERPLANK AUTOMATION LEVEL SCALE FOR AN OBJECTIVE ASSESSMENT OF BUSINESS INTELLIGENCE TOOLS

Received 29.09.2025 Revised 24.11.2025 Accepted 01.12.2025

Roman V. Klyuev

Dr. Sci. (Engr.), Chief Researcher; State University of Management, Moscow, Russian Federation
ORCID: 0000-0003-3777-7203

E-mail: rv_klyuev@guu.ru

Vladimir S. Makarov

Dr. Sci. (Engr.), Senior Researcher; State University of Management, Moscow, Russian Federation
ORCID: 0000-0002-4423-5042

E-mail: vs_makarov@guu.ru

Dmitrij V. Stefanovskij

Cand. Sci. (Engr.), Head of the Information Systems Department; State University of Management, Moscow, Russian Federation
ORCID: 0000-0002-8261-5951

E-mail: dv_stefanovskiy@guu.ru

Ilya V. Shponarskij

Assistant of the Information Systems Department; State University of Management, Moscow, Russian Federation
ORCID: 0009-0008-2097-1441

E-mail: iv_shponarskiy@guu.ru

Polina A. Barinova

Engineer of the Scientific Laboratory "Advanced Information Technologies" of the Center for Digital Management Technologies; State University of Management, Moscow, Russian Federation
ORCID: 0009-0007-8791-4938

E-mail: pa_barinova@guu.ru

ABSTRACT

With the rapid growth in the variety of business intelligence tools, the choice between visual Low-Code platforms and command line interface tools working in combination with large language models is increasingly subjective, which increases the risk of errors when implementing solutions. The study proposes the use of the Sheridan and Verplank Levels of Automation scale as an objective metric for comparing business analytics tools, adapting I.G. Korablev's operation analysis methodology for assessing the level of automation for manufacturing enterprises to the tasks of dashboard development and the use of software pipelines interacting with large language models. The materials and methods of the study have been presented on the basis of an experiment that examined the processes of developing dashboards and process diagrams on the Apache Superset and Draw.io platforms, respectively, as well as using a software pipeline based on JupyterLab, Dash, Plotly, and PlantUML tools with the use of large language models. Based on the experiment results, the results of using quantitative assessment of automation levels on the Sheridan and Verplank scales have been described and the ergodicity in terms of the "stability" of business intelligence tools outside of ideal conditions. The findings demonstrate that the use of the Sheridan and Verplank scale of automation levels can help eliminate subjectivity when choosing business intelligence tools and can become the basis for management decisions in the IT sphere.

KEYWORDS

Automation, business analytics, business process, business intelligence, evaluation, tool, operation, ergonomics, scale

ACKNOWLEDGMENTS

The article was carried out with the financial support of the Ministry of Science and Higher Education of the Russian Federation, Agreement No. 075-15-2024-542.

FOR CITATION

Klyuev R.V., Makarov V.S., Stefanovskij D.V., Shponarskij I.V., Barinova P.A. (2025) Application of the Sheridan and Verplank automation level scale for an objective assessment of business intelligence tools. *E-Management*, vol. 8, no. 4, pp. 22–34. DOI: 10.26425/2658-3445-2025-8-4-22-34

© Klyuev R.V., Makarov V.S., Stefanovskij D.V., Shponarskij I.V., Barinova P.A., 2025.

This is an open access article under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).



ВВЕДЕНИЕ / INTRODUCTION

В последние годы наблюдается стремительное расширение рынка инструментов для бизнес-аналитики (англ. Business Intelligence, далее – BI). Наряду с классическими визуальными Low-Code-платформами, предназначенными для быстрой сборки дашбордов и аналитики самообслуживания (англ. Self-Service Analytics), активно развиваются консольные решения и конвейеры с интерфейсом командной строки (англ. Command Line Interface, далее – CLI), интегрированные с большими языковыми моделями (англ. Large Language Model, далее – LLM), иначе говоря, конвейеры CLI-LLM. Эта динамика меняет сам характер рабочего процесса бизнес-аналитика и расширяет спектр автоматизируемых задач: от подготовки данных до автоматической генерации визуализаций и деплоя артефактов.

Однако наряду с технологическим прогрессом возникают новые системные вызовы, особенно в условиях геополитической нестабильности. С 2022 г. значительное число западных вендоров BI-решений прекратили поддержку своих продуктов на территории Российской Федерации. Уход таких компаний, как Tableau (Salesforce), Microsoft (Power BI), Qlik, привел не только к потере доступа к привычным коммерческим платформам, но и к резкому сокращению локальной экспертизы: уехали специалисты, прекратились обучающие программы, исчезли каналы технической поддержки и сообщества пользователей. В результате организации вынуждены перестраивать свои BI-ландшафты «с нуля», часто переходя на open-source (иначе говоря, с открытым исходным кодом) альтернативы или разрабатывая собственные решения. При этом выбор между визуальными Low-Code-платформами и CLI-инструментами все чаще носит субъективный характер, основанный на личном опыте отдельных сотрудников, а не на объективных метриках эффективности и устойчивости.

Повышение степени автоматизации выполняемых бизнес-аналитиком функций создает новые преимущества, включающие сокращение time-to-insight (времени, необходимого для получения выводов, на основе которых можно принимать решения) и снижение объема ручной рутинной работы. Однако вместе с этими преимуществами возникают и новые серьезные вызовы.

Во-первых, разнообразие BI-инструментов и подходов усложняет выбор платформы: как уже было сказано ранее, сотрудники организаций, в том числе бизнес-аналитики, по-прежнему во многом опираются на свой субъективный опыт, что может приводить к ошибкам проектирования, избыточным затратам на интеграцию BI-инструментов с используемыми корпоративными информационными системами (например, CRM, ERP и т.д.) и сложностям масштабирования решений (в больших организациях может потребоваться внедрение BI-инструментов сразу в несколько подразделений, при этом важно обеспечить одинаковый подсчет общих метрик (например, прибыли) и возможность увеличения количества пользователей корпоративной системы, а также объема данных) [Alamin, Malakar, Uddin, Afroz, Haider, Iqbal, 2021].

Во-вторых, интеграция LLM в конвейеры CLI приводит к появлению новых источников уязвимости, таких как зависимость от внешних сетевых сервисов, чувствительность результатов анализа к формулировке запросов (промптов), повышенные требования к управлению версиями и обеспечению безопасности. Так, необдуманное использование LLM и результатов их работы может повлечь за собой множество рисков: утечка данных при взаимодействии с внешними API, создание уязвимостей программного обеспечения посредством внедрения сгенерированного небезопасного кода или некорректных или неэффективных SQL-запросов. Все это делает вопрос оценки пригодности для работы конкретного BI-инструмента особенно актуальным [Sahoo, Singh, Saha, Jain, Mondal, Chadha, 2025; Fagbohun, Harrison, Dereventsov, 2024].

В этих условиях особенно остро встает проблема объективной оценки BI-инструментов. Традиционные подходы, ориентированные на удобство интерфейса, скорость построения визуализаций или совместимость с источниками данных, уже не отражают всей полноты современных требований, таких как автономность, устойчивость к сбоям, энергоэффективность и способность функционировать в условиях ограниченных ресурсов (например, на edge-устройствах с архитектурой RISC-V). Более того, в условиях ухода западных решений и экспертизы критически важно иметь методологию, позволяющую не просто сравнивать функциональность, но и оценивать степень зависимости от внешних факторов, будь то облачные LLM, проприетарные библиотеки или специфические аппаратные платформы.

В связи с этим способы оценки BI-инструментов, в том числе их степени автоматизации, становятся объектом интереса множества исследователей. Так, Ж.О. Эрнандес и П. Фонсэка-и-Касас в статье Business Intelligence's Self-Service Tools Evaluation провели сравнительный анализ self-service BI-инструментов на основе

значений 82 различных метрик, сгруппированных по таким категориям, как загрузка данных, модель данных, связи полей, анализ, дашборды и отчетность, а также предложили использовать адаптированную в их работе системную модель качества (англ. Systemic Quality Model, SQMO) в качестве инструмента для оценки BI-инструментов [Hernández, Fonseca, 2022].

В другой работе под названием *Towards Autonomous Analytics: The Evolution of Self-Service BI Platforms with Machine Learning Integration* исследователь Ш. Сайед предлагает подход, при котором BI-инструменты смогут выполнять автоматическую и автономную аналитику данных с минимальным взаимодействием с пользователем или вообще без него [Syed, 2022].

Особый интерес представляют подходы, которые позволяют количественно оценивать степень автономии системы. Так, группой исследователей Ж. Лопесом, М. Фариа и М.Ф. Сантосом в статье *Exploring trends and autonomy levels of adaptive business intelligence in healthcare: A systematic review* была описана возможность использования в сфере здравоохранения предложенной Т.Б. Шериданом классификации автоматизации для определения уровней автономии BI-инструментов, что позволяет объективно оценивать степень автоматизации процессов [Lopes, Faria, Santos, 2024; Sheridan, 2002].

В одной из своих более ранних работ Т.Б. Шеридан в соавторстве с У.Л. Вепланком разработал шкалу уровней автоматизации (англ. Levels of Automation, далее – LOA) [Sheridan, Verplank, Brooks, 1978]. Шкала уровней автоматизации позволяет количественно определить степень участия человека в выполнении операций. Изначально данная шкала была разработана в контексте управления телеоператорами, впоследствии она неоднократно использовалась в других исследованиях человеко-машинного взаимодействия [Moray, Rodriguez, Clegg, 2000].

Именно в этом контексте актуально использование в сфере бизнес-аналитики шкалы уровней автоматизации Шеридана и Вепланка – инструмента, демонстрирующего высокую применимость и в современных цифровых средах. Так, применение этой шкалы к BI-инструментам позволяет не только количественно сравнить разные подходы (например, Low-Code-платформы и конвейеры CLI-LLM), но и выявить их уязвимости в условиях реального использования, особенно, когда организация не может рассчитывать на стабильную поддержку со стороны вендора или наличие квалифицированных экспертов в области BI-инструментов «под рукой».

Среди отечественных исследований на тему применения шкалы уровней автоматизации Шеридана и Вепланка можно отметить работу И.Г. Кораблева «Оценка уровня автоматизации бизнес-процессов предприятия» [Korablev, 2016]. Автор предлагает методику анализа операций для производственных операций, где шкала уровня автоматизации Шеридана и Вепланка используется для проведения формализованной числовой оценки степени автоматизации операций. Адаптация методики И.Г. Кораблева к сфере бизнес-аналитики в виде простого алгоритма оценки открывает широкие возможности для объективного сравнения BI-инструментов по уровню автоматизации ключевых этапов процесса работы с аналитическими артефактами, от загрузки данных до публикации результатов самого процесса.

Целью настоящего исследования является проверка возможности применения шкалы уровней автоматизации Шеридана и Вепланка для объективной оценки BI-инструментов, используемых при построении интерактивных дашбордов и диаграмм процессов, а именно Low-Code платформ и конвейера CLI-LLM (далее в статье под BI-инструментами будут подразумеваться эти инструменты).

Для достижения поставленной цели решаются следующие задачи:

- постановка эксперимента для изучения возможности применения методики оценки уровня автоматизации И.Г. Кораблева, основанной на использовании шкалы уровней автоматизации Шеридана и Вепланка [Korablev, 2016];
- проведение экспериментального сравнения Low-Code платформ и конвейера CLI-LLM по степени автоматизации процессов;
- анализ эргодичности значений, используемых для оценки LOA BI-инструментов метрик, полученных в ходе экспериментального сравнения.

МАТЕРИАЛЫ И МЕТОДЫ ИССЛЕДОВАНИЯ / RESEARCH MATERIALS AND METHODS

Для проверки возможности применения шкалы уровней автоматизации Шеридана и Вепланка для объективной оценки BI-инструментов был спланирован и проведен контролируемый эксперимент с использованием различных платформ и технологических стеков. При этом в ходе эксперимента не только измерялись уровни автоматизации отдельных операций, но и оценивалась устойчивость этих решений в условиях ограниченных

ресурсов и разнообразия архитектур аппаратных платформ, что особенно актуально в контексте технологической суверенности и импортозамещения.

Эксперимент проводился на одноплатном компьютере Orange Pi RV2 с частотой 1 ГГц и 1 ГБ оперативной памяти. Такие параметры позволят проверить, как будет осуществляться работа с разными инструментами в условиях ограниченных ресурсов при анализе эргодичности. Кроме того, компьютер Orange Pi RV2 также оснащен процессором с архитектурой RISC-V (англ. Reduced Instruction Set Computing – «компьютинг с сокращенным набором инструкций»). Выбор именно этой платформы обусловлен ее стратегическим значением: RISC-V является открытой, лицензионно свободной архитектурой, что делает ее перспективной для создания независимых вычислительных систем, особенно в условиях ограничений на поставку западного оборудования и программного обеспечения. В отличие от проприетарных архитектур x86 (Intel/AMD) и ARM (Apple, Qualcomm) RISC-V позволяет разрабатывать собственные «системы на кристалле» (англ. System on Chip, SoC – микросхема, которая объединяет основные компоненты и функции устройства на одном чипе) без выплаты роялти, что открывает возможности для локализации производства и разработки отечественных микросхем. Кроме того, архитектура RISC-V может быть более энергоэффективной по сравнению с другими архитектурами за счет простоты набора команд (каждая инструкция выполняется максимально быстро), что подходит для edge-устройств [Suarez, Almeida, Blanco, 2024]. Однако эта архитектура все еще находится на этапе активного развития, и не все программные решения корректно функционируют на ней, особенно те, что включают сложные зависимости или требуют оптимизации машинного кода под заданную архитектуру.

Одним из ключевых элементов конвейера CLI-LLM стала модель gemma3:27b, запущенная через среду фреймворка Ollama. Ollama – это легковесный фреймворк для локального запуска LLM, поддерживающий множество моделей, включая Llama, Mistral, Gemma. Его преимущество заключается в простоте развертывания, автоматическом управлении зависимостями и поддержке GPU-ускорения, однако при работе на архитектуры RISC-V, где отсутствует полноценная поддержка CUDA (программная платформа от NVIDIA для выполнения вычислений на графических процессорах) и других библиотек для работы с графическими процессорами (GPU), Ollama полагается исключительно на вычисления, выполняемые центральным процессором (CPU), что значительно снижает скорость генерации. Тем не менее Ollama демонстрирует высокую портируемость: он может быть собран из исходных кодов на всех трех архитектурах (здесь и далее речь идет об архитектурах RISC-V, ARM и x86), хотя для работы на устройствах с архитектурой RISC-V требуется дополнительная настройка компилятора и ручное разрешение зависимостей.

Альтернативой Ollama является llama.cpp – библиотека с открытым исходным кодом, написанная на языке C++ и предназначенная для эффективной работы LLM на CPU. Ее ключевое преимущество – минимальные системные требования и высокая производительность даже на слабых устройствах. Библиотека llama.cpp использует квантование (например, GGUF-формат – специальный формат хранения LLM-моделей), что позволяет загружать модели размером 7–13B (от 7 до 13 млрд) параметров на устройства с 4–8 ГБ RAM. На архитектуре x86 она показывает отличную производительность благодаря поддержке AVX2/AVX-512 инструкций. На архитектуре ARM (в частности, на процессорах Apple Silicon M1/M2) библиотека работает стабильно за счет оптимизаций под NEON. NEON представляет собой технологию на базе SIMD (англ. Single Instruction, Multiple Data – это модель параллельных вычислений, позволяющая процессору несколько операций одновременно) для устройств на архитектуре ARM. Вместе с этим поддержка на архитектуре RISC-V пока ограничена: отсутствуют специфические SIMD-инструкции, необходимые для ускорения матричных операций, что делает исполнение медленным и энергозатратным. Тем не менее ведутся активные разработки по добавлению V-расширения (векторные инструкции) в RISC-V, что в будущем может кардинально улучшить ситуацию.

Сравнение фреймворка Ollama и библиотеки llama.cpp показывает четкую разницу в их философии: Ollama ориентирован на удобство и универсальность, тогда как llama.cpp – на эффективность и контроль. Ollama проще в развертывании, но менее гибкий; llama.cpp требует больше усилий для настройки, но позволяет тонко управлять потреблением ресурсов устройства. Оба решения могут работать на всех трех архитектурах, но с разной степенью успеха.

Также рассмотрены и другие инструменты, такие как MLC LLM, TinyGrad и Hugging Face Transformers с ONNX Runtime. Инструмент MLC LLM предлагает возможность компилировать модели под целевую архитектуру, что потенциально делает его идеальным для архитектуры RISC-V. TinyGrad – минималистичный фреймворк для глубокого обучения, способный работать на ограниченных устройствах, но с небольшим набором

поддерживаемых моделей. Hugging Face Transformers – наиболее популярный выбор для исследователей, однако его зависимость от PyTorch и большого объема RAM делает его малоприменимым для edge-устройств, особенно на архитектуре RISC-V.

В ходе эксперимента также была протестирована совместимость gemma3:27b на всех трех архитектурах. На устройствах с архитектурой x86 модель запускалась быстро с использованием GPU-ускорения через фреймворк Ollama. На одноплатном компьютере Raspberry Pi на архитектуре ARM модель запускалась медленнее, но все еще с приемлемой скоростью при квантовании до 4-битного формата. На архитектуре RISC-V запуск возможен только при условии использования сильно квантованной версии (например, Q4_K_M), и даже тогда время генерации одного запроса составляло более двух минут, что делает ее практически непригодной для интерактивного использования.

Можно сделать предварительный вывод: высокий уровень автоматизации, достигаемый за счет LLM, напрямую зависит от аппаратной платформы. То, что на архитектуре x86 выглядит как автономный процесс, на RISC-V может превратиться в ручную операцию из-за задержек и сбоев. Это подчеркивает необходимость оценивать не только функциональность инструмента, но и его архитектурную устойчивость – способность сохранять LOA при переходе между разными вычислительными средами.

Сам эксперимент спланирован как факторный план 2×2 . Первым фактором является инструментальная среда, в которой проводится эксперимент, он включает следующие два уровня:

- уровень А – Low-Code платформы Apache Superset и Draw.io;
- уровень В – CLI-LLM-конвейеры JupyterLab, библиотеки Python Dash и Plotly, PlantUML.

Второй фактор представляет собой тип артефакта и состоит из следующих уровней:

- уровень S1 – интерактивный дашборд As-Is, позволяющий отслеживать текущее состояние системы или бизнес-процесса;
- уровень S2 – диаграмма процессов To-Be, визуализирующая целевое состояние системы или бизнес-процесса.

Эксперимент включил четыре различных сценария разработки артефактов:

- А × S1 – создание интерактивного дашборда в Apache Superset;
- А × S2 – построение диаграммы процессов в Draw.io;
- В × S1 – создание интерактивного дашборда с помощью CLI-LLM-конвейера на базе JupyterLab, а также библиотек Python Dash и Plotly;
- В × S2 – построение диаграммы процессов с использованием CLI-LLM-конвейера и PlantUML.

JupyterLab использовался как интерактивная среда разработки, позволяющая сочетать документацию, визуализацию и генерацию кода. Библиотеки Dash и Plotly обеспечивали создание динамических веб-интерфейсов для интерактивных дашбордов, а PlantUML – текстовое описание диаграмм процессов, которое автоматически преобразуется в графическое представление. Интеграция с LLM происходила через локальный запуск модели gemma3:27b в среде фреймворка Ollama, что исключало передачу данных в облачные сервисы и повышало безопасность работы.

Каждый сценарий согласно методике поиска узких мест бизнес-процессов И.Г. Кораблева был разбит на четыре элементарные операции.

1. D1 – загрузка и очистка данных. В ходе этой операции формируется входной массив данных, с которым осуществляется дальнейшая работа. В случае Low-Code платформ это подключение к источнику данных, настройка соединений, фильтрация и обработка пропусков или дубликатов. В случае конвейеров CLI-LLM сюда входят написание или автогенерация кода для чтения файлов, удаление шумов, преобразование форматов дат, обработка числовых показателей.

2. D2 – кодирование визуальных компонентов. Данная операция преобразует данные из входного массива в элементы интерфейса или схемы, иначе говоря, определяет какие объекты данных будут показаны и в каком виде. Для интерактивных дашбордов это задание осей графиков, типов диаграмм, параметров фильтрации, для процессных диаграмм – перевод текстового описания в блоки, стрелки и условные обозначения.

3. D3 – раскладка и стилизация. Во время этой операции формируется финальный облик артефакта таким образом, чтобы было как можно проще воспринимать отображаемые в нем данные. В дашбордах в данную операцию включено позиционирование графиков на рабочем пространстве, настройка цвета, легенд, подписей, шрифтов, размеров, а в диаграммах процессов упорядочивание блоков, выравнивание элементов и цветовое кодирование этапов.

4. D4 – публикация и версионирование. Для интерактивных дашбордов это публикация в интерфейсе Apache Superset или деплой через программу, включающую библиотеки Dash и Plotly, на сервер, в то время как для диаграмм процессов это экспорт в форматах .png и drawio (для Low-Code платформ), а также сохранение версии в Git-репозитории (для конвейеров CLI-LLM). Для этой операции учитываются вопросы контроля версий, совместного доступа и возможности последующего изменения.

При реализации сценариев $A \times S1$ и $B \times S1$, в ходе которых рассматривалось создание интерактивного дашборда, использовался набор данных в файле sales.csv (пример данных из этого файла представлен в табл. 1), включающий следующие столбцы:

- event_id – уникальный идентификатор события;
- sale_id – уникальный идентификатор заказа (дополнительный идентификатор нужен для тех случаев, когда у одного заказа может быть несколько статусов, например, когда пользователь сначала получил заказа, а затем отказался от него);
- timestamp – время события;
- customer_id – уникальный идентификатор покупателя;
- total_amount – общая сумма заказа;
- channel – «канал» продажи (магазин или онлайн);
- region – регион или город, где был сделан заказ;
- status – статус заказа.

Таблица 1. Пример данных из файла sales.csv

Table 1. Example of data from the sales.csv file

event_id	sale_id	timestamp	customer_id	total_amount	channel	region	status
1	1	2025-09-15T10:30:00Z	CUST001	3 000,00	Онлайн	Москва	Completed
2	1	2025-09-16T09:15:00Z	CUST001	3 000,00	Онлайн	Москва	Returned
3	2	2025-09-15T11:45:00Z	CUST002	2 500,50	Магазин	Санкт-Петербург	Completed
4	2	2025-09-16T15:30:00Z	CUST002	2 500,50	Магазин	Санкт-Петербург	Cancelled
5	3	2025-09-15T12:15:00Z	CUST003	1 000,00	Магазин	Новосибирск	Completed

Составлено авторами по материалам исследования / Compiled by the authors on the materials of the study

Для сценариев построения диаграмм процессов $A \times S2$ и $B \times S2$ использовалось текстовое описание процесса согласования закупок в файле формата TXT. Все сценарии выполнялись одним и тем же оператором, для объективности эксперимента обладающим навыками работы как с Low-Code платформами, так и с конвейерами CLI-LLM.

Важным аспектом эксперимента стало определение и формализация критериев оценки уровня автоматизации для каждой из четырех операций D1–D4. В основу положена методика И.Г. Кораблева, которая предполагает числовую оценку степени участия человека в выполнении операции по шкале LOA Шеридана и Вепланка от 1 (полностью ручное выполнение) до 10 (полная автономия системы) [Korablev, 2016]. Операции с $LOA < 6$ классифицируются как ручные, выполняемые, $LOA = 6$ – как автоматические, требующие контроля человеком, $LOA > 6$ – как автоматические.

В статье И.Г. Кораблева предлагается логирование каждой выполняемой операции в системе с фиксацией LOA. В соответствии с этим оценка уровня автоматизации операций проводилась на основе анализа записи экрана и системных логов, что позволило детально зафиксировать каждое действие пользователя, включая клики мышью, ввод команд, корректировку сгенерированного LLM кода, повторные запросы и т.д.

В ходе проведения эксперимента получены следующие значения LOA для каждой операции по всем четырем сценариям (табл. 2–5).

Таблица 2. Степень автоматизации операций при создании интерактивного дашборда в Apache Superset (сценарий А × S1)

Table 2. Degree of automation of operations in creating an interactive dashboard in Apache Superset (Scenario A × S1)

Операция	Описание операции	LOA
D1	Пользователь вручную импортирует набор данных, очищает его с помощью GUI-помощников Superset	4
D2	Визуальный конструктор помогает выбирать чарты и фильтры без написания кода	5
D3	Ручная раскладка интерактивного дашборда с базовыми автоматическими сетками	4
D4	Публикация интерактивного дашборда через встроенный интерфейс Superset	5

*Составлено авторами по материалам исследования / Compiled by the authors on the materials of the study***Таблица 3.** Степень автоматизации операций при построении диаграммы процессов в Draw.io (сценарий А × S2)

Table 3. Degree of automation of operations in constructing a process diagram in Draw.io (Scenario A × S2)

Операция	Описание операции	LOA
D1	Элементы диаграммы процессов создаются вручную, данные подгружаются вручную	3
D2	Пользователь на основе текстового описания вручную рисует фигуры и соединяет их линиями (стрелками)	3
D3	Ручная раскладка, ограниченные функции авторазмещения	3
D4	Экспорт или публикация выполняется вручную	2

*Составлено авторами по материалам исследования / Compiled by the authors on the materials of the study***Таблица 4.** Степень автоматизации операций при создании интерактивного дашборда с помощью CLI-LLM-конвейера на базе JupyterLab, Dash и Plotly (сценарий В × S1)

Table 4. Degree of automation of operations in creating an interactive dashboard using a CLI-LLM pipeline based on JupyterLab, Dash, and Plotly (Scenario B × S1)

Операция	Описание операции	LOA
D1	Large Language Model (LLM) помогает с написанием кода для загрузки и очистки данных	6
D2	LLM генерирует код визуализаций с использованием библиотек Python Dash и Plotly, пользователь проверяет и при необходимости вносит корректировки	7
D3	LLM предлагает варианты структуры и оформления интерактивного дашборда, включая расположение визуальных элементов, цветовые схемы и шрифты, пользователь при необходимости корректирует	6
D4	Автоматизация публикации и версионирования через скрипты Command Line Interface (CLI)	7

*Составлено авторами по материалам исследования / Compiled by the authors on the materials of the study***Таблица 5.** Степень автоматизации операций при построении диаграммы процессов с использованием CLI-LLM-конвейера и PlantUML (сценарий В × S2)

Table 5. Degree of automation of operations in constructing a process diagram using a CLI-LLM pipeline and PlantUML (Scenario B × S2)

Операция	Описание операции	LOA
D1	LLM генерирует текстовое описание диаграммы UML в синтаксисе PlantUML, диаграмма строится автоматически, данные подгружаются в нее через CLI-интерфейс	7
D2	Диаграммы процессов строятся исходя из текстового описания, вмешательство пользователя минимально	8
D3	Стилизация задается автоматически, ручная корректировка требуется редко	7
D4	Автоматический экспорт и публикация через CLI-интерфейс	8

Составлено авторами по материалам исследования / Compiled by the authors on the materials of the study

Для каждого сценария эксперимента вычислен усредненный показатель степени автоматизации LOA_{mean} – среднее арифметическое значений LOA по всем четырем операциям (табл. 6). Этот показатель использовался как интегральная метрика сравнения ВІ-инструментов в штатных условиях.

Таблица 6. Степень автоматизации для каждого сценария по всем операциям

Table 6. Degree of automation for each scenario across all operations

Сценарий	LOA _{mean}
Создание интерактивного дашборда в Apache Superset (сценарий А × S1)	4,50
Построение диаграммы процессов в Draw.io (сценарий А × S2)	2,75
Создание интерактивного дашборда с помощью CLI-LLM-конвейера на базе JupyterLab, Dash и Plotly (сценарий В × S1)	6,50
Построение диаграммы процессов с использованием CLI-LLM-конвейера и PlantUML (сценарий В × S2)	7,50

Составлено авторами по материалам исследования / Compiled by the authors on the materials of the study

При сравнении LOA_{mean} по каждому сценарию можно заметить, что сценарии, в ходе которых использовались Low-Code платформы Apache Superset и Draw.io, показали сравнительно невысокую степень автоматизации. В то же время степень автоматизации сценариев с использованием CLI-LLM конвейеров является достаточной, чтобы назвать рассматриваемые в них процессы автоматическими.

При анализе поведения системы также фиксировалось изменение LOA по операциям и общее снижение LOA_{mean} по сценарию. Такой подход позволил оценить не только производительность рассматриваемых ВІ-инструментов, но и их устойчивость, что является ключевым требованием для внедрения в промышленную эксплуатацию. Особое внимание уделено фактору эргодичности – способности системы сохранять свои характеристики при переходе в неидеальные условия.

Для анализа эргодичности системы путем оценки значений, используемых для оценки LOA ВІ-инструментов метрик, полученных в ходе эксперимента, проведено стресс-тестирование с тремя видами шоков: технологическим, аппаратным и человеческим. Каждый вид шока имитировал реальный фактор риска, способный понизить уровень автоматизации операции либо увеличить количество выполняемых ручных действий.

Технологический шок включал имитацию потери сетевого соединения, недоступности API LLM (например, временный бан Ollama или отказ сервера), сбоя в работе Docker-контейнеров. В сценариях с использованием Low-Code платформ, а именно создание интерактивного дашборда в Apache Superset и построение диаграммы процессов в Draw.io, LOA оставался стабильным, так как инструменты Superset и Draw.io работают локально и не зависят от LLM. Значение LOA_{mean} снижалось не более чем на 0,2–0,3 балла, так как появлялась необходимость ручного обновления данных или повторного экспорта диаграмм процессов и интерактивных дашбордов.

При сценариях с использованием конвейера CLI-LLM LOA резко падал, особенно в операциях D1 и D2: LLM не генерировал код, в связи с чем пользователь был вынужден писать его вручную, LOA мог снижаться до 3–4, LOA_{mean} – до 4–5.

Аппаратный шок представлял собой искусственное ограничение ресурсов Orange Pi RV2 с помощью особого механизма ядра Linux, позволяющего ограничивать, учитывать и изолировать ресурсы, используемые группами процессов – cgroups. Ограничения выглядели следующим образом: снижение доступной RAM до 512 МБ и CPU-лимита до 50 %, что позволило смоделировать работу на перегруженном устройстве или в edge-среде с ограниченными возможностями. В сценариях с использованием Low-Code платформ LOA также не менялся. При использовании конвейера CLI-LLM могли происходить таймауты и падения, из-за чего пользователю приходилось разбивать задачи и повторять запросы. LOA операций D3 и D4 снижался до 4–5, LOA_{mean} – до 5–6.

К человеческому шоку относились непредсказуемые действия и ошибки пользователя: некорректные промпты, опечатки в CLI-интерфейсе, пропуск этапов подтверждения, голосовые команды с фоновым шумом. В сценариях использования Low-Code платформ LOA также оставался неизменным, а в сценариях с применением конвейеров CLI-LLM, чувствительных к качеству промпта и речи, LOA операций D2 и D3 могли падать до 5–6, LOA_{mean} – до 5–6.

Исследование было направлено не только на количественную оценку автоматизации построения диаграмм процессов и интерактивных дашбордов, но и на анализ ее устойчивости в реальных условиях, особенно в контексте работы на различных аппаратных платформах. Учет архитектурных особенностей (x86, ARM, RISC-V)

и чувствительности к внешним и внутренним сбоям позволяет сделать выводы, применимые не только в научном, но и в практическом плане – при выборе ВІ-решений для организаций, стремящихся к технологической независимости и устойчивости к внешним вызовам.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ / STUDY RESULTS

Эксперимент показал, что Low-Code-платформы (Apache Superset, Draw.io) в штатном режиме обеспечивают средний уровень автоматизации, в то время как CLI-LLM-конвейеры (JupyterLab/Dash/Plotly и PlantUML) демонстрируют значительно более высокие значения LOAmean. При стресс-тестировании выявлено, что шоки сильнее всего влияют на подходы с использованием конвейеров CLI-LLM. Это указывает на то, что высокий уровень автоматизации в штатном режиме не гарантирует устойчивости к сбоям.

На основе проведенного эксперимента разработан пошаговый алгоритм оценки ВІ-инструментов:

– определить возможные сценарии применения ВІ-инструментов с указанием выполнения конкретных операций;

– собрать данные о количестве ручных действий по каждой операции;

– присвоить LOA каждой операции по шкале Шеридана и Вепланка;

– вычислить LOAmean как среднее арифметическое по операциям;

– провести стресс-тестирование для оценки эргодичности;

– сравнить LOA операций и LOAmean сценариев до и после шоков;

– сделать вывод о пригодности инструмента с учетом как автоматизации, так и устойчивости.

В дальнейшем при оценке ВІ-инструментов предлагается измерять не только LOA, но и другие метрики, такие как время разработки, количество ручных действий, пиковое потребление оперативной памяти и пиковая загрузка процессора.

В своей статье И.Г. Кораблев отмечает, что шкала LOA Шеридана и Вепланка может дополняться другими критериями, включая порог умственной нагрузки оператора, чтобы получить более комплексную оценку человеко-машинного взаимодействия [Braarud, 2021]. В перспективе возможно интегрировать автоматическую оценку LOA с психофизиологическими опросниками, (например, NASA-TLX) и сформировать метрику «эффективность + UX», отражающую не только скорость и автоматизацию, но и комфорт, когнитивную нагрузку, а также субъективную удовлетворенность [Braarud, 2020].

Предлагается логирование каждой выполняемой операции в системе с фиксацией LOA. Автоматизированное логирование кликов, команд, загрузки данных и публикации артефактов позволит собирать объективные данные без участия наблюдателя и в реальном времени вычислять LOA и другие предложенные ранее метрики. Такой подход может лечь в основу корпоративных систем мониторинга продуктивности ВІ-инструментов и послужить основанием для внедрения новых платформ и инструментов.

ВЫВОДЫ / DISCUSSION

Результаты проведенного эксперимента подтверждают, что CLI-LLM-конвейеры обеспечивают значительно более высокий уровень автоматизации по сравнению с Low-Code-платформами в штатных условиях. Однако стресс-тестирование выявило критическую уязвимость таких решений: при возникновении технологических, аппаратных или человеческих шоков уровень автоматизации резко снижается, что ставит под сомнение их надежность в реальных условиях эксплуатации. Этот парадокс – высокая автоматизация при низкой устойчивости – требует особого внимания при выборе ВІ-инструментов, особенно в контексте современных вызовов цифровой суверенности и технологической независимости.

Особую значимость приобретает вопрос кроссплатформенной совместимости и способности ВІ-инструментов функционировать на различных процессорных архитектурах – x86, ARM и RISC-V. В условиях ухода западных вендоров и ограничений на поставку оборудования и программного обеспечения российские организации все чаще вынуждены переходить на отечественные или альтернативные вычислительные платформы. Архитектура RISC-V как открытая и лицензионно свободная становится все более привлекательной для разработки edge-устройств, встраиваемых систем и автономных аналитических решений. Однако не все ВІ-инструменты одинаково хорошо адаптированы к работе на таких платформах.

В ходе эксперимента использовалась плата Orange Pi RV2 на базе RISC-V, что позволило оценить производительность и стабильность инструментов в условиях ограниченных ресурсов и «нестандартной» архитектуры.

Low-Code-платформы, такие как Apache Superset и Draw.io, продемонстрировали высокую устойчивость: за счет своей архитектуры, основанной на использовании веб-технологий и контейнеризации, они обеспечивают относительную независимость от аппаратной платформы. В то же время CLI-LLM-конвейеры оказались чувствительны к ограничениям CPU и RAM, характерным для устройств на архитектуре RISC-V. Более того, ряд пакетов языка Python и библиотек для машинного обучения (например, TensorFlow, scikit-learn) не всегда корректно компилируются или работают на устройствах с архитектурой RISC-V без настройки окружения, что создает дополнительные барьеры для внедрения.

Это подчеркивает важность оценки не только функциональности, но и портируемости VI-инструментов. Высокие показатели метрики LOAmean в идеальных условиях на мощном сервере с архитектурой x86 могут оказаться недостижимыми на устройствах с архитектурой ARM в полевых условиях или с архитектурой RISC-V в автономном режиме. Следовательно, при выборе инструмента необходимо учитывать не только его автоматизационный потенциал, но и способность сохранять этот потенциал на разных архитектурах, то есть обеспечивать эргодичность в пространстве вычислительных платформ.

Такой подход особенно актуален для организаций, стремящихся к технологической независимости. В условиях санкционного давления и фрагментации глобального рынка информационных технологий (ИТ-рынка) способность разворачивать аналитические решения на отечественном или «нейтральном» оборудовании становится стратегическим преимуществом. Open-source-инструменты, такие как Apache Superset, JupyterLab или PlantUML, теоретически обладают такой гибкостью, однако на практике их работа на RISC-V требует значительных усилий по адаптации зависимостей, оптимизации производительности и обеспечении стабильности. В отличие от них проприетарные решения, даже если бы они оставались доступны, часто жестко привязаны к x86-экосистеме и облачным сервисам, что делает их непригодными для автономного использования.

Интеграция LLM в CLI-конвейеры вносит дополнительный слой зависимости от внешних API или локальных моделей, требующих значительных вычислительных ресурсов. На устройствах с архитектурой RISC-V запуск даже небольших LLM (например, Llama 3 8B) может быть затруднен без специализированного ускорителя. Это означает, что высокий LOA в операциях D1–D3, достигнутый за счет LLM, может оказаться недостижимым в условиях ограниченной аппаратной базы. Таким образом, автоматизация, основанная на «тяжелых» моделях искусственного интеллекта, не всегда совместима с требованиями энергоэффективности и автономности, характерными для edge- и embedded-сценариев.

Отсюда следует важный практический вывод: при проектировании VI-ландшафта необходимо проводить многоуровневую оценку, включающую:

- измерение значений метрики LOAmean в штатных условиях;
- стресс-тестирование на различных типах шоков;
- проверку работоспособности на целевых аппаратных платформах (на архитектурах x86, ARM, RISC-V);
- анализ зависимости от внешних сервисов и ресурсоемкости.

Такой подход позволяет избежать ситуаций, когда VI-инструмент, показавший высокую автоматизацию в лаборатории, оказывается неработоспособным в реальных условиях эксплуатации, особенно в распределенных, ресурсоограниченных или геополитически изолированных средах.

Наконец, стоит отметить, что устойчивость Low-Code-платформ, продемонстрированная в эксперименте, не означает их универсального превосходства. Они уступают в гибкости, масштабируемости и возможностях интеграции в DevOps-процессы. Однако в условиях, где приоритетом являются стабильность, предсказуемость и независимость от внешних факторов, такие платформы могут оказаться более предпочтительными, несмотря на более низкий LOA. Это особенно верно для организаций, где кадровая экспертиза в области программирования ограничена, а уход западных вендоров лишил доступ к обучению и поддержке.

Выбор VI-инструмента должен основываться не на максимизации уровня автоматизации как таковой, а на балансе между автоматизацией, устойчивостью и совместимостью с целевой ИТ-инфраструктурой. Шкала уровней автоматизации Шеридана и Вепланка, дополненная анализом эргодичности и кроссплатформенной оценкой, может предоставлять основу для принятия таких взвешенных решений.

ЗАКЛЮЧЕНИЕ / CONCLUSION

Проведенное исследование предоставило объективную, количественную метрику, которая устраняет субъективность в оценке VI-инструментов и позволяет принимать взвешенные архитектурные решения. Применение

шкалы уровней автоматизации Шеридана и Вепланка показало, что CLI-LLM-конвейеры в штатных условиях демонстрируют высокий уровень автоматизации. В то же время стресс-тестирование выявило ограничения высокой автоматизации: технологические, аппаратные и человеческие шоки снижали LOAmean CLI-LLM-сценариев на 1,5–2,5 балла, тогда как Low-Code-инструменты теряли не более 0,2–0,3 балла. Это подчеркивает необходимость учитывать эргодичность и устойчивость системы к сбоям при выборе VI-инструментов.

Значение настоящего исследования для практики заключается в том, что шкала LOA может быть использована для обеспечения возможности объективного выбора, наиболее подходящего для конкретных задач VI-инструмента, при сокращении избыточных затрат (на интеграцию с используемыми корпоративными системами, масштабирование VI-инструментов в пределах организации и т.д.). Представленный пошаговый алгоритм оценки VI-инструментов позволяет не только оценить уровень автоматизации по операциям, но и дополнить его стресс-тестами, метриками эргодичности и когнитивной нагрузки, создавая целостную картину пригодности инструмента для внедрения и промышленной эксплуатации в конкретных условиях деятельности организации. Расширение метрик позволит выбирать VI-инструменты исходя не только из функциональности и уровня автоматизации, но и из устойчивости, ресурсоемкости и влияния на пользователя. Это позволяет рассматривать шкалу LOA в качестве основы для построения целостной системы оценки эффективности и удобства VI-инструментов, которая может работать как в ручном, так и в полностью автоматизированном режиме через систему логирования.

В ходе распространения LLM в качестве повсеместно используемых инструментов деятельности, когда граница между человеком и машиной размывается, шкала LOA Шеридана и Вепланка предоставляет четкую карту, позволяющую понять, кто на самом деле управляет процессом проведения бизнес-аналитики от обработки данных до визуализации. Встраивание автоматической фиксации LOA и интеграция с опросниками типа NASA-TLX создают перспективу комплексной метрики «эффективность + UX», которая позволит менеджерам сферы информационных технологий, архитекторам и иным лицам, ответственным за цифровизацию, вырабатывать и принимать решения, качественно нового уровня точности и обоснованности.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

- Alamin M.A.A., Malakar S., Uddin G., Afroz S., Haider T.B., Iqbal A.* An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. arXiv. 2021. <https://doi.org/10.1109/MSR52588.2021.00018>
- Braarud P.O.* An efficient screening technique for acceptable mental workload based on the NASA Task Load Index-development and application to control room validation. *International Journal of Industrial Ergonomics*. 2020;76. DOI: <https://doi.org/10.1016/j.ergon.2019.102904>.
- Braarud P.O.* Investigating the validity of subjective workload rating (NASA TLX) and subjective situation awareness rating (SART) for cognitively complex human-machine work. *International Journal of Industrial Ergonomics*. 2021;86. <https://doi.org/10.1016/j.ergon.2021.103233>
- Fagbohun O., Harrison R.M., Dereventsov A.* An Empirical Categorization of Prompting Techniques for Large Language Models: A Practitioner's Guide. arXiv. 2024.
- Hernandez J.O., Fonseca i Casas P.* Business Intelligence's Self-Service Tools Evaluation. *Technologies*. 2022;4(10). <https://doi.org/10.3390/technologies10040092>
- Korablev I.G.* Assessment of the level of automation of business processes of the enterprise. *Bulletin of Cherepovets State University*. 2016;1:17–24.
- Lopes J., Faria M., Santos M.F.* Exploring trends and autonomy levels of adaptive business intelligence in healthcare: A systematic review. *PLoS One*. 2024;19. <https://doi.org/10.1371/journal.pone.0302697>
- Moray N., Rodriguez D., Clegg B.A.* Levels of Automation in Process Control. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2000;44:93–96. <https://doi.org/10.1177/154193120004400125>
- Sahoo P., Singh A.K., Saha S., Jain V., Mondal S., Chadha A.* A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv. 2025.
- Sheridan T.B., Verplank W.L., Brooks T.L.* Human and Computer Control of Undersea Teleoperators. In: *14th Annual Conference on Manual Control*. 1978. Pp. 343–357. <https://doi.org/10.21236/ADA057655>

Sheridan T.B. Humans and automation: system design and research issues. 2002.

Suarez D., Almeida F., Blanco V. Comprehensive analysis of energy efficiency and performance of ARM and RISC-V SoCs. *The Journal of Supercomputing*. 2024;80:12771–12789. <https://doi.org/10.1007/s11227-024-05946-9>

Syed S. Towards Autonomous Analytics: The Evolution of Self-Service BI Platforms with Machine Learning Integration. *Journal of Artificial Intelligence and Big Data*. 2022. <https://doi.org/10.31586/jaibd.2022.1157>