

УДК 004.912

DOI: [10.26102/2310-6018/2026.55.4.017](https://doi.org/10.26102/2310-6018/2026.55.4.017)

Адаптивный шаблон технического задания: графовая модель, структурный анализ и алгоритм автоматизированной верификации

А.В. Ечин¹, Н.Д. Алиева¹, А.М. Садыков¹, А.Г. Кравец^{1,2}, Е.В. Сафонова¹✉

¹Волгоградский государственный технический университет, Волгоград, Российская Федерация

²Государственный университет «Дубна», Дубна, Российская Федерация

Резюме. В условиях возрастающей неоднородности практик разработки программного обеспечения и применяемых стандартов подготовки документации, обеспечение полноты и структурной согласованности технического задания остается сложной и трудоемкой задачей. Существующие нормативные документы, включая ГОСТ 34, IEEE 830, ISO/IEC/IEEE 29148 и Volere, предлагают различные подходы к структурированию требований, однако их одновременное применение в реальных проектах приводит к дублированию разделов, несогласованности структуры и значительным затратам на ручную верификацию. В работе предлагается адаптивный шаблон технического задания, основанный на параметризованной графовой модели, позволяющей гибко структурировать требования в зависимости от типа программного обеспечения, применимых стандартов, отраслевых ограничений и уровня детализации. Разработан алгоритм структурного анализа и автоматизированной верификации документов в форматах DOCX и PDF на основе извлечения иерархии разделов и нечеткого сопоставления заголовков. Введена метрика адаптивности шаблона. Экспериментальная апробация на реальных технических заданиях показала достижение точности извлечения структуры до 92 % для документов DOCX. Предложенный подход может служить основой для интеллектуальных инструментов анализа технической документации.

Ключевые слова: техническое задание, графовая модель, адаптивность шаблона, нечеткое сопоставление, структурный анализ.

Благодарности: Исследование выполнено при поддержке «Центра цифровых научно-образовательных проектов и разработок в сфере промышленного искусственного интеллекта» Ц2RED-ИИ ВолгГТУ, созданного в рамках реализации образовательных программ топ-уровня в сфере искусственного интеллекта (Соглашение № 70-2025-000756).

Для цитирования: Ечин А.В., Алиева Н.Д., Садыков А.М., Кравец А.Г., Сафонова Е.В. Адаптивный шаблон технического задания: графовая модель, структурный анализ и алгоритм автоматизированной верификации. *Моделирование, оптимизация и информационные технологии*. 2026;14(4). URL: <https://moitvvt.ru/ru/journal/article?id=2261> DOI: 10.26102/2310-6018/2026.55.4.017

Adaptive technical specification template: graph-based model, structural analysis, and automated verification algorithm

A.V. Echin¹, N.D. Alieva¹, A.M. Sadykov¹, A.G. Kravets^{1,2}, E.V. Safonova¹✉

¹Volgograd State Technical University, Volgograd, the Russian Federation

²Dubna State University, Dubna, the Russian Federation

Abstract. In the context of increasing heterogeneity in software development practices and documentation standards, ensuring the completeness and structural consistency of technical specifications remains a complex and labor-intensive task. Existing regulatory frameworks, including

GOST 34, IEEE 830, ISO/IEC/IEEE 29148, and the Volere methodology, propose different approaches to structuring requirements; however, their simultaneous use in real-world projects often results in section duplication, structural inconsistency, and significant manual verification effort. This paper proposes an adaptive technical specification template based on a parameterized graph model that enables the formal integration of a mandatory regulatory core with flexibly connected extensions depending on the software type, industry-specific requirements, and the required level of detail. An automated structural verification algorithm for DOCX and PDF documents is developed, combining hierarchy extraction with fuzzy heading matching. Template adaptability metric has been introduced. Experimental validation on real-world technical specifications demonstrates structural extraction accuracy of up to 92 % for DOCX documents. The proposed approach can serve as a basis for intelligent tools for analyzing technical documentation.

Keywords: technical specifications, graph model, template adaptability, fuzzy matching, structural analysis.

Acknowledgements: The study was carried out with the support of the Center for Digital Scientific and Educational Projects and Developments in the Field of Industrial Artificial Intelligence (C2RED-AI) of Volgograd State Technical University, created as part of the implementation of top-level educational programs in the field of artificial intelligence (Agreement № 70-2025-000756).

For citation: Echin A.V., Alieva N.D., Sadykov A.M., Kravets A.G., Safonova E.V. Adaptive technical specification template: graph-based model, structural analysis, and automated verification algorithm. *Modeling, Optimization and Information Technology*. 2026;14(4). (In Russ.). URL: <https://moitvvt.ru/journal/article?id=2261> DOI: 10.26102/2310-6018/2026.55.4.017

Введение

Техническое задание (ТЗ) является ключевым документом жизненного цикла программного обеспечения, поскольку именно на его основе формируются архитектурные решения, планы разработки, процедуры тестирования и критерии приемки. Качество технического задания напрямую влияет на стоимость проекта, сроки реализации и уровень проектных рисков. В современных исследованиях по инженерии требований подчеркивается, что структурная полнота и формальная согласованность требований являются определяющими факторами качества программных систем [1, 2].

В современной практике разработки программных систем наблюдается одновременное использование нескольких нормативных и методических источников в рамках одного проекта, что нередко приводит к дублированию разделов, расхождениям в терминологии и различной глубине проработки требований. Отсутствие формализованной модели структуры документа приводит к высокой стоимости ручной валидации документации [3, 4].

Традиционные стандарты предоставляют полезные рамки для структурирования ТЗ. За последние годы было разработано множество стандартов, регулирующих содержание и структуру ТЗ [5, 6].

В российской практике широко применяется ГОСТ 34, ориентированный на автоматизированные системы, который закрепляет жесткую структуру с заранее определенным набором из девяти обязательных разделов: «Общие сведения», «Назначение и цели», «Характеристики объекта автоматизации», «Требования к системе (функциональные, производственные, надежности и др.)», «Объем и содержание работ по разработке», «Порядок контроля и приёмки», «Подготовительные работы по внедрению системы», «Требования к документации», «Ссылки и исходные документы». Каждый раздел включает дополнительные подразделы. Например, «Общие сведения» содержат наименования организаций, источники финансирования, юридические документы и графики работ. Документ отличается высокой формализацией и часто используется в государственных закупках и оборонных системах [7].

Международные стандарты IEEE 830 определяет спецификацию требований к программному обеспечению (SRS), состоящую из Введения («Назначение», «Область применения», «Определения и сокращения», «Ссылки», «Обзор»), Общего описания («Перспектива продукта», «Функции продукта», «Характеристики пользователей», «Ограничения и допущения») и Конкретных требований («Функциональные требования», «Производственные требования», «Логические требования к базе данных», «Внешние интерфейсы»). ISO/IEC/IEEE 29148 объединяет и обновляет несколько более ранних стандартов и различает высокоуровневые спецификации системных требований (SyRS) и детализированные спецификации требований к программному обеспечению (SRS). Методология Volere предназначена для гибких и итеративных сред. Помимо традиционных функциональных и нефункциональных требований, он включает: «Роли и ожидания заинтересованных сторон», «Юридические, политические и культурные ограничения», «Требования к удобству использования, сопровождаемости и переносимости», «Открытые проблемы, риски и отложенные требования», «Требования к обучению и документации». Volere акцентирует внимание на взаимодействии со стейкхолдерами и допускает адаптацию структуры документа в зависимости от контекста проекта [8, 9].

Дополнительную сложность вносит отраслевой фактор. В ряде областей техническое задание должно включать обязательные разделы, продиктованные нормативными и регуляторными требованиями. В медицинских информационных системах требуется формализация требований к безопасности данных и прослеживаемости изменений, в строительстве и промышленной автоматизации – требования к надежности и взаимодействию с физическими объектами, в финансовых системах – требования к аудиту и управлению рисками [10, 11].

В результате возникают структурные несоответствия, пропуски критически важных разделов и нарушения логики иерархии документа. Практика показывает, что значительная часть технических заданий формируется путем частичного заимствования фрагментов из различных стандартов или ранее разработанных документов [11, 12]. Кроме того, автоматизированная проверка соответствия практически отсутствует, что делает процесс аудита дорогостоящим и подверженным ошибкам [13, 14].

Таким образом, возникает противоречие между необходимостью строгого соблюдения нормативных требований и потребностью в гибкости структуры технического задания в зависимости от типа программного продукта и отрасли применения. Перспективным направлением является использование формальных графовых моделей для представления структуры требований, что позволяет обеспечить модульность, расширяемость и возможность алгоритмической проверки документа [15, 16].

Целью настоящей работы является разработка адаптивного формализованного шаблона технического задания, основанного на параметризованной графовой модели, позволяющей формально объединить требования различных стандартов и отраслевых регламентов и обеспечить автоматизированную проверку соответствия реальных документов заданной структуре. Предлагаемый подход сочетает структурную формализацию, алгоритм извлечения иерархии разделов из документов форматов DOCX и PDF и механизм нечеткого сопоставления заголовков.

Научная новизна работы заключается в интеграции нормативных и отраслевых требований в единую параметризованную графовую модель шаблона технического задания с возможностью количественной оценки его адаптивности и автоматизированной структурной верификации. Практическая значимость определяется применимостью разработанного подхода в задачах управления проектами.

Материалы и методы

Для преодоления структурной гетерогенности технических заданий предлагается формальная адаптивная модель шаблона, обеспечивающая объединение жесткого ядра и контекстно-зависимых расширений. Модель ориентирована на структурную верификацию документа и не предполагает на данном этапе анализа семантики требований, что позволяет четко разделить уровень структуры и уровень содержания.

Адаптивный шаблон технического задания T определяется как параметризуемая функция:

$$T = F(\mathcal{T}_p, \mathcal{D}, \mathcal{R}, c), \quad (1)$$

где $\mathcal{T}_p \in \{enterprise, consumer\}$ – тип программного обеспечения (корпоративные информационные системы, массовые программные продукты); $\mathcal{D} \subseteq \{GOST\ 34, IEEE\ 830, ISO\ 29148, Volere\}$ – выбранные стандарты; $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$ – отраслевые требования (например, безопасность, здравоохранение, финтех); $c \in [0,1]$ – параметр детализации, определяющий глубину декомпозиции разделов (от краткой рабочей версии ТЗ до формализованного документа для аудита).

Графовая модель шаблона технического задания. Структурно шаблон представляется в виде ориентированного мультиграфа:

$$T = (V_T, E_T), \quad (2)$$

где $V_T = \{v_1, v_2, \dots, v_k\}$ – множество вершин, соответствующих разделам или подразделам ТЗ; $E_T = \{(v_i, v_j, \tau_{ij})\}$ – множество ориентированных ребер с типами $\tau_{ij} \in \{structural, semantic, precedence\}$, указывающими на иерархические, семантические или последовательные зависимости.

Каждая вершина $v_i \in V_T$ описывается кортежем параметров:

$$v_i = \langle id, \mathcal{N}_i, m_i, w_i, f_i(c), P_i \rangle, \quad (3)$$

где id – уникальный идентификатор раздела; $\mathcal{N}_i = \{n_{i1}, n_{i2}, \dots, n_{ik}\}$ – множество допустимых заголовков или их альтернативных формулировок (например, «Назначение системы» | «Цели и область применения»); $m_i \in \{0,1\}$ – признак обязательности раздела (1, если раздел обязателен); $w_i \in [0,1]$ – вес критичности, отражающий значимость раздела для общей полноты документа; $f_i(c)$ – функция детализации, определяющая уровень раскрытия раздела в зависимости от параметра c (краткая, стандартная, расширенная); $P_i \subseteq \mathcal{D} \cup \mathcal{R}$ – множество стандартов или отраслевых тегов, определяющих включение раздела.

Раздел v_i включается в инстанцированный (заданный под конкретные условия) шаблон, если:

$$\phi(v_i, \mathcal{T}_p, \mathcal{D}, \mathcal{R}) = 1, \quad (4)$$

где ϕ – функция логического включения, проверяющая соответствие условий раздела выбранным параметрам шаблона.

Композиционно шаблон формируется как объединение нескольких подграфов.

– Core Layer V_{core} – включает обязательные разделы, соответствующие нормативному каркасу;

– Type-Specific Layer V_{type} – разделы, зависящие от типа ПО (\mathcal{T}_p), например, лицензирование и детали внедрения для потребительского ПО;

– Standards Layer V_D – дополнительные вершины из выбранных стандартов \mathcal{D} ;

– Domain Layer $V_{\mathcal{R}}$ – разделы, активируемые отраслевыми требованиями.
Окончательная структура формируется динамически:

$$V_T = V_{core} \cup V_{type} \cup V_D \cup V_{\mathcal{R}}. \quad (5)$$

Уровень детализации регулируется функцией $f_i(c)$, которая отображает значение параметра c в степень раскрытия структуры. При минимальном уровне детализации раздел может присутствовать только как заголовок первого уровня, при повышении параметра – автоматически активируются вложенные подразделы и уточняющие элементы. Тем самым обеспечивается масштабируемость шаблона.

$$f_i(c) = \begin{cases} \textit{summary}, & \text{при } 0 \leq c \leq 0,3 \\ \textit{standard}, & \text{при } 0,3 < c \leq 0,7. \\ \textit{extended}, & \text{при } 0,7 < c \leq 1 \end{cases} \quad (6)$$

Представленная модель обеспечивает формальную основу для автоматизированной структурной верификации технических заданий. Графовое представление позволяет сохранять модульность, объединять требования различных стандартов и поддерживать алгоритмическое сопоставление структуры шаблона с реальным документом.

Структурный анализ. Для автоматической проверки соответствия технического задания адаптивному шаблону предлагаются следующие шаги структурного анализа.

Шаг 1 – Прием входных данных. Осуществляется загрузка технического задания в формате DOCX или PDF и соответствующей конфигурации эталонного шаблона. Шаблон содержит множество разделов (8) с альтернативными вариантами заголовков и признаком обязательности m_i , что позволяет учитывать вариативность формулировок в реальных документах.

Набор спецификаций разделов:

$$S_T = \{s_1, s_2, \dots, s_k\}. \quad (8)$$

Каждая спецификация раздела s_i включает:

- $\mathcal{N}_i = \{n_{i1}, n_{i2}, \dots, n_{ik}\}$ – альтернативные допустимые заголовки;
- $m_i \in \{0,1\}$ – признак обязательности (1, если раздел обязателен);
- список ожидаемых подразделов (с их синонимами).

Шаг 2 – Извлечение и структурирование ТЗ. Для документа D в формате DOCX или PDF система извлекает его структурную иерархию, формируя дерево T_D , где каждая вершина n_j соответствует разделу или подразделу. Для DOCX алгоритм использует стили абзацев, схемы нумерации и форматирование шрифтов. Для PDF, где отсутствует семантическая разметка, алгоритм комбинирует стили текста, эвристики нумерации и позицию в макете для определения границ разделов.

Каждый узел v_j дерева документа характеризуется текстом заголовка, уровнем вложенности и ссылкой на родительский элемент. Таким образом обеспечивается унифицированное представление структуры независимо от исходного формата:

$$n_j = \langle \textit{title}(n_j), \textit{level}(n_j), \textit{pos}(n_j) \rangle. \quad (9)$$

Шаг 3 – Считывается список правил эталонного шаблона, определяющих, какие разделы и подразделы должны содержаться.

Шаг 4 – Сопоставление структуры ТЗ с шаблоном. Выполняется поиск соответствия между заголовком шаблона и разделом технической спецификации, фиксируется факт нахождения раздела. Для сравнения раздела документа n_j с разделом

шаблона s_i используется двухэтапная проверка. Сначала выполняется проверка лексического пересечения токенов заголовков, после чего применяется функция нечеткого сходства:

$$sim(s_i, n_j) = \max_{n \in \mathcal{N}} fuzzy(n, title(n_j)). \quad (10)$$

В качестве меры сходства используется токен-ориентированная метрика, устойчивая к перестановке слов. Совпадение фиксируется, если:

$$match(s_i, n_j) = \begin{cases} 1, & \text{если } sim(s_i, n_j) \geq \theta \\ 0, & \text{иначе} \end{cases}. \quad (11)$$

где θ – порог сходства, эмпирически выбранный равным 0,85.

Процесс повторяется рекурсивно для всех ожидаемых подразделов s_i .

Шаг 5 – Анализ подразделов. Выполняется рекурсивная проверка вложенных подразделов. Это позволяет оценить не только наличие основного раздела, но и полноту его внутренней структуры в соответствии с шаблоном.

Шаг 6 – Вычисление метрик.

Пусть:

- M – количество обязательных разделов;
- M_{found} – число найденных обязательных разделов;
- S_{total} – общее количество ожидаемых подразделов;
- $S_{present}$ – число совпавших подразделов.

Тогда:

Наличие обязательных разделов

$$P_s = \frac{M_{found}}{M}. \quad (12)$$

Наличие подразделов

$$P_{sub} = \frac{S_{present}}{S_{total}}, \quad (13)$$

$$\text{Отсутствующие разделы} = M - M_{found}. \quad (14)$$

Эти значения округляются до одной десятичной и используются для диагностической оценки.

Шаг 7 – Формирование детализированного отчета, включающего:

- итоговую оценку соответствия (если процент наличия обязательных разделов и подразделов $\geq 80\%$ – спецификация считается соответствующей; если $< 80\%$, но $\geq 50\%$ – частично соответствующей; если $< 50\%$ – несоответствующей);
- значения метрик соответствия;
- детализированное разбиение по каждому разделу шаблона, которое представлено в виде блок-схемы на Рисунке 1.

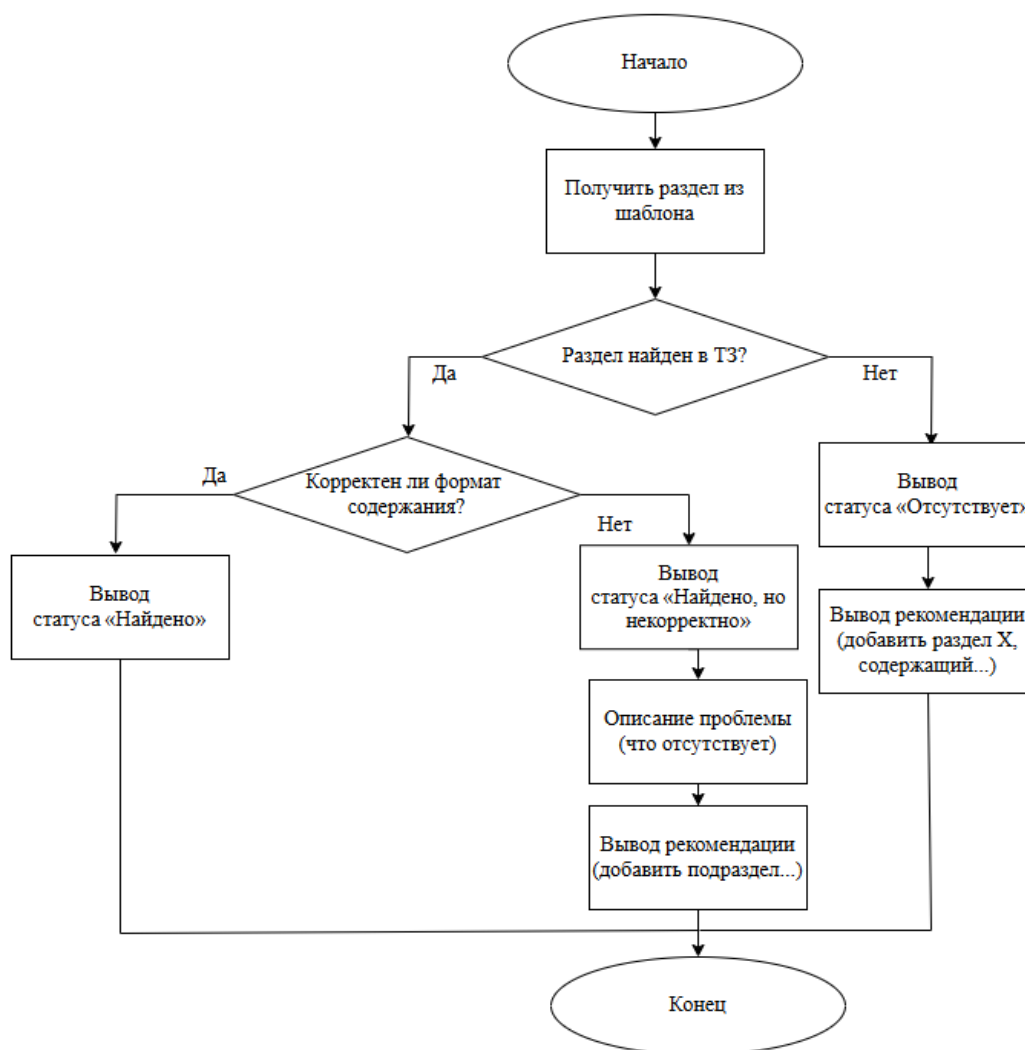


Рисунок 1 – Блок-схема детализированного разбиения по каждому разделу шаблона
 Figure 1 – Detailed decomposition flowchart for each template section

Предложенный алгоритм позволяет производить автоматизированную верификацию, сохраняя независимость от конкретного стандарта, с любой конфигурацией шаблона, сформированной на основе графовой модели.

Результаты

Для оценки работоспособности предложенной графовой модели шаблона технического задания и алгоритма структурной верификации была проведена экспериментальная апробация на корпусе реальных документов. В исследование включены пять технических заданий, различающихся по формату представления и степени структурной формализации. Четыре документа представлены в формате DOCX с использованием стандартных стилей заголовков и иерархической нумерации, один документ – в формате PDF с частично нарушенной семантической разметкой.

Для обеспечения сопоставимости результатов использовалась единая конфигурация адаптивного шаблона, сформированная для корпоративной информационной системы $\mathcal{T}_p \in \{enterprise\}$ с применением нормативного ядра и расширений, соответствующих международным стандартам спецификации требований $D \subseteq \{ГОСТ\ 34, IEEE\ 830\}$. Уровень детализации был установлен на среднем значении

$c = 0,7$, что обеспечивало включение как обязательных разделов верхнего уровня, так и типовых подразделов. Порог нечеткого сопоставления θ был установлен равным 0,85.

В ходе тестирования скрипта были проанализированы пять документов форматов DOCX и PDF. Для каждого документа строилось дерево узлов («Раздел», «Подраздел») и формировался итоговый отчет о соответствии шаблону. На Рисунке 2 приведен пример итогового отчета документа 4 формата DOCX.

```

ИТОГОВЫЙ ОТЧЁТ
=====
Общий вывод: Соответствует

Метрики соответствия:
| Показатель | Значение |
|-----|-----|
| Процент разделов | 100.0 |
| Процент подразделов | 91.7 |
| Отсутствующих разделов | 0 |

Детали по разделам:
1. Раздел «Общие сведения» – найден, но некорректен
   - Отсутствующие подразделы:
     * Порядок предоставления отчетных документов
     - Рекомендуется добавить указанные подразделы.
2. Раздел «Назначение и цели» – найден
3. Раздел «Характеристика объекта» – найден
4. Раздел «Требования к работам» – найден
5. Раздел «Состав и содержание работ по созданию» – найден
6. Раздел «Порядок контроля и приемки» – найден
7. Раздел «Требования к составу и содержанию работ» – найден
8. Раздел «Требования к документированию» – найден
9. Раздел «Источники разработки» – найден
    
```

Рисунок 2 – Итоговый отчет документа 4
Figure 2 – The final report for document 4

Результаты, представленные в Таблице 1, показали, что для документов формата DOCX алгоритм демонстрирует высокую точность извлечения структуры. В документе с корректно оформленными стилями заголовков полнота сопоставления обязательных разделов достигала 100 %, а полнота сопоставления подразделов превышала 90 %.

Таблица 1 – Результаты оценки пяти технических спецификаций
Table 1 – Evaluation results of five technical specifications

Документ	Формат	Вердикт соответствия	Наличие обязательных разделов (%)	Наличие подразделов (%)	Отсутствующие разделы
Doc 1	DOCX	Частично	66,7	83,3	3
Doc 2	DOCX	Частично	66,7	91,7	3
Doc 3	DOCX	Несоответствующий	11,1	0,0	8
Doc 4	DOCX	Соответствующий	100	91,7	0
Doc 5	PDF	Несоответствующий	11,1	0,0	8

Документы, сформированные без строгого следования нормативной структуре, демонстрировали значительно более низкие показатели соответствия. В одном из случаев отсутствовало большинство обязательных разделов нормативного ядра, что привело к минимальному значению коэффициента структурной полноты. Тем не менее алгоритм корректно идентифицировал фактически присутствующие разделы, что подтверждает устойчивость метода к неполной или нестандартной структуре документа.

Наиболее сложным для обработки оказался документ в формате PDF. Отсутствие явной семантической разметки, использование сканированных фрагментов и неоднородное форматирование привели к снижению точности извлечения иерархии.

Обсуждение и заключение

Верификация показала, что предложенная графовая модель шаблона технического задания и структурный анализ обеспечивают оценку соответствия документа заданной конфигурации. Наибольшая точность достигнута при обработке документов с корректной структурной разметкой, что подтверждает применимость метода в условиях регламентированной разработки. Использование нечеткого сопоставления заголовков позволило сохранить корректность анализа при вариативности формулировок. Ограничения подхода связаны преимущественно с обработкой PDF-документов, не содержащих явной семантической структуры.

Перспективным направлением является расширение модели в сторону содержательного анализа требований, прежде всего в разделе функциональных требований. Это позволит дополнить структурную проверку механизмами оценки полноты и согласованности формулировок.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Frattini J. Identifying Relevant Factors of Requirements Quality: An Industrial Case Study. In: *Requirements Engineering: Foundation for Software Quality: 30th International Working Conference, REFSQ 2024, 08–11 April 2024, Winterthur, Switzerland*. Cham: Springer; 2024. P. 20–36. https://doi.org/10.1007/978-3-031-57327-9_2
2. AbuSalim S.W.G., Ibrahim R., Mostafa S.A., Wahab J.A. Analyzing the Impact of Assessing Requirements Specifications on the Software Development Life Cycle. In: *Computational Science and Its Applications – ICCSA 2020: 20th International Conference: Proceedings: Part VI, 01–04 July 2024, Cagliari, Italy*. Cham: Springer; 2020. P. 632–648. https://doi.org/10.1007/978-3-030-58817-5_46
3. Wei X., Wang Zh., Yang Sh. An Automatic Generation and Verification Method of Software Requirements Specification. *Electronics*. 2023;12(12). <https://doi.org/10.3390/electronics12122734>
4. Umar M.A., Lano K., Abubakar A.K. Automated requirements engineering framework for agile model-driven development. *Frontiers in Computer Science*. 2025;7. <https://doi.org/10.3389/fcomp.2025.1537100>
5. Weitzl F., Jakšić M., Freitag B. Towards the Automated Verification of Semi-Structured Documents. *Data & Knowledge Engineering*. 2009;68(3):292–317. <https://doi.org/10.1016/j.datak.2008.10.003>
6. Moon S., Lee G., Chi S. Automated system for construction specification review using natural language processing. *Advanced Engineering Informatics*. 2022;51. <https://doi.org/10.1016/j.aei.2021.101495>
7. Nasyrov N., Komarov M., Tartynskikh P., Gorlushkina N. Automated formatting verification technique of paperwork based on the gradient boosting on decision trees. *Procedia Computer Science*. 2020;178:365–374. <https://doi.org/10.1016/j.procs.2020.11.038>
8. Ghosh Sh., Elenius D., Li W., et al. ARSENAL: Automatic Requirements Specification Extraction from Natural Language. In: *NASA Formal Methods: 8th International Symposium, 07–09 June 2016, Minneapolis, MN, USA*. Cham: Springer; 2016. P. 41–46. https://doi.org/10.1007/978-3-319-40648-0_4

9. Rodrigues A., da Silva A.R. *Validation of Rigorous Requirements Specifications and Document Automation with the ITLingo RSL Language*. arXiv. URL: <https://doi.org/10.48550/arXiv.2312.10822> [Accessed 19th December 2025].
10. Jiang Sh., Hu J., Magee Ch.L., Luo J. Deep Learning for Technical Document Classification. *IEEE Transactions on Engineering Management*. 2022;71:1163–1179. <https://doi.org/10.1109/TEM.2022.3152216>
11. Baratov D., Aripov N., Jumanov X., Muhiddinov O. Automated technology of control of technical documentation of automation and telemechanics. *AIP Conference Proceedings*. 2023;2612(1). <https://doi.org/10.1063/5.0135326>
12. Biswas Ch., Das S. ARIA-QA: AI-agent based requirements inspection and analysis through question answering. *Innovations in Systems and Software Engineering*. 2024;21:1009–1024. <https://doi.org/10.1007/s11334-024-00589-8>
13. Kolahdouz Rahimi Sh., Lano K., Tehrani S., et al. Comparative evaluation of NLP approaches for requirements formalisation. In: *Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering, 21–23 February 2024, Rome, Italy*. SCITEPRESS; 2024. P. 125–132. <https://doi.org/10.5220/0012318700003645>
14. Sonbol R., Rebdawi Gh., Ghneim N. The Use of NLP-Based Text Representation Techniques to Support Requirement Engineering Tasks: A Systematic Mapping Review. *IEEE Access*. 2022;10:62811–62830. <https://doi.org/10.1109/ACCESS.2022.3182372>
15. Ferrari A., Zhao L., Alhoshan W. NLP for Requirements Engineering: Tasks, Techniques, Tools, and Technologies. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 25–28 May 2021, Madrid, ES*. IEEE; 2021. P. 322–323. <https://doi.org/10.1109/ICSE-Companion52605.2021.00137>
16. Xu D., Chen W., Peng W., et al. Large language models for generative information extraction: a survey. *Frontiers of Computer Science*. 2024;18(6). <https://doi.org/10.1007/s11704-024-40555-y>

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Ечин Александр Васильевич, аспирант
Волгоградского государственного технического
университета, Волгоград, Российская Федерация.
e-mail: echin@vstu.ru
ORCID: [0009-0006-1419-934X](https://orcid.org/0009-0006-1419-934X)

Alexander V. Echin, Postgraduate, Volgograd
State Technical University, Volgograd, the
Russian Federation.

Алиева Наталья Денисовна, студент
Волгоградского государственного технического
университета, Волгоград, Российская Федерация.
e-mail: alieva_nata@icloud.com
ORCID: [0009-0009-9115-5000](https://orcid.org/0009-0009-9115-5000)

Natalia D. Alieva, Student, Volgograd State
Technical University, Volgograd, the Russian
Federation.

Садьков Артём Михайлович, студент
Волгоградского государственного технического
университета, Волгоград, Российская Федерация.
e-mail: asadykov1210@gmail.com
ORCID: [0009-0009-2739-0807](https://orcid.org/0009-0009-2739-0807)

Artem M. Sadykov, Student, Volgograd State
Technical University, Volgograd, the Russian
Federation.

Кравец Алла Григорьевна, доктор технических наук, профессор Волгоградского государственного технического университета, Волгоград, Российская Федерация.

e-mail: allagkravets@yandex.ru

ORCID: [0000-0003-1675-8652](https://orcid.org/0000-0003-1675-8652)

Alla G. Kravets, Doctor of Engineering Science, Professor, Volgograd State Technical University, Volgograd, the Russian Federation.

Сафонова Елена Владимировна, аспирант, преподаватель Волгоградского государственного технического университета, Волгоград, Российская Федерация.

e-mail: safonova_h@mail.ru

ORCID: [0009-0006-3798-6506](https://orcid.org/0009-0006-3798-6506)

Elena V. Safonova, Postgraduate, Lecturer, Volgograd State Technical University, Volgograd, the Russian Federation.

Статья поступила в редакцию 28.02.2026; одобрена после рецензирования 15.04.2026; принята к публикации 19.04.2026.

The article was submitted 28.02.2026; approved after reviewing 15.04.2026; accepted for publication 19.04.2026.