

Информатика и вычислительная техника Informatics and computer engineering

Научная статья

УДК 004.9, 658.5

DOI: 10.14529/ctcr260301

МЕТОД ВЫБОРА АРХИТЕКТУРЫ ВНЕДРЕНИЯ LLM, RAG И ИИ-АГЕНТОВ В КОРПОРАТИВНЫЕ ИС И СППР

О.В. Логиновский, loginovskii@susu.ru, <https://orcid.org/0000-0003-3582-2795>

А.В. Голлай, gollaiav@susu.ru, <https://orcid.org/0000-0002-5070-6779>

С.А. Браженко, bsa@uralto.ru, <https://orcid.org/0009-0004-3524-9882>

Южно-Уральский государственный университет, Челябинск, Россия

Аннотация. Цель исследования. Целью исследования является разработка метода выбора архитектуры внедрения LLM-подсистем, решений класса Retrieval-Augmented Generation (RAG) и ИИ-агентов в корпоративные информационные системы и системы поддержки принятия решений. Актуальность работы обусловлена тем, что в enterprise-среде выбор интеллектуальной подсистемы определяется не только качеством генерации, но и требованиями к объяснимости, управляемости, глубине интеграции, работе с корпоративными данными, допустимому уровню автономности и цене ошибки. **Материалы и методы.** В работе использованы системный и сравнительный анализ, архитектурное моделирование, классификация, экспертная декомпозиция критериев и элементы многокритериального выбора. Рассматриваются пять архитектурных классов: изолированная LLM-подсистема, подсистема класса RAG, LLM-контур с инструментальным доступом, агентная интеллектуальная подсистема и гибридная управляемая архитектура. Для их сопоставления сформирована система критериев, включающая тип задачи, потребность в корпоративном контексте, необходимость действий во внешних системах, требования к точности, объяснимости и аудируемости, чувствительность данных, зрелость enterprise-контура и допустимый уровень автономности. **Результаты.** Предложен полуформальный метод выбора архитектуры внедрения, сочетающий отсеивающие и ранжирующие критерии, весовую корректировку и возможность определения переходной архитектуры при недостаточной зрелости данных и интеграций. Метод представлен как воспроизводимая процедура, включающая формирование профиля задачи, фильтрацию по критическим ограничениям, оценивание допустимых архитектур по шкале соответствия и уточнение выбора с учётом зрелости корпоративной среды. Практическая применимость метода показана на типовых сценариях торгово-производственного предприятия: доступ к внутренним регламентам, подготовка управленческой сводки, рекомендация менеджеру по типовой заявке и многошаговая обработка внутреннего инцидента. Показано, что разные классы задач закономерно приводят к различным архитектурным решениям: retrieval-ориентированным, tool-augmented, agent-based или hybrid controlled. **Заключение.** Установлено, что универсальной архитектуры внедрения интеллектуальной подсистемы для корпоративных ИС и СППР не существует. Рациональный выбор определяется совокупностью технологических, организационных, инфраструктурных и управленческих факторов. Предложенный метод может использоваться при предварительном проектировании и поэтапном развитии интеллектуальных подсистем в организациях, ориентированных на автоматизацию управленческих и информационно-аналитических процессов.

Ключевые слова: большие языковые модели, LLM, Retrieval-Augmented Generation, RAG, ИИ-агенты, корпоративные информационные системы, системы поддержки принятия решений

Для цитирования: Логиновский О.В., Голлай А.В., Браженко С.А. Метод выбора архитектуры внедрения LLM, RAG и ИИ-агентов в корпоративные ИС и СППР // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2026. Т. 26, № 3. С. 5–15. DOI: 10.14529/ctcr260301

A METHOD FOR SELECTING THE ARCHITECTURE OF LLM, RAG, AND AI AGENT DEPLOYMENT IN CORPORATE INFORMATION SYSTEMS AND DECISION SUPPORT SYSTEMS

O.V. Loginovskiy, loginovskii@susu.ru, <https://orcid.org/0000-0003-3582-2795>

A.V. Hollay, gollaia@susu.ru, <https://orcid.org/0000-0002-5070-6779>

S.A. Brazhenko, bsa@uralto.ru, <https://orcid.org/0009-0004-3524-9882>

South Ural State University, Chelyabinsk, Russia

Abstract. Purpose of the study. The purpose of the study is to develop a method for selecting the deployment architecture of LLM subsystems, Retrieval-Augmented Generation (RAG) solutions, and AI agents in corporate information systems and decision support systems. The study is relevant because in the enterprise environment the choice of an intelligent subsystem depends not only on generation quality, but also on explainability, controllability, integration depth, work with corporate data, acceptable autonomy, and the cost of error. **Materials and methods.** The study uses system and comparative analysis, architectural modeling, classification, expert decomposition of criteria, and elements of multicriteria decision-making. Five architectural classes are considered: an isolated LLM subsystem, a RAG-based subsystem, an LLM contour with tool access, an agent-based intelligent subsystem, and a hybrid controlled architecture. Comparison criteria include task type, dependence on corporate context, need for actions in external systems, requirements for accuracy, explainability and auditability, data sensitivity, enterprise environment maturity, and acceptable autonomy. **Results.** A semi-formal architecture selection method is proposed. It combines filtering and ranking criteria, weight adjustment, and the possibility of determining an intermediate architecture when data and integrations are insufficiently mature. The method is implemented as a reproducible procedure including task profile formation, filtering by critical constraints, evaluation of admissible architectures, and refinement of the choice according to enterprise maturity. Its applicability is demonstrated using scenarios of a trade-and-production enterprise: access to internal regulations, managerial reporting, recommendation generation for a standard customer request, and multi-step handling of an internal incident. **Conclusion.** There is no universal deployment architecture for an intelligent subsystem in corporate information systems and decision support systems. A rational choice is determined by technological, organizational, infrastructural, and managerial factors. The proposed method can be used for preliminary design and phased development of intelligent subsystems in organizations focused on automating managerial and information-analytical processes.

Keywords: large language models, LLM, Retrieval-Augmented Generation, RAG, AI agents, corporate information systems, decision support systems

For citation: Loginovskiy O.V., Hollay A.V., Brazhenko S.A. A method for selecting the architecture of LLM, RAG, and AI agent deployment in corporate information systems and decision support systems. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2026;26(3):5–15. (In Russ.) DOI: 10.14529/ctcr260301

Введение

В последние годы большие языковые модели переходят из стадии экспериментальных решений в область практически значимых инструментов автоматизации, поиска знаний, аналитической поддержки и сопровождения управленческих процессов на предприятиях [1–3]. При этом развитие генеративного искусственного интеллекта в корпоративной среде идет по нескольким направлениям: наряду с изолированными LLM-подсистемами все шире применяются решения класса Retrieval-Augmented Generation, инструментально-расширенные конфигурации и агентные системы, способные взаимодействовать с внешними сервисами и выполнять многошаговые действия [1, 2]. Одновременно расширяется и спектр рисков, связанных с недостоверностью генерации, галлюцинациями, ограниченной проверяемостью выводов и сложностью контроля поведения интеллектуальных подсистем [3].

По мере расширения спектра доступных решений на первый план выходит не только технологическая, но и управленческая задача: какой вариант архитектуры целесообразен в конкретных

условиях предприятия. Для корпоративных информационных систем и систем поддержки принятия решений такой выбор зависит от цены ошибки, чувствительности данных, требований к объяснимости и глубины интеграции с действующей информационной инфраструктурой.

Анализ рассмотренных публикаций показывает недостаточную проработанность вопроса сопоставления изолированных LLM-подсистем, решений класса RAG и агентных систем как альтернативных архитектурных классов для корпоративных ИС и СППР. В доступной и рассмотренной литературе вопрос выбора между указанными архитектурными классами представлен фрагментарно, хотя отдельные исследования уже посвящены применению LLM и RAG в корпоративных ассистентах, использованию больших языковых моделей в производственных и MES-контурах, а также агентным моделям поддержки принятия решений [4, 5]. Вместе с тем при внедрении ИИ в организационную среду все более значимыми становятся не только характеристики самой технологии, но и стратегические, ресурсные и организационные условия внедрения [6–8].

Объектом исследования являются корпоративные информационные системы и системы поддержки принятия решений предприятий, в рамках которых рассматривается внедрение интеллектуальных подсистем на основе больших языковых моделей.

Предмет исследования составляют критерии, архитектурные паттерны и метод выбора вариантов внедрения LLM-подсистем, RAG и ИИ-агентов с учетом требований к качеству решений, безопасности, объяснимости, интеграции и степени автономности.

Цель исследования состоит в разработке метода выбора архитектуры внедрения интеллектуальных подсистем в корпоративные ИС и СППР на основе сравнительного анализа их функциональных возможностей, ограничений, рисков и условий практического применения.

Для достижения поставленной цели в работе решаются следующие задачи: проанализировать существующие подходы к внедрению больших языковых моделей в корпоративную среду; выделить и описать основные архитектурные классы внедрения; сформировать систему критериев выбора архитектуры; выполнить сравнительный анализ архитектурных классов; разработать метод предварительного выбора архитектуры; показать применимость метода на типовых сценариях торгово-производственного предприятия.

Методологическую основу исследования составляют системный и сравнительный анализ, архитектурное моделирование, классификация, экспертная декомпозиция критериев и элементы многокритериального выбора [9–13]. Практическая значимость работы определяется возможностью использования предложенного метода при проектировании и модернизации корпоративных ИС и СППР.

Научная новизна исследования заключается в следующем. Предложена классификация архитектур внедрения LLM-подсистем, решений класса RAG, tool-augmented LLM, agent-based systems и гибридной управляемой архитектуры для корпоративных ИС и СППР. Сформирована система критериев выбора архитектуры с учетом enterprise-ограничений. Разработан полуформальный метод предварительного выбора архитектуры. Показана применимость метода на типовых сценариях торгово-производственного предприятия.

Архитектурные классы внедрения LLM-подсистем в корпоративные ИС и СППР

Развитие больших языковых моделей в корпоративной среде привело к формированию нескольких устойчивых вариантов их практической реализации. Несмотря на общую опору на генеративные модели, такие решения существенно различаются по способу работы с внутренними данными предприятия, глубине интеграции с информационными сервисами, допустимому уровню автономности и требованиям к контролю. Для целей настоящего исследования выделяются пять базовых классов: изолированная LLM-подсистема, решение класса RAG, LLM-контур с инструментальным доступом, агентная интеллектуальная подсистема и гибридная управляемая конфигурация.

Изолированная LLM-подсистема

Наиболее простым вариантом реализации является изолированная LLM-подсистема, в которой ответ формируется на основе системного промпта и пользовательского запроса без опоры на внешний корпоративный контекст, retrieval-механизм или контур исполнения действий. Подобный подход удобен на ранних этапах пилотирования, однако он ограничен в зрелом контуре СППР: отсутствие доступа к актуальным корпоративным источникам снижает достоверность ответа, повышает вероятность галлюцинаций и затрудняет проверку оснований, на которых построен вывод.

Подсистема класса RAG

Подсистема класса RAG опирается на внешнее хранилище корпоративных знаний и дополняет генерацию извлечением релевантных фрагментов документов, регламентов, инструкций и записей из баз знаний. Такой подход оправдан в ситуациях, когда ответ должен опираться не на усредненные параметры модели, а на актуальное содержание внутреннего контура организации [1, 4].

Внутри данного класса целесообразно выделять graph-based RAG как специализированный вариант retrieval-подхода для случаев, где смысл распределен между связанными сущностями, а не сосредоточен в одном фрагменте текста. Graph RAG особенно полезен при multi-hop retrieval, глобальных вопросах по документному массиву, объяснимой provenance-трассировке и смешанных русско-английских корпусах, где важны связи между узлами, маршруты обхода и приоритеты наподобие PageRank [14].

LLM-подсистема с инструментальным доступом

Промежуточное положение между retrieval-решениями и агентными системами занимает LLM-подсистема, дополненная инструментальным доступом к корпоративным сервисам. В этом случае модель получает возможность обращаться к API, извлекать данные из CRM, ERP, BI, MES или иных платформ и интерпретировать результат [15]. Такой вариант особенно полезен в прикладных управленческих сценариях, где требуется не просто сослаться на документ, а получить актуальные значения показателей, статусов или остатков.

Агентная интеллектуальная подсистема

Агентный класс решений ориентирован на более сложные сценарии, в которых система должна интерпретировать цель, разложить ее на последовательность шагов, выбрать нужные инструменты, обратиться к нескольким источникам, сопоставить промежуточные результаты и при необходимости скорректировать дальнейший ход выполнения [2, 5, 16]. Подобные решения наиболее востребованы в многошаговых процессах, охватывающих несколько корпоративных сервисов и требующих orchestration на уровне бизнес-логики.

Гибридная управляемая архитектура

Наиболее зрелым вариантом внедрения для enterprise-контуров является гибридная управляемая конфигурация, объединяющая retrieval-механизм, инструментальный доступ к корпоративным сервисам, ролевые ограничения, правила валидации, журналирование и human-in-the-loop. Гибридный подход особенно уместен там, где цена ошибки высока, данные чувствительны, а система должна работать в формализованной организационной среде.

Для обобщённого сопоставления выделенных архитектурных классов далее представлена табл. 1, в которой они систематизированы по характеру работы с корпоративным контекстом и данными, уровню интеграции и автономности, требованиям к объяснимости и риск-профилю, а также по типовым сценариям применения.

Сравнительный анализ архитектурных классов внедрения

Таблица 1

Table 1

Comparative Analysis of Deployment Architecture Classes

Класс	Контекст и данные	Интеграция / автономность	Объяснимость / риск	Типовые сценарии
Изолированная LLM-подсистема	Слабая опора на корпоративные данные	Низкая / низкая	Низкая / высокий	Черновая генерация, вспомогательные задачи
RAG	Высокая опора на корпоративную базу знаний	Средняя / низкая	Высокая / средний	Доступ к регламентам, поиск знаний
Tool-augmented LLM	Данные CRM/ERP/BI	Высокая / средняя	Средняя / средний	Аналитические сводки, получение данных из ИС
Agent-based subsystem	Многошаговые процессы	Высокая / высокая	Средняя / повышенный	Инциденты, orchestration
Hybrid controlled architecture	Знания + данные + контроль	Высокая / управляемая	Высокая / контролируемый	Высокорисковые enterprise-сценарии

Как следует из табл. 1, различия между архитектурными классами проявляются не только в технической реализации, но и в степени их пригодности к использованию в конкретных управленческих и организационных условиях корпоративной среды. Следовательно, сопоставление архитектурных классов должно выполняться не только по признаку технологической сложности, но и по их соответствию профилю задачи, ограничениям enterprise-контура и требованиям к контролю результата, что требует перехода к системе критериев их выбора.

Система критериев выбора архитектуры

Сопоставление архитектурных классов внедрения требует выхода за пределы чисто технологического сравнения. В корпоративной среде решение о выборе интеллектуальной подсистемы определяется не только качеством генерации, но и тем, насколько конкретный вариант реализации соответствует управленческим задачам предприятия, ограничениям его информационной инфраструктуры и требованиям к контролю результата.

Для целей настоящего исследования критерии выбора группируются по функционально-целевому, качественному, корпоративному, интеграционно-инфраструктурному и управленческому блокам. Это позволяет связать свойства рассматриваемых архитектур не только с техническими особенностями, но и с реальными условиями внедрения на предприятии.

Функционально-целевые критерии

Ключевое значение имеет тип задачи: информационная, аналитическая, рекомендательная, исполнительная или многошаговая. Чем сложнее логика выполнения и выше зависимость от внутренних данных, тем выше требования к retrieval, интеграции и orchestration.

Критерии качества и надежности

В корпоративной среде особенно важно учитывать приемлемый уровень галлюцинаций, требования к точности, актуальности, объяснимости и проверяемости результатов [3, 17].

Критерии корпоративной среды

Существенное значение имеют чувствительность данных, необходимость закрытого развертывания и зрелость корпоративного фонда знаний. Российские исследования внедрения ИИ в организационные практики показывают, что фактическое использование подобных технологий сопровождается не только новыми возможностями, но и комплексом организационных и риск-ориентированных ограничений [6–8].

Интеграционно-инфраструктурные критерии

Критическим параметром является необходимость выполнения действий во внешних системах. Пока система ограничивается поиском, интерпретацией или подготовкой рекомендаций, достаточно информационно-аналитических решений. Если же требуется инициировать операции во внешних системах, возникает потребность в инструментальном или агентном контуре.

Управленческие критерии

К этой группе относятся цена ошибки, допустимый уровень автономности, требования к аудируемости, журналированию и регламентируемости. Именно эти параметры чаще всего определяют границы допустимого внедрения в enterprise-среде.

Сравнительный анализ архитектур внедрения

Сопоставление архитектурных классов показывает, что они различаются не только технологической сложностью, но и характером участия в информационном и управленческом цикле предприятия. Изолированные LLM-подсистемы удобны для низкорисковых вспомогательных задач; RAG-решения – для работы с корпоративными знаниями; tool-augmented LLM – для получения актуальных данных из ИС; agent-based подсистемы – для многошаговых процессов; hybrid controlled architecture – для высокорисковых enterprise-сценариев.

Следовательно, для корпоративных ИС и СППР необходим специальный метод выбора архитектуры, учитывающий одновременно тип задачи, ограничения среды и уровень зрелости интеграционного контура.

Метод выбора архитектуры внедрения

Проведенный сравнительный анализ показывает, что сопоставление архитектурных классов само по себе еще не решает задачу проектирования интеллектуальной подсистемы для корпоративной среды. Предлагаемый метод ориентирован на согласование характеристик задачи, профиля ограничений и зрелости enterprise-контура.

Предлагаемый метод задает воспроизводимую схему выбора, включающую формирование профиля задачи, фильтрацию по критическим ограничениям, ранжирование допустимых архитектур, корректировку результата по зрелости enterprise-контура и определение целевой либо переходной архитектуры.

Структура входных параметров метода

Входом метода является профиль задачи, включающий шесть групп параметров: функциональный профиль задачи; критичность ошибки; потребность в корпоративном контексте; потребность в действиях во внешних системах; требования к объяснимости и аудируемости; зрелость enterprise-контура.

Отсеивающие критерии

На первом этапе применяются критерии, которые не ранжируют архитектуры, а исключают заведомо неподходящие варианты: высокая цена ошибки, критическая чувствительность данных, обязательная объяснимость, необходимость журналирования, запрет автономных действий без подтверждения человека и отсутствие допустимой инфраструктуры развертывания.

Ранжирующие критерии и шкала оценки

После применения отсеивающих критериев допустимые архитектуры сопоставляются по интегральной шкале. На этом этапе метод уже не отвечает на вопрос, можно ли использовать тот или иной класс решений в принципе, а показывает, какая из допустимых альтернатив лучше соответствует условиям задачи.

Оценка строится по шести критериям, позволяющим связать архитектурное решение с бизнес-сценарием, корпоративным контуром и требованиями к контролю результата.

C_1 – соответствие типу задачи. Критерий отражает, насколько архитектура пригодна для справочных, аналитических, рекомендательных и многошаговых процессов.

C_2 – работа с корпоративным контекстом. Здесь учитываются глубина использования внутренних данных, плотность связей между фрагментами знаний, потребность в multi-hop retrieval и риск кросс-языковой потери контекста при смешанных корпусах.

C_3 – работа с актуальными данными. Критерий показывает, требуется ли архитектуре доступ к оперативно обновляемым источникам, внешним сервисам и транзакционным системам.

C_4 – соответствие требуемому уровню автономности. Он отражает, нужно ли решение только для ответа на запрос или для оркестрации последовательности действий и выбора промежуточных шагов.

C_5 – соответствие требованиям explainability и audit. Здесь оцениваются объяснимость результата, прозрачность цепочки вывода, возможность проверки источников и требования к provenance.

C_6 – реализуемость в организационно-технологическом контуре предприятия. При оценке учитываются зрелость интеграций, качество данных, стоимость внедрения и готовность поддерживать более сложные механизмы retrieval, включая graph-based processing [14].

Интегральная оценка архитектуры определяется формулой

$$S(A) = \sum(w_i \cdot C_i), \quad (1)$$

где $S(A)$ – интегральная оценка архитектуры, C_i – значение соответствующего критерия, а w_i – его относительная значимость.

Весовая корректировка

Весовая корректировка вводится для того, чтобы одна и та же шкала могла применяться в задачах с разной ценой ошибки и разным профилем ограничений.

Итоговая оценка допустимой архитектуры в этом случае рассчитывается по формуле

$$S(A) = \sum(w_i \cdot C_i), \quad (2)$$

где $S(A)$ – интегральная оценка архитектуры A , C_i – значение критерия, а w_i – его относительная значимость в конкретном классе задач.

Такой механизм позволяет смещать приоритеты без изменения самой системы критериев. Для справочных сценариев обычно усиливается значение критериев C_2 и C_3 . Для высокорисковых управленческих решений возрастает роль C_5 . Если задача связана с плотной сетью ссылок, расследованием инцидентов или traversal по связанным документам, дополнительно усиливаются C_2 и C_6 , поскольку graph-based retrieval требует иной зрелости данных и инфраструктуры [14].

Определение весов критериев

В предлагаемом методе веса критериев не задаются раз и навсегда, а определяются под конкретный класс задач и состояние корпоративной среды.

Практическое определение весов целесообразно строить как экспертную процедуру с участием архитекторов корпоративных ИС, аналитиков бизнес-процессов и специалистов по данным.

На первом этапе формируется перечень сценариев, для которых метод будет применяться, после чего эксперты выполняют попарные сравнения критериев применительно к выбранному сценарию.

Индивидуальные матрицы сравнений агрегируются, после чего вычисляется нормированный вектор весов, удовлетворяющий условию формулы

$$\sum_{i=1}^n w_i = 1. \quad (3)$$

Полученный вектор интерпретируется как профиль приоритетов для конкретного сценария выбора архитектуры.

Далее веса могут применяться повторно для близких задач, однако при изменении структуры данных, регуляторных требований или глубины knowledge graph их следует пересматривать [14].

Для контроля устойчивости экспертных суждений целесообразно использовать индекс и отношение согласованности; при заметных расхождениях процедура повторяется.

Тем самым метод не навязывает универсального набора приоритетов, а адаптирует ранжирование архитектур к условиям предприятия.

Таблица 2

Критерии метода выбора архитектуры

Table 2

Criteria of the Architecture Selection Method

Критерий	Тип	Значения	Влияние
Цена ошибки	Отсев / вес	Низкая / средняя / высокая	Исключает isolated LLM и часть agent-based решений
Чувствительность данных	Отсев	Низкая / высокая	Задаёт требования к закрытому контуру
Корпоративный контекст	Ранг	Нет / умеренный / высокий	Усиливает RAG и hybrid
Действия в системах	Ранг	Нет / ограничено / да	Усиливает tools и agents
Explainability / audit	Отсев / ранг	Низкое / высокое	Усиливает hybrid
Автономность	Ранг	Низкая / средняя / высокая	Влияет на выбор между hybrid и agents

Табл. 2 показывает, что критерии метода работают как единая система: часть из них отсекает архитектурно неподходящие решения, а часть позволяет различить допустимые альтернативы по качеству соответствия задаче.

При этом признаки, характерные для graph-based retrieval, не образуют отдельного класса критериев, а уточняют оценку корпоративного контекста, explainability и реализуемости, когда задача требует traversal по связанным документам и provenance по цепочке источников [14].

Правило выбора архитектуры

На первом шаге формируется множество допустимых архитектур A^* после применения критических ограничений. Затем каждая из оставшихся альтернатив получает интегральную оценку $S(A)$ по выбранным критериям и весам. Архитектура с наибольшим значением рассматривается как целевая. Если для ее внедрения недостаточны зрелость данных, глубина интеграции или средства контроля, метод допускает выбор переходного варианта с последующим усложнением.

Псевдоалгоритм выбора архитектуры

Шаг 1 – определить тип задачи, ожидаемый формат результата и цену ошибки.

Шаг 2 – зафиксировать критические ограничения по данным, безопасности, explainability и степени допустимой автономности.

Шаг 3 – установить, нужен ли задаче корпоративный контекст, доступ к актуальным системам и multi-hop retrieval по связанным знаниям.

Шаг 4 – исключить архитектуры, которые не удовлетворяют обязательным ограничениям.

Шаг 5 – оценить допустимые архитектуры по критериям C_1 – C_6 .

Шаг 6 – назначить веса критериев с учетом сценария применения и зрелости корпоративного контура.

Шаг 7 – рассчитать интегральные оценки и упорядочить альтернативы.

Шаг 8 – проверить реализуемость лидирующей архитектуры, включая качество данных, интеграции, provenance и эксплуатационные ограничения.

Шаг 9 – выбрать целевую архитектуру и, при необходимости, зафиксировать переходный вариант внедрения.

Интерпретация результатов выбора

Предлагаемый метод не сводится к механическому выбору единственной архитектуры. В реальном enterprise-проекте он позволяет развести целевой и переходный варианты, если зрелость данных, глубина интеграции и механизмы контроля развиваются не одновременно.

Именно поэтому результат интерпретируется не только как максимум интегральной оценки, но и как аргументированная рекомендация по конфигурации решения: где достаточно классического RAG, где нужен tool-augmented LLM, а где оправдано подключение agent-based subsystem или hybrid controlled architecture.

Пример применения метода на типовых сценариях торгово-производственного предприятия

Для проверки практической применимости предложенного метода рассмотрим его использование на типовых сценариях торгово-производственного предприятия, в информационном контуре которого функционируют учетная система, CRM, внутренняя база регламентов, документы по закупкам и логистике, аналитическая отчетность и сервисы обработки заявок.

Сценарий 1. Доступ к внутренним регламентам и инструкциям.

По отсеивающим критериям из числа допустимых решений исключаются варианты, не обеспечивающие устойчивой привязки к корпоративным источникам. На этапе ранжирования наибольший вес получают критерии корпоративного контекста и объяснимости. В результате базовой архитектурой остается RAG, однако при большом числе перекрестных ссылок между документами, версиями и связанными сущностями предпочтителен graph-based RAG, поскольку он лучше поддерживает multi-hop retrieval, удерживает структуру связей и делает путь к источнику проверяемым [14].

Сценарий 2. Подготовка управленческой сводки для руководителя.

На этапе предварительного отбора исключаются архитектурные варианты, не обеспечивающие взаимодействие с внешними корпоративными сервисами. При последующем ранжировании наибольшую значимость приобретают критерии интеграции с информационными системами предприятия, актуальности используемых данных и объяснимости результатов. В этих условиях в качестве целевой архитектуры рассматривается tool-augmented LLM, тогда как при повышенных требованиях к журналированию, контролю и управляемости альтернативой может выступать hybrid controlled architecture.

Сценарий 3. Рекомендация менеджеру по типовой заявке клиента.

Критическими оказываются цена ошибки, требование к explainability и необходимость сохранения контроля со стороны человека. На этапе ранжирования наибольший вес получают критерии корпоративного контекста, explainability и audit. В результате целевой архитектурой становится hybrid controlled architecture; если зрелость контура недостаточна, переходным вариантом может выступать tool-augmented LLM.

Сценарий 4. Многошаговая обработка внутреннего инцидента.

На этапе ранжирования максимальный вес получают критерии автономности, интеграции и orchestration. Целевой архитектурой становится agent-based subsystem; при повышенной цене ошибки и жестких требованиях к аудиту более предпочтительной становится гибридная конфигурация. Graph-based retrieval здесь полезен не как самостоятельный класс, а как поддерживающий слой, который связывает события, документы, связанные инциденты и регламенты в единую трассируемую цепочку.

Аналитическое представление результатов апробации

Проведенная апробация показывает, что разные типы задач закономерно приводят к различным архитектурным решениям. Справочно-информационные процессы тяготеют к retrieval-ориентированным конфигурациям, аналитические сценарии, связанные с данными нескольких систем, – к tool-augmented LLM, а управленческие и многошаговые процессы – к агентным и гибридным конфигурациям. Добавление аспектов graph-based retrieval уточняет границу между случаями, где достаточно обычного RAG, и сценариями, в которых критичны связи между документами, provenance и кросс-языковой риск потери контекста [14].

Таблица 3

Результаты применения метода к типовым сценариям

Table 3

Results of Applying the Method to Typical Scenarios

Сценарий	Ограничения	Действия в системах	Целевая архитектура	Обоснование
Доступ к регламентам	Проверяемость, актуальность, связи	Нет	RAG / graph-based RAG	Нужен доступ к актуальным источникам и цепочкам ссылок
Аналитическая сводка	Высокая цена ошибки	Ограниченно	Tool-augmented LLM	Нужны актуальные данные CRM/ERP
Рекомендация менеджеру	Контроль человека	Ограниченно	Hybrid controlled	Нужны правила и human-in-the-loop
Обработка инцидента	Многошаговая логика	Да	Agent-based / hybrid	Нужны orchestration, контроль и связный контекст инцидента

Согласно табл. 3, классический RAG и его graph-based вариант естественным образом применимы в сценариях доступа к регламентам и поиска знаний, тогда как tool-augmented LLM предпочтителен для аналитических сводок, agent-based subsystem – для инцидентов, а hybrid controlled architecture – для рекомендаций руководителю.

Тем самым апробация подтверждает, что Graph RAG не образует самостоятельный архитектурный класс вне общей схемы исследования, а выступает специализированным профилем retrieval-подхода для задач со связным знанием, multi-hop retrieval и жесткими требованиями к provenance.

Для enterprise-среды это означает, что выбор retrieval-механизма следует оценивать не только по полноте ответа, но и по способности восстановить цепочку источников и связанных объектов без потери контекста при переходе между документами.

Заключение

В статье рассмотрена задача выбора архитектуры интеллектуальной подсистемы поддержки принятия решений в корпоративной информационной среде. Показано, что корректный выбор требует учета не только класса задачи, но и структуры корпоративного знания, глубины интеграции, цены ошибки и требований к контролируемости результата.

В ходе исследования выделены пять базовых архитектурных классов: базовая LLM-подсистема, подсистема класса RAG, tool-augmented LLM, agent-based subsystem и hybrid controlled architecture. Одновременно показано, что graph-based RAG следует рассматривать как специализированный вариант retrieval-подхода внутри класса RAG, а не как отдельный шестой класс [14].

Основным результатом исследования является разработка метода выбора архитектуры, сочетающего отсеивающие критерии, ранжирование по интегральной шкале и весовую корректировку. В отличие от упрощенных схем выбора, метод учитывает explainability, audit, provenance, потребность в multi-hop retrieval и риск кросс-языковой потери контекста.

Рассмотренные сценарии показывают, что для задач доступа к регламентам достаточно RAG или graph-based RAG, для аналитической сводки оправдан tool-augmented LLM, для обработки инцидентов требуется agent-based subsystem с возможной graph-based поддержкой, а для подготовки рекомендаций руководителю предпочтительна hybrid controlled architecture.

Список литературы

1. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis, E. Perez, A. Piktus et al. // ArXiv preprint: arXiv:2005.11401. 2020.
2. A Survey on Large Language Model Based Autonomous Agents / L. Wang, C. Ma, X. Feng et al. // ArXiv preprint: arXiv:2308.11432. 2023.
3. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions / L. Huang, W. Yu, W. Ma et al. // ArXiv preprint: arXiv:2311.05232. 2023.
4. Добренко Н.В., Добренко Д.А., Улизько М.В. Интеллектуальная поддержка принятия управленческих решений в MES-системах с использованием больших языковых моделей // Экономика. Право. Инновации. 2024. № 3. С. 47–59. DOI: 10.17586/2713-1874-2024-3-47-59
5. Бурый А.С., Цаплина О.С. Агентные модели принятия решений на основе генеративного искусственного интеллекта // Правовая информатика. 2025. № 1. С. 101–109. DOI: 10.24412/1994-1404-2025-1-101-109
6. Синякова Е.А. Стратегии внедрения ИИ в редакциях местных газет: типология и организационный контекст // Вестник Московского университета. Серия 10. Журналистика. 2025. № 6. С. 80–113. DOI: 10.55959/msu.vestnik.journ.6.2025.80113
7. Использование технологий искусственного интеллекта в российских медиа и журналистике / С.Г. Давыдов, А.В. Замков, М.А. Крашенинникова, М.М. Лукина // Вестник Московского университета. Серия 10. Журналистика. 2023. № 5. С. 3–21. DOI: 10.30547/vestnik.journ.5.2023.321
8. Зуйкина К.Л., Разумова Д.В. Технологии искусственного интеллекта в работе российских СМИ: инструменты, проблемы и угрозы // Вестник Московского университета. Серия 10. Журналистика. 2025. № 5. С. 79–103. DOI: 10.55959/msu.vestnik.journ.5.2025.79103
9. Velasquez M., Hester P.T. An Analysis of Multi-Criteria Decision Making Methods // International Journal of Operations Research. 2013. Vol. 10, no. 2. P. 56–66.
10. Ishizaka A., Nemery P. Multi-Criteria Decision Analysis: Methods and Software. Chichester: John Wiley & Sons, 2013. DOI: 10.1002/9781118644898
11. Saaty T.L. Decision Making with the Analytic Hierarchy Process // International Journal of Services Sciences. 2008. Vol. 1, no. 1. P. 83–98. DOI: 10.1504/IJSSCI.2008.017590
12. Tyree J., Akerman A. Architecture Decisions: Demystifying Architecture // IEEE Software. 2005. Vol. 22, no. 2. P. 19–27. DOI: 10.1109/MS.2005.27
13. Jansen A., Bosch J. Software Architecture as a Set of Architectural Design Decisions // Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture. 2005. P. 109–120. DOI: 10.1109/WICSA.2005.61
14. VladSpace. Как я построил Graph RAG систему с точностью 96.7% за 5 дней: от научных статей до production-ready пайплайна // Хабр. 24.02.2026. URL: <https://habr.com/ru/articles/1003064/> (дата обращения: 23.04.2026).
15. Toolformer: Language Models Can Teach Themselves to Use Tools / T. Schick, J. Dwivedi-Yu, R. Dessi et al. // ArXiv preprint: arXiv:2302.04761. 2023.
16. ReAct: Synergizing Reasoning and Acting in Language Models / S. Yao, J. Zhao, D. Yu et al. // ArXiv preprint: arXiv:2210.03629. 2022.
17. Holistic Evaluation of Language Models / P. Liang, R. Bommasani, T. Lee et al. // ArXiv preprint: arXiv:2211.09110. 2022.

References

1. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W.-t., Rocktäschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *ArXiv preprint: arXiv:2005.11401*. 2020.
2. Wang L., Ma C., Feng X., Zhang Z., Yang H., Zhang J., Chen Z., Tang J., Chen X., Lin Y., Zhao W.X., Wei Z., Wen J.-R. A Survey on Large Language Model Based Autonomous Agents. *ArXiv preprint: arXiv:2308.11432*. 2023.
3. Huang L., Yu W., Ma W., Zhong W., Feng Z., Wang H., Chen Q., Peng W., Feng X., Qin B., Liu T. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ArXiv preprint: arXiv:2311.05232*. 2023.
4. Dobrenko N.V., Dobrenko D.A., Ulizko M.V. Intelligent management decision support in MES systems using large language models. *Economics. Law. Innovations*. 2024;(3):47–59. (In Russ.) DOI: 10.17586/2713-1874-2024-3-47-59

5. Buryi A.S., Tsaplina O.S. Agent-based models for decision-making using generative artificial intelligence. *Legal informatics*. 2025;(1):101–109. (In Russ.) DOI: 10.24412/1994-1404-2025-1-101-109
6. Sinyakova E.A. Strategies for AI implementation in local newsrooms: typology and organizational context. *Vestnik Moskovskogo Universiteta. Seriya 10. Zhurnalistika = Lomonosov Journalism Journal*. 2025;(6):80–113. (In Russ.) DOI: 10.55959/msu.vestnik.journ.6.2025.80113
7. Davydov S.G., Zamkov A.V., Krasheninnikova M.A., Lukina M.M. The use of artificial intelligence technologies in Russian media and journalism. *Vestnik Moskovskogo Universiteta. Seriya 10. Zhurnalistika = Lomonosov Journalism Journal*. 2023;(5):3–21. (In Russ.) DOI: 10.30547/vestnik.journ.5.2023.321
8. Zuykina K.L., Razumova D.V. Artificial intelligence in Russian mass media: instruments, problems, and threats. *Vestnik Moskovskogo Universiteta. Seriya 10. Zhurnalistika = Lomonosov Journalism Journal*. 2025;(5):79–103. (In Russ.) DOI: 10.55959/msu.vestnik.journ.5.2025.79103
9. Velasquez M., Hester P.T. An Analysis of Multi-Criteria Decision Making Methods. *International Journal of Operations Research*. 2013;10(2):56–66.
10. Ishizaka A., Nemery P. *Multi-Criteria Decision Analysis: Methods and Software*. Chichester, John Wiley & Sons, 2013. DOI: 10.1002/9781118644898
11. Saaty T.L. Decision Making with the Analytic Hierarchy Process. *International Journal of Services Sciences*. 2008;1(1):83–98. DOI: 10.1504/IJSSCI.2008.017590
12. Tyree J., Akerman A. Architecture Decisions: Demystifying Architecture. *IEEE Software*. 2005;22(2):19–27. DOI: 10.1109/MS.2005.27
13. Jansen A., Bosch J. Software Architecture as a Set of Architectural Design Decisions. In: *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture*. 2005. P. 109–120. DOI: 10.1109/WICSA.2005.61
14. VladSpace. *Kak ya postroil Graph RAG sistemu s tochnost'yu 96.7% za 5 dney: ot nauchnykh statey do production-ready payplayna* [How I Built a Graph RAG System with 96.7% Accuracy in 5 Days: From Research Papers to a Production-Ready Pipeline]. *Habr*. February 24, 2026. (In Russ.) Available at: <https://habr.com/ru/articles/1003064/> (accessed April 23, 2026).
15. Schick T., Dwivedi-Yu J., Dessi R., Raileanu R., Lomeli M., Zettlemoyer L., Cancedda N., Scialom T. Toolformer: Language Models Can Teach Themselves to Use Tools. *ArXiv preprint: arXiv:2302.04761*. 2023.
16. Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y. ReAct: Synergizing Reasoning and Acting in Language Models. *ArXiv preprint: arXiv:2210.03629*. 2022.
17. Liang P., Bommasani R., Lee T., Tsipras D., Soylu D., Yasunaga M., Zhang Y., Narayanan D., Wu Y., Kumar A. et al. Holistic Evaluation of Language Models. *ArXiv preprint: arXiv:2211.09110*. 2022.

Информация об авторах

Логиновский Олег Витальевич, д-р техн. наук, проф., проф. кафедры информационных систем и технологий, Южно-Уральский государственный университет, Челябинск, Россия; loginovskiiiov@susu.ru.

Голлай Александр Владимирович, д-р техн. наук, доц., заведующий кафедрой информационных систем и технологий, Южно-Уральский государственный университет, Челябинск, Россия; gollaiav@susu.ru.

Браженко Сергей Александрович, аспирант кафедры информационных систем и технологий, Южно-Уральский государственный университет, Челябинск, Россия; bsa@uralto.ru.

Information about the authors

Oleg V. Loginovskiy, Dr. Sci. (Eng.), Prof., Prof. of the Department of Information Systems and Technologies, South Ural State University, Chelyabinsk, Russia; loginovskiiiov@susu.ru.

Alexander V. Hollay, Dr. Sci. (Eng.), Ass. Prof., Head of the Department of Information Systems and Technologies, South Ural State University, Chelyabinsk, Russia; gollaiav@susu.ru.

Sergey A. Brazhenko, Postgraduate Student of the Department of Information Systems and Technologies, South Ural State University, Chelyabinsk, Russia; bsa@uralto.ru

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.

Авторы заявляют об отсутствии конфликта интересов.

Contribution of the authors: the authors contributed equally to this article.

The authors declare no conflicts of interests.

Статья поступила в редакцию 23.04.2026

The article was submitted 23.04.2026