Себто Д. В.

Информационные системы и технологии

Качество данных, надёжность и практические решения

Сборник статей

Д. В. Себто

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ:

качество данных, надёжность и практические решения

Сборник статей

Электронное текстовое издание

Санкт-Петербург
Наукоемкие технологии
2026

© Себто Д. В., 2026
ISBN 978-5-00271-172-7

УДК 004
ББК 16
С28

С28 Себто Д. В. Информационные системы и технологии: качество данных, надёжность и практические решения [Электронный ресурс]: сборник статей / Д. В. Себто. – СПб.: Наукоемкие технологии, 2026. – 45 с. – URL: https://publishing.intelgr.com/archive/Informatsionnie-sistemi-i-tekhnologii.pdf.

ISBN 978-5-00271-172-7

В сборнике представлены десять работ, объединённых общей задачей — обеспечением достоверности, объяснимости и устойчивости информационных систем на всех этапах их жизненного цикла. Рассматриваются вопросы от переноса данных из электронных таблиц до предотвращения ошибок пользователей через доступные электронные формы. Особенность издания состоит в системном взгляде на качество данных и надёжность инфраструктуры: каждая глава дополняет другие, показывая, как отдельные технические решения влияют на целостность сведений, обработку ошибок и интерпретируемость результатов. В отличие от узкоспециализированных руководств, сборник сочетает аналитические обзоры с проектными подходами, предлагая читателю не готовые рецепты, а методы, которые можно адаптировать под конкретные организационные условия.

Издание адресовано разработчикам, аналитикам данных, системным архитекторам и руководителям ИТ-подразделений, которые стремятся повысить надёжность эксплуатируемых систем. Материалы будут полезны специалистам, занимающимся проектированием внутренних сервисов, миграцией баз данных, внедрением машинного обучения и поддержанием актуальности информационных активов. Сборник также может служить дополнительным источником для преподавателей и студентов технических направлений, интересующихся прикладными аспектами управления качеством и проверяемостью в разработке информационных систем.

УДК 004
ББК 16

ISBN 978-5-00271-172-7 © Себто Д. В., 2026

Научное издание

Себто Денис Владимирович

Информационные системы и технологии:
качество данных, надёжность и практические решения

Сборник статей

Электронное текстовое издание

Издается в авторской редакции
Публикуется из оригинал-макета, подготовленного автором
Главный редактор *В. М. Коровин*
Изображение на обложке www.magnific.com

Подписано к использованию 07.09.2026.
Объем издания – 0,9 Мб.

Издательство «Наукоемкие технологии»
ООО «Корпорация «Интел Групп»
https://publishing.intelgr.com
E-mail: publishing@intelgr.com
Тел.: +7 (812) 945-50-63

ISBN 978-5-00271-172-7



Содержание

Предисловие 5

Контроль качества данных при переносе сведений из электронных таблиц в информационную систему 6

Обеспечение проверяемости ответов искусственного интеллекта при поиске по документам организации 10

Сочетание правил и машинного обучения при выявлении аномалий в учетных операциях 14

Согласование данных мобильной информационной системы при работе без постоянного подключения к сети 18

Учет нерегулярного спроса при формировании рекомендаций по пополнению запасов 22

Минимизация персональных данных в журналах событий информационной системы 26

Контроль актуальности данных как часть наблюдаемости информационной системы 30

Критерии выбора модульной архитектуры информационной системы для небольшой команды разработки 34

Обеспечение воспроизводимости документов, автоматически формируемых информационной системой 38

Предотвращение ошибок пользователей посредством доступных электронных форм 42

Предисловие

В сборник вошли десять работ о проектировании и использовании информационных систем. Их объединяет вопрос о том, как сохранить достоверность сведений, объяснимость результатов и устойчивость рабочих процессов. Рассмотрение начинается с переноса данных из электронных таблиц и охватывает поиск по документам с помощью искусственного интеллекта, выявление аномалий, автономную работу приложений, прогнозирование спроса, защиту журналов событий, наблюдение за актуальностью данных, выбор архитектуры, формирование документов и доступность электронных форм.

Каждая глава представляет самостоятельную статью. Материалы можно читать последовательно либо выбирать по практической задаче. Главы объединяет внимание к проверяемым свойствам системы: происхождению сведений, сохранению целостности, обработке ошибок и возможности объяснить полученный результат. Рассмотренные подходы дополняют друг друга, поскольку надёжность отдельной операции зависит от качества данных, устройства программных компонентов и понятности действий пользователя.

Работы имеют аналитический и проектный характер. Условные примеры служат пояснением методов и не являются результатами производственных испытаний. Предлагаемые решения следует проверять на данных и процессах конкретной организации. Численные иллюстрации не следует переносить на другие условия без дополнительной оценки.

Контроль качества данных при переносе сведений из электронных таблиц в информационную систему

Электронная таблица часто становится промежуточным звеном между сотрудником и информационной системой. В ней удобно собирать заявки, редактировать адреса и сверять выполненные работы. Однако свободное изменение структуры создает неоднозначность при автоматическом переносе сведений. Одинаковое название может обозначать разные объекты, пустая ячейка может означать отсутствие данных или намеренное удаление значения, а повторный файл способен повторно увеличить итоговый показатель. Поэтому успешное открытие файла еще не свидетельствует о корректности загрузки.

Цель исследования состоит в разработке согласованной схемы контроля импорта для прикладной системы учета. Метод исследования включает сравнительный анализ публикаций о качестве данных и документации механизмов их хранения, а также разбор условных сценариев отказа. Работа имеет проектно-аналитический характер: производственный эксперимент не проводился. Качество рассматривается относительно задачи использования, что соответствует подходу Р. Ванга и Д. Стронг [1]. Основной вопрос заключается в том, какие проверки должны выполняться до записи, во время записи и после завершения загрузки.

До разработки преобразований требуется определить смысл одной строки. Если строка обозначает факт обслуживания объекта, его идентичность может задаваться номером события во внешней системе. Адрес при этом остается изменяемым атрибутом. Использование адреса вместо устойчивого идентификатора опасно: исправление опечатки будет выглядеть как появление нового объекта. Если исходная система не выдает ключ, составной идентификатор формируется только из признаков, сочетание которых действительно различает события. Это правило должно согласовываться с владельцем учета.

В исследованиях очистки данных различаются проблемы отдельных источников и проблемы объединения неоднородных сведений [2]. Для рассматриваемой задачи это означает необходимость отделять проверку формата от проверки смысла. Текстовое значение в колонке количества нарушает формат. Положительное количество, отнесенное к неизвестному товару, может соответствовать формату, но нарушать связи учета. Обе ошибки требуют понятного сообщения, однако исправляются по-разному: изменением значения либо уточнением справочника.

Предлагается принимать исходный файл в промежуточную область, сохраняя его контрольную сумму, время приема и номер строки. Преобразованное значение хранится отдельно от исходного. Такой прием позволяет объяснить происхождение результата и повторить обработку после

исправления программы. Автоматическая очистка допустима для заранее согласованных операций, например удаления внешних пробелов. Неоднозначное преобразование даты или объединение похожих наименований требует отдельного решения, поскольку внешнее сходство не доказывает тождественность объектов.

Проверка времени должна учитывать различие между календарной датой и моментом события. Дата обслуживания без указания часов не нуждается в искусственном переводе в другой часовой пояс. Для момента отправки сообщения, напротив, требуется однозначная временная привязка. RFC 3339 определяет представление времени со смещением относительно UTC [3]. В предлагаемой схеме сохраняется исходное представление, а для сравнения моментов используется единая шкала. Отчетный период задается интервалом с включенным началом и исключенным концом.

После предварительной проверки целостность поддерживается самой базой данных. Ограничение уникальности препятствует появлению повторного ключа, внешний ключ проверяет связь со справочником, а запрет пустого значения защищает обязательный реквизит [4]. Проверка только в пользовательском интерфейсе недостаточна: одновременно могут работать несколько загрузчиков. На уровне базы должно закрепляться именно бизнес-правило, а не случайная особенность файла. Например, название организации обычно не следует объявлять уникальным без дополнительного обоснования.

Повторная передача требует идемпотентности: обработка одного события несколько раз должна давать тот же учетный результат, что и однократная обработка. PostgreSQL предоставляет механизм INSERT ON CONFLICT для согласованной реакции на конфликт уникальности [5]. Однако выбор реакции остается задачей проектирования. Полное совпадение допускает пропуск, новая версия события требует обновления, а одинаковый номер при несовместимых реквизитах переводит запись на разбор. Безусловное игнорирование конфликтов скрывает исправления и потому не решает проблему качества.

Условный пример показывает различие между повторами и исправлениями. В поступившем файле имеется 100 строк: 90 новых корректных событий, шесть точных повторов и четыре строки без обязательного идентификатора. При независимой обработке записей в основную таблицу поступят 90 событий, а остальные строки сохранят отдельные статусы. Доля принятых новых строк составит 90%, доля повторов — 6%, доля отклоненных строк — 4%. Эти числа иллюстрируют расчет показателей и не являются результатами наблюдения за действующей организацией.

Если две строки из числа повторов содержат новую версию сведений, их нельзя продолжать учитывать как точные копии. В журнале загрузки должны различаться добавление, подтвержденное обновление и отсутствие изменений. Для связанного набора строк, представляющего один документ, частичный

прием может быть недопустим. Тогда единицей атомарности становится документ: все его строки записываются совместно либо не записываются вовсе. Граница такой операции определяется смыслом данных, а не размером файла или удобством программного цикла.

Оценка импорта должна включать баланс исходных и обработанных строк. Каждая строка получает ровно один итоговый статус, поэтому сумма добавленных, обновленных, пропущенных и отклоненных записей равна числу входных строк. При этом доля корректных строк не заменяет полноту поступления: полностью правильный файл может содержать лишь половину ожидаемых событий. Для выявления такой ситуации результат сопоставляется с контрольным количеством или итогом источника, если источник способен предоставить достоверную контрольную величину.

Проверка предлагаемой схемы возможна на заранее подготовленном наборе файлов. В него включаются повторная загрузка, изменение порядка строк, исправление события, разрыв операции и одновременный запуск двух загрузчиков. Сравниваются итоговые записи и объяснения отказов. Критерием приемки служит сохранение ожидаемого состояния учета при каждом сценарии. Такой протокол проверяет свойства системы, но не позволяет заранее утверждать определенный процент сокращения ошибок. Его ограничение связано с полнотой выбранных сценариев и качеством исходных правил.

Проведенный анализ позволяет разделить контроль импорта на проверку содержания, защиту целостности и подтверждение результата. Предложенная схема связывает каждую итоговую запись с источником и различает повтор, исправление и отклонение. Практический результат состоит в наборе проверяемых требований к загрузчику, который можно применять независимо от выбранного интерфейса и формата исходной таблицы.

Надежность переноса определяется согласованностью правил на всех этапах. Ограничения базы не устанавливают истинность введенного адреса, а предварительная проверка не защищает от всех конкурентных изменений. Поэтому дальнейшая работа должна включать испытание схемы на реальных файлах с подтвержденными ошибками и оценку времени их разбора. До такого испытания ожидаемые преимущества следует рассматривать как обоснованные проектные предположения. Снижение ошибок на этапе ввода дополнительно обеспечивается понятностью электронных форм; соответствующие требования рассмотрены в статье настоящего сборника [6]. Сохранение достоверности сведений во времени, включая контроль их актуальности и полноты, рассмотрено в статье о наблюдаемости данных настоящего сборника [7].

Список литературы

1. Wang, R. Y. Beyond Accuracy: What Data Quality Means to Data Consumers / R. Y. Wang, D. M. Strong // Journal of Management Information Systems. — 1996. — Vol. 12, no. 4. — P. 5–33. — DOI: 10.1080/07421222.1996.11518099.
2. Rahm, E. Data Cleaning: Problems and Current Approaches / E. Rahm, H. H. Do // IEEE Data Engineering Bulletin. — 2000. — Vol. 23, no. 4. — P. 3–13. — URL: https://dbs.uni-leipzig.de/research/publications/data-cleaning-problems-and-current-approaches (дата обращения: 02.09.2026). — Текст : электронный.
3. Klyne, G. Date and Time on the Internet: Timestamps : RFC 3339 / G. Klyne, C. Newman. — 2002. — DOI: 10.17487/RFC3339. — URL: https://www.rfc-editor.org/rfc/rfc3339 (дата обращения: 02.09.2026). — Текст : электронный.
4. PostgreSQL Global Development Group. Constraints // PostgreSQL 18 Documentation. — URL: https://www.postgresql.org/docs/18/ddl-constraints.html (дата обращения: 02.09.2026). — Текст : электронный.
5. PostgreSQL Global Development Group. INSERT // PostgreSQL 18 Documentation. — URL: https://www.postgresql.org/docs/18/sql-insert.html (дата обращения: 02.09.2026). — Текст : электронный.
6. Себто, Д. В. Предотвращение ошибок пользователей посредством доступных электронных форм / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 42–45.
7. Себто, Д. В. Контроль актуальности данных как часть наблюдаемости информационной системы / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 30–33.

Обеспечение проверяемости ответов искусственного интеллекта при поиске по документам организации

Сотрудник организации может тратить значительное время на поиск ответа в положениях, инструкциях и переписке. Языковая модель позволяет сформулировать вопрос обычными словами и получить связное объяснение. Однако правдоподобная формулировка не гарантирует, что ответ соответствует действующему документу. Для внутренней информационной системы существенна возможность установить происхождение каждого значимого утверждения. Особенно важна эта возможность при изменении сроков, последовательности действий или условий предоставления услуги.

Цель исследования заключается в определении условий проверяемого ответа при поиске по корпоративным документам. Использованы анализ исследований в области поиска и генерации текста и моделирование типовых ошибок. Работа не содержит результатов испытания конкретного программного продукта. Подход RAG объединяет получение внешних материалов и генерацию ответа с их использованием [1]. Предмет исследования ограничен справочными ответами: автоматическое принятие управленческих решений не рассматривается. Проверяемость понимается как возможность сопоставить утверждение с доступным источником и его действующей версией.

Обычный поиск возвращает документы, а языковая модель формирует из найденных фрагментов объяснение. Между этими действиями существует принципиальная граница. Поисковый результат показывает вероятную полезность материала, но не доказывает истинность будущего ответа. Поэтому оценка всей системы одним показателем удобства недостаточна. В предлагаемом подходе отдельно проверяется нахождение нужного фрагмента и отдельно — соответствие сформулированного вывода содержанию этого фрагмента. Ошибка на первом этапе не должна маскироваться уверенным стилем второго.

При подготовке базы знаний каждому документу присваивается устойчивый идентификатор. Вместе с текстом сохраняются версия, дата начала действия и сведения о допустимой аудитории. Для отмененного документа фиксируется его статус. Разбиение текста на части должно сохранять связи условия с исключением, заголовка с абзацем и таблицы с пояснением. Если исключение попадет в недоступную для поиска часть, формально правильная цитата может поддерживать ошибочный общий вывод. Поэтому размер фрагмента определяется структурой документа, а не только техническим пределом модели.

Поиск по смыслу представляет вопрос и текст числовыми векторами, позволяющими сопоставлять близкие формулировки. В работе о Dense Passage Retrieval исследуется использование таких представлений для извлечения

фрагментов [2]. Для организационных документов предлагается сочетать смысловой поиск с точным поиском реквизитов. Номер приказа или обозначение оборудования может иметь большее значение, чем общая близость темы. После первичного отбора требуется проверять дату действия и соответствие документа области вопроса.

Права доступа применяются до передачи материалов генератору. Если система сначала извлечет закрытый документ, а затем скроет ссылку, содержание уже может попасть в ответ. Такой результат несовместим с задачей контролируемого внутреннего поиска. Фильтрация должна распространяться и на промежуточные хранилища, в которых сохраняются подготовленные ответы. При изменении прав пользователя ранее созданный ответ нельзя автоматически считать доступным только потому, что он уже присутствует в памяти приложения.

Передача большого количества текста не устраняет необходимость отбора. Исследование Н. Лю и соавторов показывает зависимость качества рассмотренных моделей от положения полезной информации в длинном контексте [3]. Этот результат не следует превращать в универсальную оценку всех современных моделей. Однако он обосновывает проверку конкретной конфигурации при изменении длины и порядка фрагментов. В предлагаемой схеме предпочтение отдается компактному набору взаимосвязанных источников, достаточному для ответа на поставленный вопрос.

Генератор должен связывать проверяемые утверждения с конкретными фрагментами. Наличие ссылки само по себе недостаточно: источник может не подтверждать указанное число или относиться к другой ситуации. Исследование генерации текста с цитированием выделяет качество ссылок в отдельное направление оценки [4]. Поэтому предлагается проверять как подтвержденность утверждения, так и полноту ссылок. Если ответ содержит три существенных положения, подтверждение одного из них не означает обоснованности всего ответа.

Условный пример связан с вопросом о сроке согласования заявки. В архивной инструкции указан срок пять дней, а в действующей редакции — три рабочих дня после получения полного комплекта сведений. Ответ «три дня» теряет условие и характер дней. Корректный результат должен сохранить эти ограничения и указать действующую редакцию. При отсутствии сведений о комплектности система сообщает, что начало отсчета определить нельзя. Такой отказ относится к недостающим данным, а не к технической неисправности поиска.

Для оценки предлагается заранее составить набор вопросов с экспертно подтвержденными фрагментами. Часть вопросов должна иметь однозначный ответ, часть — зависеть от условий, а часть — не иметь ответа в доступных материалах. Отдельные задания включают противоречащие друг другу

редакции. RAGAs предлагает средства раздельной оценки извлечения и генерации [5]. Автоматические оценки в рассматриваемой схеме используются для отбора подозрительных случаев, а выборочная ручная проверка нужна для контроля ошибок самого оценщика.

Основными показателями становятся доля найденных подтверждающих фрагментов, доля подтвержденных утверждений и обоснованность отказов. Последний показатель препятствует формальному улучшению надежности за счет отказа от большинства ответов. Например, система, которая всегда сообщает о недостаточности данных, почти не создает ложных утверждений, но не решает пользовательскую задачу. Поэтому результаты сопоставляются с простым поиском по документам и оцениваются одновременно по полезности и проверяемости при одинаковом наборе вопросов.

Предлагаемый порядок также предусматривает журналирование версии источников и настроек обработки. Если документ изменился, ранее подготовленный ответ должен быть пересмотрен либо помечен датой актуальности. Нельзя считать все старые ответы неверными, но нельзя и переносить их в новый контекст без проверки. Ограничением подхода остается зависимость от качества самой базы знаний. Если действующий документ содержит ошибку, точное воспроизведение его содержания обеспечивает прослеживаемость ответа, однако не устанавливает фактическую правильность исходного положения.

Результатом исследования является порядок формирования ответа, в котором поиск, проверка действия документа и подтверждение утверждений оцениваются раздельно. Такой подход делает причины ошибок наблюдаемыми и позволяет определить, требуется ли улучшить извлечение, изменить подготовку текста или уточнить содержание источника. Отказ от ответа рассматривается как допустимый результат при недостаточности оснований, но не как самостоятельная цель системы.

Практическая применимость предложенной схемы требует проверки на документах конкретной организации. До проведения испытаний нельзя утверждать количественное снижение недостоверных ответов. Дальнейшее исследование целесообразно связать с сопоставлением времени поиска и числа подтвержденных ответов при ручной работе, обычном поиске и использовании RAG. Во всех вариантах должны применяться одинаковые документы, права доступа и критерии правильности. Близкая задача фиксации происхождения результата применительно к автоматически формируемым документам рассмотрена в статье настоящего сборника [6].

Список литературы

1. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis, E. Perez, A. Piktus [et al.] // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 9459–9474. — URL: https://arxiv.org/abs/2005.11401 (дата обращения: 02.09.2026). — Текст : электронный.
2. Dense Passage Retrieval for Open-Domain Question Answering / V. Karpukhin, B. Oguz, S. Min [et al.] // Proceedings of EMNLP 2020. — 2020. — P. 6769–6781. — DOI: 10.18653/v1/2020.emnlp-main.550.
3. Lost in the Middle: How Language Models Use Long Contexts / N. F. Liu, K. Lin, J. Hewitt [et al.] // Transactions of the Association for Computational Linguistics. — 2024. — Vol. 12. — P. 157–173. — DOI: 10.1162/tacl_a_00638.
4. Enabling Large Language Models to Generate Text with Citations / T. Gao, H. Yen, J. Yu, D. Chen // Proceedings of EMNLP 2023. — 2023. — P. 6465–6488. — DOI: 10.18653/v1/2023.emnlp-main.398.
5. RAGAs: Automated Evaluation of Retrieval Augmented Generation / S. Es, J. James, L. Espinosa Anke, S. Schockaert // Proceedings of EACL 2024: System Demonstrations. — 2024. — P. 150–158. — DOI: 10.18653/v1/2024.eacl-demo.16.
6. Себто, Д. В. Обеспечение воспроизводимости документов, автоматически формируемых информационной системой / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 38–41.

Сочетание правил и машинного обучения при выявлении аномалий в учетных операциях

Информационные системы накапливают сведения о заявках, перемещениях товаров и выполненных работах. Некоторые записи заметно отличаются от обычного поведения: заявка появляется в непривычное время, количество многократно превышает типичное значение или одна операция повторяется несколько раз. Такие отличия полезны для контроля, но сами по себе не доказывают ошибку или нарушение. Необычный объем может быть связан с разовой поставкой, а поздняя регистрация — с восстановлением связи после перерыва.

Цель исследования состоит в разработке подхода к отбору операций для проверки с учетом ограниченных ресурсов специалиста. Метод включает анализ научных публикаций, сопоставление способов выявления отклонений и условный расчет показателей. Производственные данные не использовались. В обзоре В. Чандолы и соавторов показано разнообразие постановок задачи обнаружения аномалий [1]. В настоящей работе аномалия определяется как сигнал для дополнительного рассмотрения, а подтвержденная проблема — как результат проверки по установленным предметным критериям.

Явные правила подходят для условий, которые можно сформулировать однозначно. Например, отрицательная длительность между началом и завершением операции требует проверки исходных дат. Преимущество правила заключается в понятном основании срабатывания. Ограничение возникает при попытке описать все допустимые сочетания признаков. Изменение графика работы или появление нового типа заказов может сделать ранее полезный порог слишком строгим. Поэтому каждое правило должно иметь владельца, объяснение назначения и дату пересмотра.

Модель обнаружения аномалий может находить сочетания признаков, заранее не записанные в правилах. Метод Isolation Forest использует идею изоляции наблюдений случайными разбиениями: некоторые необычные объекты отделяются от остальных сравнительно быстро [2]. Полученная оценка отражает особенности выбранного представления данных. Она не является доказанной вероятностью нарушения. Если в модель переданы неинформативные признаки или смешаны разные процессы, высокий балл может описывать устройство набора данных, а не практическую проблему.

Предлагается формировать сравнимые группы операций до расчета отклонений. Заявки на плановое обслуживание не следует без обоснования сопоставлять со срочными аварийными работами. Для количества полезно учитывать тип товара, а для времени — режим работы подразделения. Такая подготовка снижает риск считать нормальное различие между группами аномалией внутри процесса. При этом слишком мелкое разделение также нежелательно: группа

из нескольких наблюдений не дает устойчивого представления об обычном поведении.

Совместная схема использует правила для определенных ограничений, а модель — для ранжирования остальных случаев. Результаты объединяются в очереди проверки с сохранением отдельных оснований. Суммирование всех баллов без калибровки не предлагается, поскольку шкалы могут иметь разный смысл. Повторное срабатывание нескольких правил на одну операцию не должно создавать несколько одинаковых задач специалисту. В карточке случая сохраняются исходные значения, сработавшие условия и сопоставимые исторические наблюдения.

Значение редкости проблем показывает условный расчет. Пусть из 10 000 операций действительно проблемными являются 100. Система обнаруживает 90 из них, но ошибочно отмечает 1% остальных 9900 операций, то есть 99 записей. В очередь поступает 189 случаев, из которых подтверждаются 90. Точность предупреждений составляет 90/189, или приблизительно 47,6%, а полнота обнаружения — 90%. Пример рассчитан для заданных предпосылок и не описывает качество какого-либо действующего решения.

Общая доля правильных классификаций в этом примере равна 98,91%, поскольку верно распознаны 90 проблемных и 9801 обычная операция. Тем не менее более половины предупреждений оказываются ложными. Следовательно, высокий общий показатель может скрывать значительную нагрузку на проверяющего. Исследование Т. Сайто и М. Ремсмайер обосновывает внимание к соотношению точности и полноты при несбалансированных классах [3]. Для прикладной системы дополнительно требуется измерять число предупреждений за рабочий период.

Выбор порога должен учитывать доступную емкость проверки. Если специалист способен рассмотреть 30 случаев за день, очередь из 300 событий нельзя считать работоспособным результатом без механизма приоритизации. В предлагаемой схеме отдельно оценивается доля подтвержденных проблем среди первых случаев очереди. При этом ограничение очереди способно снизить полноту, и данный эффект должен отражаться в отчете. Утверждение об улучшении качества допустимо только при сопоставимых ресурсах и одинаковом определении подтвержденной проблемы.

Объяснение решения должно быть понятным проверяющему и соответствовать фактическому механизму модели. Локальные методы интерпретации, рассматриваемые М. Рибейру и соавторами, помогают исследовать вклад признаков в отдельный прогноз [4]. Однако такое объяснение не устанавливает причинно-следственную связь. Формулировка «операция отмечена из-за необычного количества» сообщает об основании отбора, но не подтверждает умышленное действие сотрудника. Интерфейс должен сохранять это различие и позволять перейти к исходным записям.

Обратная связь включает результат проверки и причину принятого решения. Полезно различать ошибку данных, допустимое исключение и подтвержденное отклонение процесса. Если для обучения используются только проверенные предупреждения, возникает смещение: сведения о неотмеченных операциях остаются неизвестными. Поэтому план оценки должен предусматривать допустимую по трудозатратам случайную проверку части обычных записей. Этот прием не гарантирует нахождение всех пропусков, но позволяет получить сведения о качестве вне уже сформированной очереди.

Испытание системы проводится с временным разделением данных: настройка выполняется на более раннем периоде, а оценка — на последующем. Иначе сведения о будущих событиях могут незаметно попасть в признаки. Мониторинг после внедрения необходим, поскольку процесс и состав данных меняются. Рамочная модель NIST связывает управление рисками ИИ с непрерывной оценкой в контексте применения [5]. В предлагаемой схеме пересмотр правил и модели запускается при изменении структуры операций или устойчивом ухудшении полезности предупреждений.

Исследование обосновывает совместное использование правил, модели и экспертной проверки как разных компонентов контроля. Результат модели следует трактовать как основание для приоритизации, сохраняя различие между необычностью и подтвержденной проблемой. Условный расчет показывает, почему оценка по общей доле верных ответов недостаточна при редких событиях и почему нагрузка на специалиста должна входить в критерии качества.

Предложенный подход задает проверяемую организацию очереди и обратной связи, но не подтверждает превосходство конкретного алгоритма. Для эмпирической проверки необходимы размеченные операции, временное разделение и согласованные правила экспертной оценки. Дальнейшее исследование должно сопоставить полезность предупреждений и число пропущенных проблем при одинаковой доступной емкости проверки. Смежные вопросы оценки прогнозов на нерегулярных данных рассмотрены в статье о пополнении запасов настоящего сборника [6].

Список литературы

1. Chandola, V. Anomaly Detection: A Survey / V. Chandola, A. Banerjee, V. Kumar // ACM Computing Surveys. — 2009. — Vol. 41, no. 3. — Article 15. — DOI: 10.1145/1541880.1541882.
2. Liu, F. T. Isolation Forest / F. T. Liu, K. M. Ting, Z.-H. Zhou // 2008 Eighth IEEE International Conference on Data Mining. — 2008. — P. 413–422. — DOI: 10.1109/ICDM.2008.17.

3. Saito, T. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets / T. Saito, M. Rehmsmeier // PLOS ONE. — 2015. — Vol. 10, no. 3. — e0118432. — DOI: 10.1371/journal.pone.0118432.

4. Ribeiro, M. T. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier / M. T. Ribeiro, S. Singh, C. Guestrin // Proceedings of KDD 2016. — 2016. — P. 1135–1144. — DOI: 10.1145/2939672.2939778.

5. Tabassi, E. Artificial Intelligence Risk Management Framework (AI RMF 1.0) : NIST AI 100-1. — Gaithersburg : NIST, 2023. — DOI: 10.6028/NIST.AI.100-1. — URL: https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10 (дата обращения: 02.09.2026). — Текст : электронный.

6. Себто, Д. В. Учет нерегулярного спроса при формировании рекомендаций по пополнению запасов / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 22–25.

Согласование данных мобильной информационной системы при работе без постоянного подключения к сети

Мобильные информационные системы применяются при осмотре объектов, выполнении выездных работ и сборе показаний. Постоянное подключение к серверу в таких условиях может отсутствовать. Если приложение требует связи для каждого действия, сотрудник вынужден переносить сведения в бумажные записи или повторно вводить их после возвращения. Локальное сохранение уменьшает зависимость от доступности сети, однако создает новую задачу: несколько устройств могут изменять один объект, не зная о действиях друг друга.

Цель исследования состоит в определении правил безопасного согласования локальных изменений с общей базой данных. Метод включает анализ публикаций о локально ориентированных приложениях и согласованности распределенных систем, а также разбор условных конфликтов. В работе не измеряются характеристики конкретного мобильного приложения. Идея local-first предполагает значимую роль локальной копии данных и возможность продолжать работу без постоянной связи [1]. Исследование сосредоточено на учетных операциях, для которых потеря изменения или ложное подтверждение завершения недопустимы.

Предлагается разделять состояние записи на локально сохраненное и подтвержденное общей системой. Нажатие кнопки сохранения сначала фиксирует изменение на устройстве. Отправка становится отдельной задачей, которую приложение повторяет при восстановлении связи. Пользователь должен видеть различие между этими состояниями. Сообщение «сохранено» без уточнения места хранения создает неверное ожидание: сотрудник может считать, что диспетчер уже получил результат, хотя данные еще не покинули устройство.

Изменение записи и добавление задания на отправку должны выполняться в одной локальной транзакции. Иначе возможна ситуация, когда запись сохранена, но отправлять ее нечему, либо задание создано без соответствующих данных. Документация SQLite описывает механизмы атомарной фиксации изменений и предположения о поведении накопителя [2]. Для проекта это означает необходимость проверять сохранность не только при нормальном закрытии приложения, но и при принудительном завершении. Надежность локальной записи зависит также от исправности устройства и его хранилища.

Каждое изменение получает отдельный идентификатор операции. При повторной отправке используется тот же идентификатор, а сервер сохраняет результат его обработки. Если связь оборвалась после принятия изменения, но до получения ответа, повторный запрос не должен создавать второй факт. Идентификатор объекта и идентификатор изменения выполняют разные функции:

один обозначает карточку, другой — конкретное действие над ней. Их смешение затрудняет обработку нескольких последовательных изменений одного объекта.

Для редактируемой карточки дополнительно сохраняется версия, на основании которой пользователь начал работу. Сервер сравнивает ее с текущей версией перед применением изменения. Механизмы условных HTTP-запросов, включая If-Match, предназначены в том числе для предотвращения потери обновлений [3]. Несовпадение версии должно возвращать понятный конфликт. Автоматическая перезапись по времени устройства ненадежна, поскольку часы могут расходиться, а более позднее сохранение не обязательно содержит более правильные сведения.

Условный пример связан с остатком, равным 12 единицам. Два устройства независимо фиксируют выдачу одной единицы и каждое получает локальное значение 11. Если сервер принимает последнее переданное значение, итог остается равным 11 вместо 10. Передача двух разных событий выдачи позволяет сохранить оба действия. Однако и событийная модель не решает автоматически проблему превышения доступного остатка: при отсутствии связи несколько устройств могут совместно выдать больше, чем имеется в наличии.

Для ограниченного ресурса требуется отдельная предметная политика. Возможен предварительный резерв количества для каждого устройства или запрет окончательного подтверждения выдачи до связи с сервером. Другой вариант допускает регистрацию факта с последующим разбором превышения. Выбор зависит от процесса и не может быть сделан универсальным алгоритмом синхронизации. Предлагается явно различать сообщение о наблюдавшемся действии и разрешение на действие, которое требует знания общего состояния системы.

Некоторые изменения допускают автоматическое объединение. Добавление разных фотографий к осмотру обычно не требует выбора одной из них. Для определенных типов данных структуры CRDT обеспечивают сходимость реплик при соблюдении заданных условий [4]. Однако сходимость означает получение одинакового результата на копиях, а не его предметную правильность. Объединение двух несовместимых заключений о состоянии оборудования требует правила процесса или решения ответственного сотрудника, даже если техническое хранение обеих версий не вызывает трудностей.

Удаление также должно передаваться как изменение. Если запись просто исчезнет с одного устройства, другая копия может отправить ее повторно и восстановить объект. Поэтому предлагается хранить отметку удаления до подтверждения необходимыми участниками синхронизации либо до выполнения согласованной политики хранения. Для вложений требуется самостоятельный контроль завершенности передачи. Наличие карточки с указанием фотографии

не доказывает, что файл уже доступен на сервере и может быть открыт другим пользователем.

Ограничения согласованности нельзя устранить только увеличением числа повторных запросов. Результат С. Гилберта и Н. Линч показывает несовместимость определенных гарантий сильной согласованности и доступности при разделении сети [5]. Для рассматриваемой системы из этого следует необходимость выбирать поведение каждой операции. Черновик осмотра может сохраняться локально, тогда как окончательное распределение общего ресурса может требовать связи. Такой выбор позволяет сохранить полезную часть работы без ложного обещания немедленной общей согласованности.

Испытания должны включать потерю сети в разные моменты операции, повторную доставку, изменение порядка сообщений, несовпадение версий и восстановление после перезапуска. Проверяется, что каждое подтвержденное действие учтено один раз, неподтвержденные изменения остаются видимыми, а конфликт не уничтожает исходные варианты. Отдельно измеряются возраст неотправленных данных и время до подтверждения после восстановления связи. Эти показатели описывают качество синхронизации точнее, чем единственный признак доступности сервера.

Предложенная схема связывает локальную транзакцию, устойчивый идентификатор изменения и проверку версии на сервере. Ее результатом является явное разделение сохранения, передачи и принятия операции. Анализ конфликтов показывает, что автоматическое объединение применимо только при наличии подходящих свойств данных и согласованных предметных правил. Техническая сходимость копий не заменяет проверку допустимости учетного результата.

Практическая проверка подхода требует испытаний на устройствах и в условиях связи, соответствующих реальному применению. До таких испытаний нельзя гарантировать время синхронизации или отсутствие потерь при повреждении оборудования. Перспективным направлением является сопоставление политик резервирования и последующего согласования для разных типов выездных операций с учетом удобства работы и частоты конфликтов. Связь распределения данных с выбором архитектуры приложения рассмотрена в статье настоящего сборника [6].

Список литературы

1. Local-first Software: You Own Your Data, in spite of the Cloud / M. Kleppmann, A. Wiggins, P. van Hardenberg, M. McGranaghan // Proceedings of Onward! 2019. — 2019. — P. 154–178. — DOI: 10.1145/3359591.3359737. — URL: https://www.inkandswitch.com/essay/local-first/ (дата обращения: 02.09.2026). — Текст : электронный.

2. SQLite. Atomic Commit in SQLite. — URL: https://sqlite.org/atomiccommit.html (дата обращения: 02.09.2026). — Текст : электронный.

3. Fielding, R. HTTP Semantics : RFC 9110 / R. Fielding, M. Nottingham, J. Reschke. — 2022. — DOI: 10.17487/RFC9110. — URL: https://www.rfc-editor.org/rfc/rfc9110.html (дата обращения: 02.09.2026). — Текст : электронный.

4. A Comprehensive Study of Convergent and Commutative Replicated Data Types / M. Shapiro, N. Preguiça, C. Baquero, M. Zawirski. — INRIA, 2011. — Research Report RR-7506. — URL: https://inria.hal.science/inria-00555588 (дата обращения: 02.09.2026). — Текст : электронный.

5. Gilbert, S. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services / S. Gilbert, N. Lynch // ACM SIGACT News. — 2002. — Vol. 33, no. 2. — P. 51–59. — DOI: 10.1145/564585.564601.

6. Себто, Д. В. Критерии выбора модульной архитектуры информационной системы для небольшой команды разработки / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 34–37.

Учет нерегулярного спроса при формировании рекомендаций по пополнению запасов

Система управления запасами должна помогать определять момент и объем пополнения. Для регулярно продаваемых товаров среднее потребление может служить понятным ориентиром. Однако у запасных частей и редко используемых материалов периоды без расхода чередуются с отдельными крупными заявками. В таких условиях рекомендация, основанная только на последней продаже или текущем остатке, может быть недостаточно обоснованной. Нулевой расход за неделю не доказывает отсутствия потребности в следующем периоде.

Цель исследования заключается в разработке объяснимой схемы рекомендаций для информационной системы учета нерегулярного спроса. Метод включает анализ исследований прогнозирования, сопоставление учетных величин и расчет условного примера. Реальные сведения о продажах не использовались, экономический эффект внедрения не оценивался. В литературе по прогнозированию подчеркивается необходимость проверки модели на данных, которые не участвовали в ее настройке [1]. В настоящей работе прогноз рассматривается как один из входов решения о пополнении, а не как готовое распоряжение о закупке.

Подготовка данных начинается с различения спроса и наблюдаемой продажи. Если товар отсутствовал, нулевая продажа может означать невозможность выполнить запрос. Запись такого дня как обычного нулевого спроса занижает оценку потребности. Поэтому предлагается сохранять сведения о наличии товара и невыполненных запросах, когда они доступны. Возвраты и исправления учета также выделяются отдельно: отрицательное движение не является отрицательной потребностью покупателя. Перед расчетом определяется единая единица измерения, чтобы упаковки не смешивались со штуками.

Для нерегулярного спроса характерны два разных источника неопределенности: время следующего события и его объем. Метод Дж. Кростона раздельно оценивает ненулевой размер спроса и интервал между событиями [2]. Это делает постановку более подходящей для разреженных рядов, чем простое продолжение последнего значения. Однако полученный средний темп не задает точную дату следующего заказа. Пользовательский интерфейс должен отражать эту разницу и не представлять среднее значение как обещанный расход конкретного дня.

Классификация характера спроса помогает выбирать кандидатов для сравнения. А. Синтетос и соавторы рассматривают частоту возникновения спроса и изменчивость его размера как основания такой классификации [3]. В предлагаемой системе эти признаки служат диагностикой, а не безусловным правилом

выбора одного алгоритма. История товара может быть короткой, а наблюдаемый режим — временным. Поэтому выбранный метод должен сопоставляться с простым базовым прогнозом на нескольких последующих отрезках времени.

Рекомендация по пополнению зависит от позиции запаса, а не только от физического остатка. В рассматриваемой схеме она равна доступному количеству на складе плюс подтвержденным поступлениям минус обязательствам, еще не вычтенным из доступного количества. Важно избежать двойного учета резервов: если резерв уже исключен из показателя доступности, повторно вычитать его нельзя. Каждая составляющая должна иметь понятное определение, дату актуальности и источник. Без этого одинаковая формула может давать разные результаты в разных организациях.

Условный пример использует физический остаток 18 единиц, подтвержденное поступление 10 единиц и незакрытые обязательства на восемь единиц. Предполагается, что обязательства еще не вычтены из остатка, поэтому позиция запаса составляет $18 + 10 - 8 = 20$ единиц. Если выбранный целевой уровень равен 26 единицам, расчетное пополнение составляет шесть единиц до учета кратности упаковки. Целевой уровень здесь задан для иллюстрации и не обоснован реальными наблюдениями; пример показывает только взаимосвязь учетных величин.

При непрерывном контроле существенен спрос за время поставки. При периодическом пересмотре требуется учитывать также промежуток до следующего решения. Если заказ проверяется раз в неделю, расчет только на срок доставки может оставить незащищенный интервал. Предлагается хранить длительность поставки как наблюдаемую характеристику с возможными отклонениями, а не как неизменную константу. Подтвержденное поступление, ожидаемое после защищаемого периода, нельзя автоматически использовать для покрытия более ранней потребности.

Оценка прогноза должна учитывать нулевые значения. Процентная ошибка, в знаменателе которой стоит фактический спрос, не определена при нуле. Работа Р. Хайндмана и А. Келер показывает ограничения распространенных мер точности и рассматривает масштабированную абсолютную ошибку [4]. Для прикладной проверки предлагается дополнительно анализировать ошибку суммарного спроса на срок поставки. Масштабированные показатели также требуют проверки знаменателя, особенно если обучающая последовательность постоянна или полностью состоит из нулей.

Даже приемлемая ошибка прогноза не гарантирует полезной политики запаса. Занижение и завышение могут иметь неодинаковые последствия: первое увеличивает риск отсутствия товара, второе — объем хранения. Поэтому сравнение методов должно включать имитацию одной и той же политики пополнения на исторических периодах. Фиксируются доля удовлетворенной потребности и средний запас при явно заданных сроках поставки. До получения

данных о затратах эти результаты нельзя автоматически переводить в денежную экономию.

Длительное отсутствие спроса требует отдельного внимания. Классический прогноз может сохранять положительное значение после прекращения использования товара. Р. Теунтер и соавторы рассматривают связь прогнозирования нерегулярного спроса с устареванием запасов [5]. В предлагаемой схеме длительный период без потребности запускает проверку статуса позиции. При этом отсутствие продаж из-за недоступности товара нельзя смешивать с прекращением спроса. Для решения полезны сведения о замене модели оборудования и завершении обслуживаемого проекта.

Проверка рекомендаций выполняется последовательно по времени. Для каждой контрольной даты используются только сведения, которые были доступны к этому моменту. Поступления, фактически задержанные позднее, не должны заранее заменяться известной итоговой датой при формировании решения. Отдельно анализируются товары с короткой историей и длительными перерывами. Рекомендация сопровождается объяснением использованного остатка, ожидаемого поступления и горизонта потребности. При недостаточности данных система указывает причину ограничения, сохраняя возможность решения специалистом.

Предложенная схема связывает прогноз с позицией запаса, сроком поставки и достоверностью учетных величин. Анализ показывает, что нерегулярность спроса требует различать размер события и время его появления, а полезность рекомендации следует проверять отдельно от ошибки прогноза. Условный расчет демонстрирует, почему сравнение текущего остатка с произвольным порогом не заменяет согласованного расчета пополнения.

Результатом работы являются требования к данным и протоколу оценки рекомендаций. Преимущество конкретного метода может быть установлено только на сопоставимых временных периодах при одинаковой политике обслуживания потребности. Дальнейшее исследование целесообразно направить на моделирование задержек поставки и неполноты регистрации спроса, поскольку эти факторы способны изменить решение даже при неизменном алгоритме прогнозирования. Задача выявления необычных операций и ограничения общих показателей точности рассмотрены в статье настоящего сборника [6].

Список литературы

1. Hyndman, R. J. Forecasting: Principles and Practice / R. J. Hyndman, G. Athanasopoulos. — 3rd ed. — Melbourne : OTexts, 2021. — URL: https://otexts.com/fpp3/ (дата обращения: 02.09.2026). — Текст : электронный.

2. Croston, J. D. Forecasting and Stock Control for Intermittent Demands // Operational Research Quarterly. — 1972. — Vol. 23, no. 3. — P. 289–303. — DOI: 10.1057/jors.1972.50.

3. Syntetos, A. A. On the Categorization of Demand Patterns / A. A. Syntetos, J. E. Boylan, J. D. Croston // Journal of the Operational Research Society. — 2005. — Vol. 56, no. 5. — P. 495–503. — DOI: 10.1057/palgrave.jors.2601841.

4. Hyndman, R. J. Another Look at Measures of Forecast Accuracy / R. J. Hyndman, A. B. Koehler // International Journal of Forecasting. — 2006. — Vol. 22, no. 4. — P. 679–688. — DOI: 10.1016/j.ijforecast.2006.03.001.

5. Teunter, R. H. Intermittent Demand: Linking Forecasting to Inventory Obsolescence / R. H. Teunter, A. A. Syntetos, M. Z. Babai // European Journal of Operational Research. — 2011. — Vol. 214, no. 3. — P. 606–615. — DOI: 10.1016/j.ejor.2011.05.018.

6. Себто, Д. В. Сочетание правил и машинного обучения при выявлении аномалий в учетных операциях / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 14–17.

Минимизация персональных данных в журналах событий информационной системы

Журналы событий помогают выяснить, почему операция завершилась ошибкой и какие действия предшествовали сбою. Для удобства диагностики разработчик может сохранять полное содержимое запроса, включая имя, адрес, телефон и данные доступа. В результате технический журнал превращается в дополнительное хранилище чувствительной информации. Его состав часто контролируется слабее, чем основная база, хотя доступ к нему могут иметь специалисты сопровождения и внешние системы наблюдения. Поэтому полезность подробной записи необходимо сопоставлять с назначением каждого сохраняемого поля.

Цель исследования состоит в разработке схемы журналирования, сохраняющей диагностическую ценность при ограничении избыточных сведений. Использованы анализ рекомендаций NIST и OWASP и моделирование жизненного цикла события. Работа не является правовым заключением и не устанавливает универсальные сроки хранения. Рамочная модель приватности NIST рассматривает риски, связанные с обработкой данных, в контексте деятельности организации [1]. В настоящем исследовании внимание сосредоточено на инженерных решениях, позволяющих объяснять сбой без автоматического копирования всей пользовательской информации.

Проектирование журнала начинается с вопроса, какое действие должен выполнить специалист после получения записи. Для ошибки отправки документа обычно требуется установить операцию, этап обработки и код отказа. Полный текст документа нужен далеко не всегда. Предлагается задавать схему события заранее и разрешать только поля, связанные с конкретной диагностической задачей. Такой подход делает состав записи проверяемым и уменьшает зависимость от случайного содержания объектов, которые программа передает стандартному механизму вывода ошибок.

Рекомендации OWASP по журналированию выделяют сведения, которые не следует напрямую включать в записи, в том числе пароли и токены доступа [2]. Для рассматриваемой системы предлагается хранить идентификатор операции, тип события, время, результат и технический код причины. Если требуется связать событие с учетной записью, используется внутренний идентификатор. Передача полного имени или телефона в каждую строку увеличивает количество копий без обязательного увеличения диагностической полезности.

Идентификатор корреляции объединяет записи одного процесса, проходящего через несколько компонентов. Он должен быть самостоятельным техническим значением, а не номером телефона или адресом электронной почты. По нему специалист восстанавливает последовательность действий, не получая

автоматически все сведения о человеке. При необходимости переход к карточке выполняется через основной интерфейс с отдельной проверкой полномочий. Таким образом, техническое расследование и доступ к персональным реквизитам становятся связанными, но различными операциями.

Замена имени идентификатором не делает набор данных автоматически анонимным. Если существует таблица соответствия или запись легко сопоставляется с другими источниками, идентификация остается возможной. NIST SP 800-188 рассматривает методы удаления идентифицирующих признаков вместе с управлением риском повторной идентификации [3]. Поэтому предложенная схема обозначает такие значения как псевдонимы. При оценке учитывается совокупность полей: редкое сочетание времени, места и характера события может раскрывать принадлежность записи даже без имени.

Простое хеширование телефона также не гарантирует сокрытия. Возможных телефонных номеров существенно меньше, чем произвольных длинных строк, поэтому потенциальные значения могут быть перебраны и сопоставлены с хешами. Для устойчивой связи событий иногда применим ключевой механизм преобразования, однако его результат остается связывающим идентификатором. Решение о таком преобразовании должно исходить из модели угроз. Если сопоставление событий по человеку не требуется, более надежной минимизацией становится отсутствие соответствующего поля в журнале.

Защита хранения и минимизация решают разные задачи. Шифрование ограничивает чтение при определенных сценариях доступа к носителю, но не сокращает объем накопленных сведений и не препятствует чтению законно авторизованным приложением. OWASP рассматривает выбор шифрования и обращение с ключами как отдельные элементы защиты [4]. В предлагаемой схеме сначала определяется необходимый состав записи, затем выбирается защита хранения. Ключи и секреты не должны попадать в тот же журнал, который они предназначены защищать.

Доступ к журналам должен соответствовать рабочей роли. Специалисту диагностики может быть достаточно обезличенного описания последовательности событий, а доступ к таблице соответствия идентификаторов нужен только для обоснованного расследования. Каталог мер NIST SP 800-53 включает управление доступом и аудит действий [5]. Для проектируемой схемы предлагается фиксировать обращения к расширенным сведениям и регулярно пересматривать права. Наличие технической учетной записи не должно автоматически означать разрешение просматривать все категории журналов.

Условный сценарий рассматривает отказ при отправке уведомления. Запись полного запроса раскрывает адрес получателя и содержание сообщения. Альтернативная запись содержит номер отправки, состояние «ошибка соединения» и идентификатор компонента. Если диагностика подтверждает проблему соединения, личные сведения не требуются. Если позднее обнаруживается

ошибка адреса, уполномоченный сотрудник открывает исходную карточку. Пример демонстрирует разделение задач, но не доказывает одинаковую достаточность сокращенного журнала для всех возможных неисправностей.

Жизненный цикл сведений охватывает не только рабочий журнал. Записи могут попадать в резервные копии, архивы и временные выгрузки. Предлагается устанавливать срок и основание хранения для каждого такого места, а также проверять фактическое удаление по выбранной политике. Бессрочное сохранение «на всякий случай» не является техническим требованием диагностики. Вместе с тем слишком раннее удаление способно сделать расследование невозможным, поэтому срок определяется совместно с владельцем процесса и ответственным за защиту данных.

Проверка схемы проводится на тестовых событиях с узнаваемыми искусственными реквизитами. После прохождения обычных и аварийных сценариев анализируются все выходные журналы, включая сообщения библиотек и трассировки исключений. Проверяется отсутствие секретов и достаточность информации для воспроизведения причины отказа. Отдельно контролируется защита от внедрения управляющих символов в запись. Измеряемым результатом испытания становится число обнаруженных лишних полей и доля диагностических сценариев, для которых сохраненных сведений оказалось достаточно.

Исследование предлагает разделять описание технического события и идентификацию участника процесса. Диагностическая запись должна иметь заранее определенный состав, а получение расширенных сведений — отдельное основание и проверку доступа. Анализ показывает, что псевдонимизация, хеширование и шифрование не заменяют минимизацию и требуют оценки с учетом возможностей сопоставления данных.

Практический результат состоит в схеме контроля содержания, доступа и жизненного цикла журналов. Ее эффективность требует испытаний на типовых неисправностях конкретного приложения. Дальнейшая работа должна оценить, какие поля действительно необходимы для диагностики и как изменение состава записей влияет на время восстановления. До такой оценки нельзя утверждать, что удаление любого персонального реквизита не повлияет на работу сопровождения. Связь журналирования с наблюдаемостью информационной системы рассмотрена в статье настоящего сборника [6].

Список литературы

1. National Institute of Standards and Technology. NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management. Version 1.0. — 2020. — DOI: 10.6028/NIST.CSWP.01162020. — URL: https://www.nist.gov/privacy-framework/privacy-framework (дата обращения: 02.09.2026). — Текст : электронный.

2. OWASP Foundation. Logging Cheat Sheet // OWASP Cheat Sheet Series. — URL: https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html (дата обращения: 02.09.2026). — Текст : электронный.

3. National Institute of Standards and Technology. De-Identifying Government Datasets: Techniques and Governance : NIST SP 800-188. — 2023. — DOI: 10.6028/NIST.SP.800-188. — URL: https://csrc.nist.gov/pubs/sp/800/188/final (дата обращения: 02.09.2026). — Текст : электронный.

4. OWASP Foundation. Cryptographic Storage Cheat Sheet // OWASP Cheat Sheet Series. — URL: https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html (дата обращения: 02.09.2026). — Текст : электронный.

5. National Institute of Standards and Technology. Security and Privacy Controls for Information Systems and Organizations : NIST SP 800-53, Revision 5. — 2020. — DOI: 10.6028/NIST.SP.800-53r5. — URL: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final (дата обращения: 02.09.2026). — Текст : электронный.

6. Себто, Д. В. Контроль актуальности данных как часть наблюдаемости информационной системы / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 30–33.

Контроль актуальности данных как часть наблюдаемости информационной системы

Пользователь может открыть отчет без задержек и увидеть таблицу, которая выглядит корректно, но содержит вчерашние сведения. В этой ситуации сервер работает, соединение с базой установлено, а обычная проверка доступности показывает положительный результат. Тем не менее информационная задача не выполнена: решение принимается на основании устаревшего состояния. Подобные ситуации возникают при периодическом обмене между учетной программой, промежуточным хранилищем и пользовательским интерфейсом.

Цель исследования заключается в определении показателей, позволяющих отличать техническую доступность от актуальности и полноты данных. Метод включает анализ материалов по наблюдаемости и потоковой обработке, а также рассмотрение условной схемы обмена. Промышленная система не исследовалась. В подходе Google SRE мониторинг ориентируется на характеристики работы сервиса и последствия для пользователя [1]. В настоящей работе эта логика применяется к цепочке доставки сведений, где отсутствие технической ошибки еще не означает своевременного завершения полезного результата.

Предлагается описывать путь данных от возникновения события до его появления в отчете. Для каждой записи или согласованного пакета сохраняются время события, время приема и время публикации. Эти отметки отвечают на разные вопросы. Время запуска задачи показывает активность программы, но не подтверждает появление новых сведений. Время изменения файла также может обновиться при повторной выгрузке старого содержимого. Поэтому показатель актуальности должен быть связан с содержанием и подтвержденной границей обработанных событий.

Прослеживание отдельной операции помогает определить место задержки. В OpenTelemetry трасса описывает связанные действия, представленные отдельными интервалами выполнения [2]. Для обмена данными предлагается связывать запуск выгрузки, прием пакета и обновление пользовательского представления одним идентификатором. Такое сопоставление показывает, произошла ли задержка у источника, при передаче или при обработке. При этом длительность технической трассы не всегда включает ожидание следующего запуска по расписанию, поэтому ожидание учитывается отдельно.

Условный пример включает выгрузку из учетной системы раз в четыре часа и независимый импорт с таким же интервалом. Событие, возникшее сразу после выгрузки, может ждать следующего запуска почти четыре часа. Если импорт прошел незадолго до появления нового файла, к ожиданию добавляется еще почти четыре часа. При исправной работе обоих этапов суммарная

задержка тогда приближается к восьми часам без учета обработки. Это верхняя оценка ожидания для заданной схемы, а не среднее время любого обмена.

Если импорт запускается непосредственно после завершения выгрузки, второй интервал ожидания уменьшается. Однако изменение расписания само по себе не гарантирует полноту: выгрузка может завершиться частично, а уведомление о готовности — появиться преждевременно. Предлагается публиковать пакет только после фиксации признака завершения и контрольных сведений. Получатель должен различать отсутствие нового пакета, повтор прежнего пакета и получение нового завершенного пакета. Для каждого состояния требуется собственное объяснение в журнале.

Различие между временем события и временем обработки подробно рассматривается в модели Dataflow [3]. Для прикладного отчета это важно при позднем поступлении сведений. Операция, совершенная утром, может попасть в систему вечером и изменить уже рассчитанный итог. Поэтому момент последнего приема не следует считать границей полноты. Даже одна свежая запись способна создать впечатление актуальности, хотя сведения за несколько предыдущих часов отсутствуют. Контроль должен опираться на подтвержденное покрытие периода, если источник может его предоставить.

При потоковой обработке используются оценки продвижения времени и механизмы обработки запаздывающих событий. Руководство Apache Beam описывает водяные знаки и настройку окон обработки [4]. Для рассматриваемой системы предлагается переносить не конкретную технологию, а принцип явной работы с запаздыванием. Отчет должен иметь политику пересчета при поздних событиях и статус предварительности. Водяной знак без достаточных гарантий источника нельзя трактовать как абсолютное доказательство того, что более ранние события никогда не поступят.

Показатель возраста данных определяется относительно подтвержденной границы, а не относительно последней успешной технической проверки. Если источник подтверждает полноту до 12:00, а отчет открыт в 14:30, возраст подтвержденного состояния составляет два часа тридцать минут. Если источник такой границы не предоставляет, система должна показывать ограничение оценки. Подмена неизвестной полноты максимальной датой одной записи создает ложную уверенность и затрудняет обнаружение пропусков между пакетами.

Отдельно задается допустимая задержка для пользовательского процесса. Справочник оборудования и оперативные сведения об остатках могут требовать разных уровней актуальности. Поэтому универсальное правило «данные старше часа считаются ошибкой» не предлагается. Порог определяется тем, в течение какого времени устаревшее значение еще пригодно для конкретного решения. Для оценки полезно измерять долю запросов, при которых возраст

сведений укладывался в согласованную границу, сохраняя возможность отдельно увидеть редкие длительные задержки.

Предупреждение должно возникать при нарушении значимого условия и содержать действие для ответственного специалиста. Практика оповещений на основе целей уровня сервиса связывает сигналы с расходованием допустимого бюджета ошибок [5]. Для периодического обмена предлагается учитывать ожидаемое расписание и допустимое опоздание. Пустой пакет в день без операций не обязательно означает сбой, а успешный запуск без обновления не обязательно означает норму. Состояние определяется вместе с ожидаемой активностью источника.

Проверка подхода включает искусственную задержку пакета, пропуск части записей, повтор старого файла и поступление события задним числом. Оценивается, отражает ли интерфейс фактический статус данных и правильно ли определяется проблемный этап. Система наблюдения также может отказать, поэтому контролируется поступление самих диагностических сигналов. Результаты испытаний выражаются во времени обнаружения и доле выявленных сценариев. Они не должны заменяться общим утверждением, что наличие панели мониторинга обеспечивает надежность обмена.

Исследование предлагает раздельно оценивать доступность компонентов, своевременность доставки и полноту покрытия периода. Результатом является набор связанных временных отметок и статусов, который позволяет объяснить возраст сведений в пользовательском отчете. Условный пример показывает, что независимые расписания могут увеличивать задержку даже при отсутствии технических отказов.

Применение подхода ограничено возможностями источника подтверждать завершенность выгрузки и границу полноты. При отсутствии таких сведений неопределенность должна сохраняться в представлении результата. Дальнейшая проверка может сопоставить время обнаружения нарушений при обычном контроле доступности и при использовании предложенных показателей. Количественное преимущество требует испытаний на одинаковом наборе сценариев. Проверка качества сведений на этапе загрузки рассмотрена в статье о переносе данных из электронных таблиц настоящего сборника [6]. Ограничение состава технических журналов рассмотрено в статье о минимизации персональных данных настоящего сборника [7].

Список литературы

1. Ewaschuk, R. Monitoring Distributed Systems // Site Reliability Engineering: How Google Runs Production Systems / ed. by B. Beyer [et al.]. — O'Reilly Media, 2016. — URL: https://sre.google/sre-book/monitoring-distributed-systems/ (дата обращения: 02.09.2026). — Текст : электронный.

2. OpenTelemetry. Traces // OpenTelemetry Documentation. — URL: https://opentelemetry.io/docs/concepts/signals/traces/ (дата обращения: 02.09.2026). — Текст : электронный.

3. The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing / T. Akidau, R. Bradshaw, C. Chambers [et al.] // Proceedings of the VLDB Endowment. — 2015. — Vol. 8, no. 12. — P. 1792–1803. — DOI: 10.14778/2824032.2824076.

4. Apache Software Foundation. Beam Programming Guide. — URL: https://beam.apache.org/documentation/programming-guide/ (дата обращения: 02.09.2026). — Текст : электронный.

5. Alerting on SLOs // The Site Reliability Workbook / ed. by B. Beyer, N. R. Murphy, D. K. Rensin [et al.]. — O'Reilly Media, 2018. — URL: https://sre.google/workbook/alerting-on-slos/ (дата обращения: 02.09.2026). — Текст : электронный.

6. Себто, Д. В. Контроль качества данных при переносе сведений из электронных таблиц в информационную систему / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 6–9.

7. Себто, Д. В. Минимизация персональных данных в журналах событий информационной системы / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 26–29.

Критерии выбора модульной архитектуры информационной системы для небольшой команды разработки

При создании информационной системы команда выбирает не только язык программирования, но и способ разделения приложения. Микросервисы часто связываются с возможностью независимого развития частей системы. Однако отдельное развертывание требует настройки сетевого взаимодействия, наблюдения и восстановления после отказов. Для небольшой команды эти работы могут занимать заметную часть доступного времени. Поэтому архитектурный выбор следует связывать с характером изменений и эксплуатации, а не с самим фактом популярности определенного подхода.

Цель исследования состоит в формулировании критериев выбора между модульным монолитом и набором взаимодействующих сервисов. Метод включает сравнительный анализ публикаций и рассмотрение условного проекта учета заявок. Нагрузочное тестирование и оценка конкретной организации не проводились. Принцип выделения модулей по скрываемым проектным решениям сформулирован Д. Парнасом [1]. В настоящей работе он используется для различения логической модульности и физического разделения: четкие границы ответственности возможны и внутри одного развертываемого приложения.

Модульный монолит представляет собой единое приложение, внутренние части которого имеют определенные обязанности и договоренности о взаимодействии. Например, учет заявок, формирование документов и уведомления могут находиться в одном процессе, но обращаться друг к другу через согласованные интерфейсы. Недостаточно просто разнести файлы по папкам: если любая часть напрямую изменяет таблицы другой, фактическая граница отсутствует. Проверяемым признаком модуля становится возможность изменить его внутреннюю реализацию без неконтролируемого изменения остальных частей.

Подход к первоначальной разработке единого приложения обсуждается М. Фаулером как способ уточнить границы предметной области до выделения сервисов [2]. Это наблюдение не доказывает превосходство монолита во всех ситуациях. Для рассматриваемой команды оно задает полезный вопрос: достаточно ли устойчиво понимание процесса, чтобы зафиксировать сетевые договоренности между его частями. Если основные сущности и правила часто меняются совместно, преждевременное разделение может перенести внутреннюю неопределенность в распределенный обмен.

Микросервисная архитектура предполагает отдельные сервисы, которые могут развертываться независимо и организуются вокруг определенных возможностей системы [3]. Такое свойство ценно, если независимость

используется на практике. Когда каждое изменение требует одновременного выпуска всех компонентов, физическое разделение не обеспечивает ожидаемой свободы. Поэтому предлагается измерять долю изменений, требующих согласованного выпуска нескольких частей, и выяснять причины связи. Само число сервисов не является показателем качества архитектуры.

Граница хранения данных определяет сложность обеспечения целостности. В единой базе изменение нескольких связанных записей может выполняться одной транзакцией. После разделения хранилищ прежняя операция превращается в обмен сообщениями и допускает промежуточные состояния. П. Хелланд рассматривает ограничения и особенности построения масштабируемых систем за пределами общей распределенной транзакции [4]. Для прикладного проекта требуется заранее определить, какие промежуточные состояния допустимы и как исправляется частично выполненный процесс.

Условный пример включает создание заявки и назначение обязательного исполнителя. Если запись заявки и назначение должны становиться видимыми одновременно, их разделение между сервисами потребует дополнительного согласования. Если же отправка уведомления допускается немного позднее, ее можно отделить от фиксации заявки. В обоих случаях границу определяет требование процесса. Разделение по названиям таблиц без анализа момента завершения операции способно усложнить систему, не создавая полезной независимости.

Эксплуатационные затраты включают не только серверные ресурсы. Для каждого самостоятельного компонента требуется понимать версию, состояние, конфигурацию и ответственного за восстановление. Ошибка взаимодействия может быть связана с сетью, несовместимым форматом сообщения или доступностью зависимости. Предлагается учитывать время диагностики и выпуска изменения как часть архитектурной оценки. Сравнение только потребления памяти или скорости выполнения одной функции не отражает совокупную нагрузку на команду сопровождения.

Основанием для выделения сервиса может стать подтвержденная потребность независимо масштабировать ресурсоемкую обработку или выпускать изменения с другим ритмом. Например, генерация большого количества отчетов способна требовать отдельного управления нагрузкой. Однако предварительно следует проверить более простые варианты: фоновую задачу, ограничение параллелизма или отдельный рабочий процесс с прежней логической моделью. Эти варианты не равнозначны полноценному выделению независимого сервиса и могут закрыть часть эксплуатационных потребностей.

Для принятия решения предлагается описывать конкретное ограничение существующей архитектуры, ожидаемое улучшение и способ проверки. Если проблема связана с длительным общим выпуском, измеряется время от готовности изменения до его доставки пользователю. Если ограничение связано с

нагрузкой, оценивается выполнение целевых операций при сопоставимом потоке запросов. В решение включается стоимость наблюдения и восстановления. Отсутствие наблюдаемого ограничения означает, что количественное преимущество разделения пока не установлено.

Переход к сервисам может выполняться постепенно. Подход Strangler Fig предусматривает поэтапную замену возможностей существующей системы [5]. Для рассматриваемой задачи предлагается сначала закрепить внутренний интерфейс модуля, затем обеспечить наблюдение за обращениями и только после этого переносить реализацию. Во время перехода определяется единственный ответственный источник для изменяемых данных. Две независимо редактируемые копии без правил согласования создают отдельную проблему, которую нельзя считать временной технической мелочью.

Обратимость решения также относится к критериям качества. Экспериментальное выделение компонента должно сопровождаться условием прекращения или пересмотра, если ожидаемая независимость не достигнута. При этом возврат не означает автоматическое объединение баз без анализа: потребуется восстановить целостность и историю изменений. Предлагаемый подход не назначает универсального количества разработчиков, после которого необходимы микросервисы. Размер команды является контекстом, а решающими становятся реальные зависимости, требования к выпуску и способность поддерживать распределенную систему.

Анализ показывает, что модульность и распределенность представляют разные свойства архитектуры. Для небольшой команды исходным предметом оценки должны быть границы ответственности, совместность изменений и требования к целостности данных. Результатом исследования является порядок выделения сервисов по подтвержденным ограничениям с заранее определенными показателями полезности и эксплуатационной стоимости.

Модульный монолит остается обоснованным вариантом при тесно связанных операциях и отсутствии потребности в независимом развертывании. Отдельные сервисы могут быть оправданы устойчивыми границами и различающимися требованиями к работе компонентов. Проверка этих выводов требует наблюдения за конкретным проектом. Дальнейшее исследование должно сопоставлять не архитектурные названия, а время изменений, отказов и восстановления при одинаковых функциональных требованиях. Вопросы согласования данных при автономной работе без постоянного подключения к сети рассмотрены в статье настоящего сборника [6].

Список литературы

1. Parnas, D. L. On the Criteria to Be Used in Decomposing Systems into Modules // Communications of the ACM. — 1972. — Vol. 15, no. 12. — P. 1053–1058. — DOI: 10.1145/361598.361623.

2. Fowler, M. Monolith First. — 2015. — URL: https://martinfowler.com/bliki/MonolithFirst.html (дата обращения: 02.09.2026). — Текст : электронный.

3. Lewis, J. Microservices / J. Lewis, M. Fowler. — 2014. — URL: https://martinfowler.com/articles/microservices.html (дата обращения: 02.09.2026). — Текст : электронный.

4. Helland, P. Life beyond Distributed Transactions: An Apostate's Opinion // Proceedings of CIDR 2007. — 2007. — P. 132–141. — URL: https://www.cidrdb.org/cidr2007/papers/cidr07p15.pdf (дата обращения: 02.09.2026). — Текст : электронный.

5. Fowler, M. Strangler Fig. — URL: https://martinfowler.com/bliki/StranglerFigApplication.html (дата обращения: 02.09.2026). — Текст : электронный.

6. Себто, Д. В. Согласование данных мобильной информационной системы при работе без постоянного подключения к сети / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 18–21.

Обеспечение воспроизводимости документов, автоматически формируемых информационной системой

Автоматическое формирование уведомлений, актов и отчетов сокращает ручной перенос реквизитов. Однако через несколько месяцев может потребоваться объяснить, почему в документе указаны определенный адрес, сумма или должность подписанта. Если система хранит только текущие карточки и последнюю редакцию шаблона, повторное формирование даст другой результат. Наличие сохраненного файла позволяет увидеть ранее выпущенный документ, но не всегда позволяет установить, из каких сведений и по каким правилам он был получен.

Цель исследования заключается в разработке схемы воспроизводимого формирования документов в прикладной информационной системе. Использованы анализ моделей происхождения данных и принципов воспроизводимых вычислений, а также разбор условного изменения реквизитов. Практический эксперимент не проводился. Рекомендации Г. Сандве и соавторов связывают воспроизводимость с сохранением данных и последовательности вычислительных действий [1]. В настоящей работе эти положения применяются к документу как результату обработки, который должен оставаться объяснимым после изменения исходной базы и программного окружения.

Прежде всего требуется определить, какой результат должен воспроизводиться. Содержательное совпадение означает одинаковые факты и расчетные значения. Визуальное совпадение дополнительно требует одинакового расположения элементов, переносов и страниц. Побайтовое совпадение означает полное равенство файлов. Эти уровни различаются: два документа могут выглядеть одинаково, но содержать разные служебные даты внутри формата. Поэтому требование «сформировать тот же документ» должно уточняться до выбора способа хранения и проверки.

Предлагается сохранять неизменяемый снимок сведений, использованных при выпуске. В него входят только необходимые реквизиты и результаты расчетов с установленной точностью. Текущая карточка объекта остается рабочим источником для будущих документов, но не заменяет историческое состояние. Если исправление затрагивает уже выпущенный документ, создается новая версия со связью с предыдущей. Молчаливое изменение исходного снимка уничтожило бы возможность объяснить ранее полученный результат.

Модель W3C PROV-DM описывает происхождение через сущности, действия и участников [2]. В рассматриваемой схеме сущностями являются снимок данных, шаблон и итоговый файл, действием — формирование, а участником — инициировавший его пользователь или программный компонент. Такая запись отвечает на вопросы о том, что использовалось и что получилось. Она не

обязана превращаться в сложную самостоятельную платформу: для ограниченного процесса достаточно согласованной структуры связей в основной базе.

Шаблон также является изменяемым входом. Если в нем меняется расположение подписи или условие включения абзаца, тот же набор реквизитов может дать другой документ. Поэтому сохраняются идентификатор и версия шаблона, а при необходимости — его исходный файл. Версия программы формирования фиксируется отдельно. Запись только даты выпуска недостаточна, поскольку несколько разных вариантов шаблона могут использоваться в один день разными подразделениями или параллельными задачами.

Условный пример связан с изменением должности подписанта. Документ сформирован в марте с должностью «начальник отдела», а в апреле карточка изменена на «директор подразделения». Повторное обращение к текущей карточке подменяет исторический реквизит. При использовании снимка мартовский документ сохраняет прежнее содержание, а апрельский получает новое. Это различие должно быть предусмотрено явно. Сохранение истории не означает, что ошибочный реквизит запрещено исправлять: исправление оформляется отдельной версией с указанием причины.

Для контроля целостности итогового файла может сохраняться криптографическая контрольная сумма. Стандарт FIPS 180-4 определяет семейство алгоритмов защищенного хеширования [3]. Сравнение суммы с доверенно сохраненным значением позволяет выявлять изменение байтов. Однако сумма сама по себе не подтверждает личность автора и не доказывает достоверность содержания. Если злоумышленник способен заменить одновременно файл и его контрольное значение, простое сравнение не обнаружит подмену. Поэтому доверие к месту хранения является отдельным условием.

Побайтовая воспроизводимость требует учета служебных полей и порядка записи частей формата. Для XML существуют правила канонического представления, устраняющие некоторые различия записи эквивалентного содержимого [4]. Но применение такой процедуры к одному XML-фрагменту не гарантирует совпадения всего файла DOCX. Предлагается проверять побайтовое равенство только там, где оно действительно требуется. Для большинства задач объяснения документа более существенны неизменность исходных реквизитов и доступность сохраненного оригинала результата.

Визуальная воспроизводимость зависит от среды формирования. Отсутствующий шрифт может быть заменен другим, из-за чего изменятся переносы и число страниц. На результат также влияют версия механизма преобразования и настройки печати. Поэтому схема предусматривает фиксацию используемой среды и сохранение окончательного представления документа. При повторном формировании сравниваются не только извлеченные строки, но и страницы. Равенство текста не обнаруживает подпись, перемещенную на отдельный лист, или обрезанную строку таблицы.

Связи происхождения могут передаваться между компонентами в согласованной форме. PROV-O предоставляет онтологическое представление соответствующих понятий [5]. Для прикладной системы его использование оправдано, когда происхождение требуется сопоставлять между несколькими сервисами. Если обмена нет, достаточно внутренней модели, сохраняющей тот же смысл связей. Важна не сложность формата, а возможность однозначно определить входы и выходы формирования, а также отличить новую версию документа от повторной выдачи уже существующего файла.

Проверка схемы проводится на заранее подготовленных случаях изменения данных, шаблона и программной среды. Для каждого случая задается ожидаемый уровень совпадения. Система должна восстановить исторические реквизиты, обнаружить изменение итогового файла и показать причину появления новой версии. Отдельно проверяется сценарий недоступности старой среды: даже если повторное выполнение невозможно, сохраненный результат и сведения о его происхождении должны оставаться доступными. Ограничения хранения определяются назначением документов и принятой политикой доступа.

Исследование предлагает рассматривать автоматически сформированный документ как результат определенной версии данных, шаблона и процедуры обработки. Практический результат состоит в схеме фиксации этих связей и разграничении уровней воспроизводимости. Контрольная сумма защищает проверку целостности в пределах доверенного хранения, а журнал происхождения объясняет получение результата; ни один из этих механизмов по отдельности не решает обе задачи.

Предложенная схема требует проверки на реальных шаблонах и средствах формирования. До испытаний нельзя обещать побайтовое или визуальное совпадение при произвольной смене программной среды. Дальнейшее исследование может оценить объем дополнительного хранения и время восстановления документа. Эти затраты следует сопоставлять с требованиями к объяснимости и сохранности конкретного документного процесса. Вопросы прослеживаемости ответов, формируемых языковой моделью, рассмотрены в статье настоящего сборника [6].

Список литературы

1. Ten Simple Rules for Reproducible Computational Research / G. K. Sandve, A. Nekrutenko, J. Taylor, E. Hovig // PLOS Computational Biology. — 2013. — Vol. 9, no. 10. — e1003285. — DOI: 10.1371/journal.pcbi.1003285.
2. World Wide Web Consortium. PROV-DM: The PROV Data Model : W3C Recommendation, 30 April 2013. — URL: https://www.w3.org/TR/prov-dm/ (дата обращения: 02.09.2026). — Текст : электронный.

3. National Institute of Standards and Technology. Secure Hash Standard (SHS) : FIPS PUB 180-4. — 2015. — DOI: 10.6028/NIST.FIPS.180-4. — URL: https://csrc.nist.gov/pubs/fips/180-4/upd1/final (дата обращения: 02.09.2026). — Текст : электронный.

4. World Wide Web Consortium. Canonical XML Version 1.1 : W3C Recommendation, 2 May 2008. — URL: https://www.w3.org/TR/xml-c14n11/ (дата обращения: 02.09.2026). — Текст : электронный.

5. World Wide Web Consortium. PROV-O: The PROV Ontology : W3C Recommendation, 30 April 2013. — URL: https://www.w3.org/TR/prov-o/ (дата обращения: 02.09.2026). — Текст : электронный.

6. Себто, Д. В. Обеспечение проверяемости ответов искусственного интеллекта при поиске по документам организации / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 10–13.

Предотвращение ошибок пользователей посредством доступных электронных форм

Электронная форма связывает человека с правилами информационной системы. Пользователь вводит адрес, выбирает услугу или описывает неисправность, а программа преобразует эти действия в запись. Если требование к полю непонятно, человек может указать формально допустимое, но неподходящее значение. Поэтому качество входных данных зависит не только от проверок базы, но и от того, насколько ясно интерфейс объясняет задачу. Неудачное сообщение об ошибке способно привести к повторной отправке или отказу от заполнения.

Цель исследования заключается в определении требований к форме, которые помогают предупреждать ошибки и поддерживают разные способы взаимодействия. Метод включает анализ рекомендаций W3C и принципов удобства интерфейсов с последующим построением сценариев проверки. Испытание с реальными пользователями не проводилось. WCAG 2.2 задает проверяемые критерии доступности веб-содержимого [1]. В настоящей работе доступность рассматривается совместно с понятностью процесса: техническая возможность перейти к полю не гарантирует, что его назначение и требуемое значение будут правильно поняты.

Подпись поля должна обозначать смысл информации, а не внутреннее имя параметра. Например, формулировка «дата выполнения работы» точнее, чем «дата», если в процессе существует несколько временных отметок. Подсказка формата дополняет подпись, но не заменяет ее. После начала ввода пользователь должен по-прежнему видеть, какой реквизит заполняется. Руководство WAI по формам связывает подписи, инструкции и сообщения с доступностью взаимодействия [2]. Предлагается проверять эти элементы как единую последовательность, а не как независимые декоративные части.

Обязательность реквизита должна объясняться до отправки. При этом сама потребность в поле требует проверки: если сведения не используются для выполнения задачи, их запрос увеличивает объем работы и вероятность ошибки. В предлагаемом подходе для каждого поля фиксируется назначение и момент, когда информация действительно необходима. Сокращение формы не означает безусловного удаления всех необязательных реквизитов. Дополнительное описание может быть полезным, если его назначение понятно и отсутствие не препятствует продолжению.

Проверка формата должна помогать исправлению, сохраняя введенное значение. Сообщение «неверные данные» не указывает, что именно произошло. Более полезно назвать поле, объяснить требование и показать допустимый пример. Если система требует дату в определенном порядке, она должна явно

описать этот порядок или предоставить однозначный выбор. Автоматическое преобразование неоднозначного ввода без подтверждения может создавать скрытую ошибку: результат выглядит корректно, но соответствует другому намерению пользователя.

Момент проверки также влияет на понятность. Ошибка, появляющаяся до завершения ввода, может восприниматься как неисправность формы. Например, короткий промежуточный фрагмент адреса еще не является окончательным значением. Предлагается различать подсказку во время заполнения и окончательное отклонение при завершении поля или отправке. Принципы Я. Нильсена связывают предотвращение ошибок с понятной обратной связью и возможностью восстановления [3]. Для каждого сценария должно быть ясно, какое действие позволяет продолжить работу.

Доступность с клавиатуры необходима для пользователей, которые не работают мышью, и полезна при последовательном вводе. Порядок переходов должен соответствовать смыслу формы, а положение фокуса — оставаться заметным. Ошибку нельзя обозначать только изменением цвета, поскольку это не обеспечивает одинаковое восприятие. Для сообщений, появляющихся динамически, требуется учитывать работу вспомогательных технологий. Эти требования проверяются в реальном интерфейсе: правильная визуальная картинка не доказывает доступности управляющих элементов.

Поддержка понимания включает согласованные термины и предсказуемые действия. Документ W3C о доступности для людей с когнитивными и учебными особенностями рассматривает ясность содержания и помощь при выполнении задач [4]. Для прикладной формы предлагается использовать одно название действия на всем пути заполнения. Если кнопка называется «Сохранить черновик», ее результат не должен выглядеть как окончательная отправка. Состояние после действия должно прямо сообщать, что сохранено и требуется ли дальнейшее участие пользователя.

Условный сценарий касается заявки на обслуживание. Пользователь заполняет адрес и описание, но пропускает обязательную дату. При отправке форма сообщает об ошибке рядом с датой, связывает сообщение с полем и сохраняет остальные значения. Дополнительно общий блок ошибок позволяет перейти к проблемному месту. Если вместо этого страница очищается, человек вынужден повторять уже выполненную работу. Пример показывает проектное различие между вариантами, но не содержит измеренного результата снижения отказов или ускорения заполнения.

Сохранение введенного не должно приводить к незаметному выполнению действия дважды. После нажатия кнопки пользователь получает состояние обработки, а после завершения — однозначное подтверждение с идентификатором заявки. При сетевом сбое нельзя автоматически утверждать, что заявка не создана: ответ мог потеряться после успешной обработки. Поэтому форма

должна поддерживать проверку статуса и согласованное повторение операции. Таким образом, понятность интерфейса зависит и от поведения серверной части, а не только от текста сообщений.

Оценка проводится по завершению задачи, числу исправлений и времени выполнения, но эти показатели требуют совместной интерпретации. Быстрое заполнение с неверно понятым адресом не является успешным результатом. Для каждого задания заранее определяется правильный смысловой итог. WAI рекомендует участие пользователей в оценке доступности, дополняющее проверку требований [5]. Предлагается включать людей с разными способами взаимодействия и опытом работы. Небольшая группа способна выявить проблемы, но не подтверждает отсутствие затруднений у всей аудитории.

Автоматическая проверка помогает находить часть технических нарушений, однако не устанавливает понятность формулировки или достаточность инструкции. Поэтому протокол объединяет анализ разметки, ручную проверку клавиатурой и наблюдение за выполнением заданий. До испытания фиксируются условия, версия формы и критерии ошибки. При сравнении вариантов учитывается эффект обучения: человек, уже знающий задачу, может заполнить второй вариант быстрее независимо от его качества. Порядок предъявления и состав участников должны отражаться при интерпретации результатов.

Проведенный анализ позволяет рассматривать доступную форму как средство предупреждения ошибок на этапе возникновения данных. Предложенный подход объединяет ясность назначения полей, своевременную обратную связь, сохранение введенного и понятное подтверждение результата. Практический вклад состоит в протоколе проверки, который различает техническую допустимость значения и правильное понимание пользовательской задачи.

Соответствие отдельным требованиям доступности не гарантирует полной понятности формы, а успешное заполнение одним человеком не подтверждает пригодности для всех пользователей. Дальнейшее исследование должно включать испытания с представителями целевой аудитории и сопоставление конкретных вариантов интерфейса. Количественное снижение ошибок может быть заявлено только после такого сравнения при заранее определенных условиях. Контроль качества сведений при переносе их в систему рассмотрен в статье настоящего сборника [6].

Список литературы

1. World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 2.2 : W3C Recommendation. — URL: https://www.w3.org/TR/WCAG22/ (дата обращения: 02.09.2026). — Текст : электронный.
2. World Wide Web Consortium. Forms Tutorial // Web Accessibility Initiative. — URL: https://www.w3.org/WAI/tutorials/forms/ (дата обращения: 02.09.2026). — Текст : электронный.

3. Nielsen, J. 10 Usability Heuristics for User Interface Design // Nielsen Norman Group. — URL: https://www.nngroup.com/articles/ten-usability-heuristics/ (дата обращения: 02.09.2026). — Текст : электронный.
4. World Wide Web Consortium. Making Content Usable for People with Cognitive and Learning Disabilities : Working Group Note, 29 April 2021. — URL: https://www.w3.org/TR/coga-usable/ (дата обращения: 02.09.2026). — Текст : электронный.
5. World Wide Web Consortium. Involving Users in Evaluating Web Accessibility // Web Accessibility Initiative. — URL: https://www.w3.org/WAI/test-evaluate/involving-users/ (дата обращения: 02.09.2026). — Текст : электронный.
6. Себто, Д. В. Контроль качества данных при переносе сведений из электронных таблиц в информационную систему / Д. В. Себто // Информационные системы и технологии: качество данных, надёжность и практические решения : сборник статей. — СПб.: Наукоемкие технологии, 2026. — С. 6–9.