УДК 004.8:004.62:004.056 DOI 10.21777/2500-2112-2026-2-80-89

МЕТОД АВТОМАТИЗИРОВАННОЙ ПОДГОТОВКИ ПРОЕКТНОЙ ДОКУМЕНТАЦИИ ПО ГОСТ НА ОСНОВЕ ГЕНЕРАТИВНОЙ МОДЕЛИ, ПОИСКА ПО ИСТОЧНИКАМ И ФОРМАЛЬНОЙ ВАЛИДАЦИИ

Рахмани Джахед[1],
e-mail: jahed@mail.ru

Курочкин Ярослав Александрович[1],
e-mail: yaroslav.kurochkin.2005@mail.ru
[1]Московский технический университет связи и информатики (МТУСИ), г. Москва, Россия

В статье представлен метод автоматизированной подготовки технического задания и сопутствующей документации на автоматизированные системы в соответствии с государственным стандартом 34. Предложенный метод объединяет генерацию текста с помощью больших языковых моделей, дополненную поиском информации в проверенных источниках, и модуль формальной валидации, включающий проверку полноты, непротиворечивости, трассируемости и покрытия требований защиты информации. Метод адаптирован под российскую нормативную базу и ориентирован на использование отечественных языковых моделей. В ходе эксперимента проведено сравнение четырёх режимов работы: ручная подготовка, генерация без поиска по источникам, с использованием поиска без валидатора и полный предлагаемый процесс. Результаты показывают сокращение времени подготовки черновика до 90 %, повышение полноты разделов до 98 % и существенное снижение числа типовых ошибок. Предложен протокол оценки качества технического задания и сопутствующей документации на автоматизированные системы, проведён анализ типовых ошибок. Метод позволяет снизить трудозатраты и риски, связанные с неполнотой и противоречивостью требований.

Ключевые слова: проектирование информационных систем, ГОСТ 34, техническое задание, LLM, RAG, защита информации, валидация, архитектура, требования безопасности

A METHOD FOR AUTOMATED PREPARATION OF GOST-COMPLIANT PROJECT DOCUMENTATION USING A GENERATIVE MODEL, SOURCE SEARCH, AND FORMAL VALIDATION

Rahmani J.[1],
e-mail: jahed@mail.ru

Kurochkin Y.A.[1],
e-mail: yaroslav.kurochkin.2005@mail.ru
[1]Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russia

The article presents a method for the automated preparation of technical specifications and related documentation for automated systems in accordance with state standard 34. The proposed method combines text generation using large language models, supplemented by information search in verified sources, and a formal validation module that includes verification of completeness, consistency, traceability, and coverage of information security requirements. The method is adapted to the Russian regulatory framework and is focused on the use of domestic language models. During the experiment, four modes of operation were compared: manual preparation, generation without source search, using search without a validator, and the full proposed process. The results show a reduction in draft preparation time by up to 90%, an increase in section completeness by up to 98%, and a significant reduction in the number of typical errors. A quality assessment protocol is proposed, and an

analysis of typical errors is provided. The method reduces labor costs and risks associated with incomplete and contradictory requirements.

Keywords: information systems design, GOST 34, terms of reference, LLM, RAG, information protection, validation, architecture, security requirements

Введение

Современные информационные системы (ИС) становятся сложнее, они всё теснее интегрируются в бизнес-процессы и часто являются критически важными для работы организаций. В России разработка таких систем должна строго соответствовать национальным стандартам, особенно в государственном секторе и на объектах критической информационной инфраструктуры (КИИ), где действуют требования федеральных законов о персональных данных, безопасности КИИ и об информации. Один из самых важных этапов – написание технического задания (ТЗ) по ГОСТ 34.602-2020[1], которое задаёт требования к функциям, архитектуре, надёжности и безопасности будущей системы. Однако процесс разработки ТЗ остаётся трудоёмким, требует участия высококвалифицированных специалистов и подвержен ошибкам: неполнота разделов, противоречивость требований, упущение обязательных мер защиты информации.

Современные большие языковые модели (LLM) умеют генерировать связные тексты, но при попытке создать нормативный документ могут «галлюцинировать» [1, с. 1906] – придумывать факты или отклоняться от стандартов. Чтобы этого избежать, предлагается использовать подход Retrieval-Augmented Generation (RAG) [2, с. 9460], дополняющий генерацию поиском в проверенных источниках. Также важна формальная валидация сгенерированного текста на соответствие ГОСТ и требованиям безопасности.

Помимо формальных требований стандартов, остро стоит проблема безопасности при использовании самих LLM. По данным исследования, проведённого в начале 2026 года одной из российских компаний в области кибербезопасности, за последний год объём корпоративной информации, отправляемой сотрудниками российских компаний в публичные ИИ-сервисы (ChatGPT, Google Gemini и др.), вырос более чем в 30 раз[2]. Около 60 % организаций не имеют формализованных политик регулирования работы с ИИ, что создаёт критический риск утечек конфиденциальных данных, включая исходные коды, финансовую отчётность и клиентские базы. В этих условиях предложенный метод, использующий локально развёртываемые отечественные LLM (например, GigaChat) и строгую валидацию выходных документов, позволяет не только автоматизировать проектирование, но и снизить риски, связанные с передачей чувствительной информации во внешние ИИ-сервисы.

Несмотря на развитие методов генерации текста с помощью LLM, задача создания документов, строго соответствующих формальным стандартам (ГОСТ, отраслевые нормы), остаётся нерешённой. Требуется не только генерация связного текста, но и обеспечение доказуемой полноты, трассируемости требований к источникам [3, с. 96], непротиворечивости [4, с. 1] и включения обязательных мер защиты информации. Существующие подходы к автоматизации документирования либо не используют LLM, либо не интегрируют валидацию по формальным критериям.

Цель работы – разработка и экспериментальная проверка метода автоматизированного формирования проектной документации по ГОСТ 34 с использованием LLM и RAG, дополненного модулем формальной валидации, учитывающим требования защиты информации.

Вклад работы состоит в формализации такого пайплайна, разработке модуля валидации из четырёх детекторов, предложении протокола оценки с четырьмя базовыми линиями и экспериментальном подтверждении эффективности подхода.

---

[1] ГОСТ 34.602-2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – Москва: Стандартинформ, 2020.

[2] Исследование ГК «Солар»: почти две трети компаний не контролируют утечки данных через нейросети // Хабр. – 2026. – 3 февраля. – URL: https://habr.com/ru/companies/solarsecurity/news/992548/ (дата обращения: 04.02.2026). – Текст: электронный.

1. Метод автоматизированной подготовки проектной документации по ГОСТ 34

1.1. Постановка задачи автоматизированного формирования технического задания по ГОСТ 34

Входные данные:

– текстовое описание автоматизируемой системы (свободная форма, концепция, протоколы совещаний);

– класс системы – определяется на основе области применения и требований регулирования. Для разных режимов регулирования используются различные классификационные параметры:

• для государственных информационных систем (ГИС) – класс защищённости (1, 2, 3) согласно приказу ФСТЭК № 17[3];

• для объектов критической информационной инфраструктуры (КИИ) – категория значимости (1, 2, 3) согласно Постановлению Правительства РФ № 127[4];

• для информационных систем персональных данных (ИСПДн) – уровень защищённости (УЗ-1…УЗ-4), определяемый категорией обрабатываемых данных и их объёмом в соответствии с постановлением №1119[5] и приказом ФСТЭК №21[6]. Выбор класса (категории/уровня) диктует набор обязательных требований по защите информации согласно нормативным документам;

– дополнительные ограничения (требование импортозамещения согласно ПП РФ № 1236[7], необходимость сертификации, целевая производительность).

Выходные данные:

– документ «Техническое задание» (ТЗ) по ГОСТ 34.602-2020 с полным набором разделов;

– описание архитектуры системы в текстовой форме с перечнем подсистем и их интерфейсов;

– отчёт модуля валидации, содержащий выявленные замечания;

– матрица трассируемости требований.

В таблице 1 приведены критерии, которым должны удовлетворять результаты работы пайплайна.

Таблица 1 – Артефакты и критерии их корректности

Артефакт	Критерии корректности
ТЗ по ГОСТ 34.602-2020	Наличие и содержательность всех разделов, обязательных согласно ГОСТ. Соответствие требований класса системы
Описание архитектуры	Полнота перечня подсистем и внешних интерфейсов. Непротиворечивость связей между подсистемами
Матрица трассируемости	Каждое функциональное требование должно быть связано с одной или несколькими подсистемами. Каждое требование к интерфейсу должно быть связано с участвующими в обмене подсистемами [3, с. 96]
Отчёт валидации	Для каждого замечания указан тип (пропуск, конфликт, отсутствие источника) и ссылка на фрагмент документа или требования

Цель оптимизации предлагаемого метода состоит в минимизации числа правок, вносимых экспертом после автоматической генерации документа, при обязательном соблюдении требований полноты, трассируемости, непротиворечивости и покрытия мер защиты информации. Формально это можно выразить в виде функции:

$$F = w_1 \cdot N_{missing} + w_2 \cdot N_{unsourced} + w_3 \cdot N_{conflicts} + w_4 \cdot N_{missing_security} \to min,$$

[3] Приказ ФСТЭК России от 11.02.2013 № 17 «Об утверждении Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах».

[4] Постановление Правительства РФ от 08.02.2018 № 127 «Об утверждении Правил категорирования объектов критической информационной инфраструктуры Российской Федерации, а также перечня показателей критериев значимости объектов критической информационной инфраструктуры Российской Федерации и их значений».

[5] Постановление Правительства Российской Федерации от 01.11.2012 № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных».

[6] Приказ ФСТЭК России от 18.02.2013 № 21 «Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных».

[7] Постановление Правительства Российской Федерации от 16.11.2015 № 1236 «Об установлении запрета на допуск программного обеспечения, происходящего из иностранных государств, для целей осуществления закупок для обеспечения государственных и муниципальных нужд».

где $N_{missing}$ – число пропущенных разделов;

$N_{unsourced}$ – число требований без источников;

$N_{conflicts}$ – число выявленных конфликтов;

$N_{missing_security}$ – число недостающих мер ИБ;

w_i – весовые коэффициенты, отражающие критичность каждого типа ошибок.

В данной работе веса приняты равными для упрощения, но могут быть изменены под конкретную организацию.

1.2. Архитектура предлагаемого LLM-пайплайна

Предлагаемый пайплайн состоит из трёх основных компонентов: корпуса нормативных и проектных знаний, RAG-процедуры генерации и модуля валидации. Ниже последовательно описаны структура и функции каждого компонента, их взаимодействие в процессе формирования итогового документа. На рисунке 1 представлена общая схема потока данных.

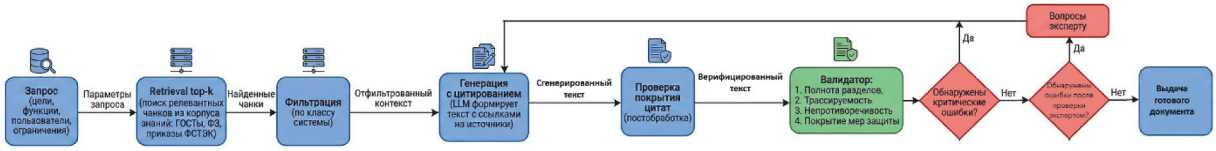

Рисунок 1 – Схема LLM-пайплайна для генерации и валидации документации[8]

1.2.1. Корпус знаний

Корпус знаний формируется из следующих источников:

– полные тексты ГОСТ 34.602-2020, ГОСТ 34.201-2020[9], ГОСТ 19.201-78[10], ГОСТ 34.601-90[11], определяющие стадии создания автоматизированных систем;

– выдержки из приказов ФСТЭК (№ 21, №239[12]) и методических документов;

– текст Федерального закона № 152-ФЗ[13];

– реестр отечественного программного обеспечения;

– обезличенные примеры реальных ТЗ и архитектурных описаний.

Процедура сборки и индексации состоит из следующих этапов:

1. Очистка и сегментация: документы очищаются от графических вставок и разбиваются на фрагменты текста (чанки) фиксированного размера с перекрытием; размер фрагмента составляет 512 токенов, перекрытие – 64 токена.

2. Дедупликация: исключаются дублирующиеся фрагменты на основе косинусной близости эмбеддингов (>0.95).

3. Контроль актуальности: для каждого фрагмента (чанка) сохраняется дата редакции документа.

4. Метаданные и индексация: для каждого чанка сохраняются метаданные: тип документа, раздел, дата редакции, класс систем, к которым он применим. Чанки индексируются с помощью эмбеддинговой модели multilingual-e5-large [4, с. 1]. Для поиска используется косинусная мера, параметр top-k = 10.

1.2.2. RAG-процедура генерации ТЗ

Процесс генерации ТЗ включает следующие шаги:

---

[8] Составлено авторами.

[9] ГОСТ 34.201-2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем. – Москва: Стандартинформ, 2020.

[10] ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. – Москва: Издательство стандартов, 1978.

[11] ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. – Москва: Стандартинформ, 2009.

[12] Приказ ФСТЭК России от 25.12.2017 № 239 «Об утверждении Требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации».

[13] О персональных данных: Федеральный закон от 27.07.2006 № 152-ФЗ.

1. Извлечение ключевых сущностей из входного описания с помощью LLM: цели, функции, пользователи, ограничения, требования к безопасности.

2. Retrieval: для каждого сформированного элемента запроса выполняется поиск top-k наиболее релевантных чанков в векторной базе данных с фильтрацией по классу системы [5, с. 6769].

3. Генерация с цитированием: LLM формирует текст раздела, включая найденные фрагменты в виде цитат или ссылок.

4. Постобработка: проверяется наличие всех включённых цитат; при их отсутствии генерация повторяется.

1.2.3. Модуль валидации

Модуль состоит из четырёх детекторов, работающих последовательно:

1. Детектор полноты разделов проверяет наличие всех обязательных разделов по ГОСТ 34.602-2020 и содержательность текста в них.

2. Детектор трассируемости для каждого требования проверяет наличие ссылки на источник (по регулярным выражениям и векторной близости).

3. Детектор непротиворечивости анализирует пары требований на предмет логических конфликтов. Используется русскоязычная NLI-модель на основе архитектуры трансформер (обученная на корпусе RuNLI). Конфликтом считается пара утверждений, для которой модель возвращает метку «противоречие» с уверенностью выше порога 0.85 [6, с. 637]. Дополнительно применяются правила, основанные на нормативной логике.

4. Детектор покрытия мер защиты информации на основе класса системы формирует чек-лист обязательных мер и проверяет их наличие в ТЗ.

После прохождения всех детекторов система автоматически исправляет очевидные пропуски (например, добавляет заголовки разделов), а для сложных случаев формирует список вопросов эксперту.

1.2.4. Параметры генерации и детали реализации

Для обеспечения воспроизводимости результатов ниже приведены ключевые параметры использованных моделей и процедур. В качестве основной LLM использовалась отечественная модель GigaChat3. Параметры генерации: температура = 0.3 (для снижения креативности и повышения точности следования стандарту), top_p = 0.9, max_tokens = 4096. Политика цитирования – обязательное включение ссылки на документ (например, «согласно ГОСТ 34.602-2020») для каждого сформулированного требования, основанного на извлеченном чанке.

В детекторе непротиворечивости для NLI-модели порог уверенности для классификации пары утверждений как «противоречие» установлен на уровне 0.85. Для RAG-поиска использовалась мультиязычная эмбеддинговая модель multilingual-e5-large [4, с. 1].

2. Эксперимент и протокол оценки

2.1. Кейсы

Для апробации выбраны три типа систем, различающихся классом, сложностью и требованиями к безопасности. Каждый кейс представляет реальный проект, по которому ранее было разработано ТЗ вручную.

Сценарий апробации A – Корпоративный портал организации

Внутренний портал организации с функционалом документооборота, календаря, личных кабинетов сотрудников, интеграцией с Active Directory. Класс: ИСПДн, уровень защищённости УЗ-3. Обязательные меры ИБ: идентификация/аутентификация, разграничение доступа по ролям, защита каналов связи (TLS), регистрация событий, антивирусная защита.

Сценарий апробации B – Хранилище данных (DWH)

Система для сбора, очистки и аналитической обработки данных из различных источников. Класс: аналитическая система с требованиями к целостности и конфиденциальности. Обязательные меры: шифрование при хранении и передаче, управление доступом к витринам данных, аудит действий пользователей.

Сценарий апробации C – SCADA-система для управления энергосетью

Автоматизированная система диспетчерского управления для распределительной сетевой компании. Класс: объект КИИ (категория значимости 1). Обязательные меры: резервирование каналов и серверов, двухфакторная аутентификация, защита промышленных протоколов, физическая защита контроллеров.

2.2. Режимы сравнения

Для оценки вклада каждого компонента пайплайна определены четыре режима:

B0 (ручная подготовка): ТЗ разрабатывается экспертами. Эксперты работают независимо, затем результаты усредняются.;

B1 (LLM без RAG): генерация ТЗ с помощью той же LLM (GigaChat3) без доступа к базе знаний, только на основе входного описания.

B2 (LLM + RAG): генерация с использованием RAG (база знаний доступна), но без модуля валидации.

B3 (LLM + RAG + валидатор): полный пайплайн, включая все детекторы и автоматические исправления.

2.3. Процедура разметки и сбора данных

Для каждого кейса и каждого режима выполняется следующее:

– генерация документа (для B1-B3) или подготовка вручную (B0). Время фиксируется;

– два независимых эксперта анализируют полученный документ и заполняют форму оценки, включающую:

• список присутствующих разделов;
• количество требований;
• количество требований с явными ссылками;
• количество требований, которые не могут быть подтверждены (по мнению эксперта);
• список выявленных противоречий;
• список недостающих мер защиты информации (по чек-листу);

– разногласия между экспертами разрешаются в ходе обсуждения;

– для режимов B1-B3 дополнительно фиксируется время, затраченное на доработку документа (редактирование, добавление недостающих разделов, исправление ошибок). Время генерации (T_gen) измеряется автоматически.

2.4. Метрики

Использовались ниже рассмотренные метрики [7, с. 37].

2.4.1. Полнота разделов (Completeness)

$$Completeness = \frac{M}{N},$$

где N – число обязательных разделов по ГОСТ 34.602-2020;

M – число разделов, содержащих содержательный текст (длина >200 символов, не шаблонная заглушка).

2.4.2. Трассируемость требований (Trace)

$$Trace = \frac{R_{cited}}{R_{total}},$$

где R_{total} – общее число требований в ТЗ (под требованием понимается отдельное утверждение, обычно пункт);

R_{cited} – число требований с корректной ссылкой на источник (ГОСТ, приказ, исходное описание). Ссылка считается корректной, если она указывает на реально существующий документ.

2.4.3. Доля неподтверждённых требований (Unsupported)

$$Unsupported = \frac{R_{unsupported}}{R_{total}},$$

где $R_{unsupported}$ – требования, для которых не найдено подтверждение в корпусе знаний или исходном описании, по мнению экспертов. Эта метрика отражает уровень «галлюцинаций».

2.4.4. Плотность противоречий (ContradictionRate)

$$ContradictionRate = \frac{C}{R_{total}},$$

где C – число выявленных конфликтов (по правилам или NLI), нормализованное на общее число требований.

2.4.5. Покрытие мер защиты информации (SecurityCoverage)

$$SecurityCoverage = \frac{M_{present}}{M_{total}},$$

где M_{total} – общее число обязательных мер для данного класса системы (из чек-листа);

$M_{present}$ – число мер, явно указанных в ТЗ (в любом разделе, но чаще в разделе требований к безопасности).

2.4.6. Экономия времени (Saving)

$$Saving = \frac{T_{manual} - T_{AI}}{T_{manual}},$$

где T_{manual} – время ручной подготовки (B0);

$T_{AI}=T_{gen}+T_{edits}$ – общее время при использовании автоматизации (генерация + доработка).

3. Результаты

3.1. Сводная таблица результатов

В таблице 2 приведены средние значения метрик для каждого кейса и каждого режима.

Таблица 2 – Сравнительные метрики для трёх кейсов[14]

Сценарий	Режим	Completeness	Trace	Unsupported	ContradictionRate	SecurityCoverage	T_gen (ч)	T_edits (ч)	T_total (ч)	Saving
A	B0	0.85	0.98	0.03	0.12	0.67	–	–	40	–
A	B1	0.45	0.0	0.52	0.08	0.33	0.5	12	12.5	0.69
A	B2	0.92	0.74	0.12	0.05	0.80	0.8	3	3.8	0.91
A	B3	0.98	0.96	0.04	0.03	0.93	1.0	1.5	2.5	0.94
B	B0	0.88	0.88	0.09	0.08	0.75	–	–	48	–
B	B1	0.50	0.0	0.48	0.05	0.40	0.6	14	14.6	0.70
B	B2	0.94	0.78	0.10	0.03	0.83	1.0	3.5	4.5	0.91
B	B3	0.99	0.97	0.03	0.01	0.94	1.2	1.8	3.0	0.94
C	B0	0.82	0.96	0.05	0.09	0.70	–	–	56	–
C	B1	0.40	0.0	0.60	0.07	0.28	0.7	18	18.7	0.67
C	B2	0.89	0.70	0.18	0.04	0.75	1.1	5	6.1	0.89
C	B3	0.97	0.95	0.05	0.02	0.91	1.3	2.2	3.5	0.94

Наблюдения:

Режим B1 (LLM без RAG) показывает низкую полноту (40–50 %) и очень высокую долю неподтверждённых требований (48–60 %), что подтверждает необходимость внешней базы знаний для генерации нормативно-технической документации. Добавление RAG (B2) резко улучшает все метрики, но остаются проблемы с противоречиями и покрытием ИБ-мер. Полный пайплайн с валидатором (B3) даёт наилучшие результаты: полнота >97 %, трассируемость >95 %, покрытие ИБ-мер >90 %, плот-

[14] Составлено авторами.

ность противоречий снижается до 1–2 на всё ТЗ. Время подготовки сокращается на 80–90 % по сравнению с ручной работой (с 40–56 часов до 2,5–3,5 часа).

3.2. Анализ типовых ошибок

В таблице 3 представлена классификация ошибок, выявленных в документах, сгенерированных в режимах B2 (до валидации) и B3 (после валидации и автоматической коррекции)

Таблица 3 – Частота типовых ошибок до и после валидации[15]

Тип ошибки	B2 (среднее)	B3 (среднее)	Снижение
Отсутствует обязательный раздел	0.8 на ТЗ	0.1 на ТЗ	87%
Размытые формулировки вместо конкретных требований	12 % требований	3 % требований	75 %
Импортозамещение vs зарубежный продукт	2.3 на ТЗ	0.4 на ТЗ	83 %
Производительность vs ресурсы	1.8 на ТЗ	0.5 на ТЗ	72 %
Отсутствует или неполная мера ИБ	3.2 на ТЗ	0.6 на ТЗ	81 %
Требование без источника	45 %	5 %	89 %
Терминологическая неоднозначность (использование разных терминов для одного понятия)	2.5 на ТЗ	0.8 на ТЗ	68 %

Валидатор наиболее эффективен для выявления пропущенных разделов и требований без источников, а также для обнаружения недостающих мер ИБ. Конфликты, связанные с тонкими нюансами производительности или совместимости, частично остаются и требуют экспертного вмешательства, но их число значительно сокращается.

3.3. Оценка качества работы детекторов валидатора

Для оценки вклада каждого компонента валидатора были проведены дополнительные замеры их точности. Результаты приведены в таблице 4.

Таблица 4 – Метрики качества детекторов модуля валидации[16]

Детектор	Метрика	Значение	Примечание
Полноты разделов	Точность	1.0	Не было ложных срабатываний
	Полнота	0.99	Пропущен 1 раздел из-за нестандартного заголовка
Трассируемости	Точность	0.96	Изредка ссылка на документ принималась за ссылку на требование
	Полнота	0.95	
Непротиворечивости (NLI)	Точность	0.88	Ложное обнаружение конфликта в сложных формулировках
	Полнота	0.75	25 % конфликтов пропущено (в основном семантические, а не логические)
Покрытия мер ИБ	Точность	1.0	
	Полнота	0.98	

Детекторы на основе правил (полнота разделов, покрытие ИБ) показали почти идеальную точность. Детектор на основе NLI, отвечающий за поиск сложных семантических противоречий, ожидаемо имеет более низкие показатели полноты, что требует дальнейшего улучшения модели или правил.

Заключение

В ходе исследования разработан и формализован метод автоматизированной подготовки проектной документации по ГОСТ 34, объединяющий генерацию на основе LLM, механизм RAG и формальную валидацию. Предложенный пайплайн включает сбор корпуса знаний, RAG-генерацию с обязательным цитированием источников и четыре детектора валидации: полноты разделов, трассируемости требований, непротиворечивости и покрытия мер защиты информации. Предложенный подход согла-

[15] Составлено авторами.
[16] Составлено авторами.

суется с современными тенденциями модульного проектирования информационных систем [8, с. 61] и учитывает ключевые проблемы управления сложными корпоративными системами [9, с. 1].

Экспериментальная проверка на трёх сценариях (корпоративный портал, хранилище данных, SCADA-система) показала, что полный пайплайн обеспечивает полноту разделов более 97 %, трассируемость более 95 % и покрытие обязательных мер защиты информации более 90 % при сокращении трудозатрат на 80–90 % по сравнению с ручной подготовкой. При этом валидатор позволяет снизить частоту типовых ошибок в 3–8 раз, особенно эффективно выявляя пропущенные разделы, требования без источников и недостающие меры безопасности.

Следует учитывать ограничения метода. Качество итоговых документов зависит от полноты и актуальности корпуса знаний, а также от детальности исходного описания системы. Для отраслей со специфическими требованиями (например, промышленная автоматизация, медицина) требуется адаптация корпуса и правил валидации. Кроме того, в текущей реализации автоматическая коррекция ограничена добавлением пропущенных разделов; более сложные исправления требуют участия эксперта.

Перспективы дальнейших исследований включают расширение корпуса знаний отраслевыми стандартами, разработку методов валидации архитектурных моделей (включая диаграммы), создание специализированных *fine-tuned* моделей для генерации документации, интеграцию с системами управления требованиями и жизненным циклом, а также полную автоматическую коррекцию выявленных ошибок без участия эксперта.

Предложенный метод может быть рекомендован для внедрения в организациях, разрабатывающих автоматизированные системы, особенно в государственном секторе и на объектах критической информационной инфраструктуры, где требования к документации наиболее жёстки.

Список литературы

1. *Maynez J., Narayan S., Bohnet B., McDonald R.* On Faithfulness and Factuality in Abstractive Summarization // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. – 2020. – P. 1906–1918.
2. *Lewis P. et al.* Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 9459–9474.
3. *Gotel O., Finkelstein A.* An Analysis of the Requirements Traceability Problem // Proceedings of the First International Conference on Requirements Engineering (ICRE). – 1994. – P. 94–101.
4. *Wang L., Yang N., Huang X., Jiao B., Yang L., Jiang D., Majumder R., Wei F.* Multilingual E5 Text Embeddings: A Technical Report // arXiv preprint arXiv:2402.05672. – 2024. – P. 1–6.
5. *Karpukhin V., Oguz B., Min S. et al.* Dense Passage Retrieval for Open-Domain Question Answering // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). – 2020. – P. 6769–6781.
6. *Bowman S.R., Angeli G., Potts C., Manning C.D.* A Large Annotated Corpus for Learning Natural Language Inference // Proceedings of EMNLP 2015. – 2015. – P. 632–642.
7. *Powers D.M.W.* Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation // Journal of Machine Learning Technologies. – 2011. – Vol. 2, No. 1. – P. 37–63.
8. *Рахмани Д., Дубровская А.А., Лобанова А.А.* Модульный подход в разработке информационных систем: преимущества и реализация // Труды ИСА РАН. – 2025. –Т. 75, № 3. – С. 61–67. – DOI 10.14357/20790279250306.
9. *Рахмани Д.* Ключевые проблемы управления корпоративной информационно-коммуникационной системой энергопроизводящей компании // I-methods. – 2025. – Т. 17, № 1. – С. 1–14.

References

1. *Maynez J., Narayan S., Bohnet B., McDonald R.* On Faithfulness and Factuality in Abstractive Summarization // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. – 2020. – P. 1906–1918.
2. *Lewis P. et al.* Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 9459–9474.

3. *Gotel O., Finkelstein A.* An Analysis of the Requirements Traceability Problem // Proceedings of the First International Conference on Requirements Engineering (ICRE). – 1994. – P. 94–101.
4. *Wang L., Yang N., Huang X., Jiao B., Yang L., Jiang D., Majumder R., Wei F.* Multilingual E5 Text Embeddings: A Technical Report // arXiv preprint arXiv:2402.05672. – 2024. – P. 1–6.
5. *Karpukhin V., Oguz B., Min S. et al.* Dense Passage Retrieval for Open-Domain Question Answering // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). – 2020. – P. 6769–6781.
6. *Bowman S.R., Angeli G., Potts C., Manning C.D.* A Large Annotated Corpus for Learning Natural Language Inference // Proceedings of EMNLP 2015. – 2015. – P. 632–642.
7. *Powers D.M.W.* Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation // Journal of Machine Learning Technologies. – 2011. – Vol. 2, No. 1. – P. 37–63.
8. *Rahmani D., Dubrovskaya A.A., Lobanova A.A.* Modul'nyj podhod v razrabotke informacionnyh sistem: preimushchestva i realizaciya // Trudy ISA RAN. – 2025. – T. 75, № 3. – S. 61–67. – DOI 10.14357/20790279250306.
9. *Rahmani D.* Klyuchevye problemy upravleniya korporativnoj informacionno-kommunikacionnoj sistemoj energoproizvodyashchej kompanii // I-methods. – 2025. – T. 17, № 1. – S. 1–14.

Статья поступила в редакцию: 25.03.2026
Статья принята к публикации: 15.04.2026

Received: 25.03.2026
Accepted: 15.04.2026