

Использование инструментов и практик разработчиков для управления содержанием проекта



Лобзов Алексей Владимирович^{1,2}

MBA, руководитель направления развития компетенции системного анализа¹,
ментор программы магистратуры «Системный анализ и управление на основе данных»²

ORCID: 0000-0002-7146-1715

e-mail: aleks.lobzov@gmail.com

¹Акционерное общество «Альфа-Банк», г. Москва, Россия

²Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации, г. Москва, Россия

Ключевые слова: Docs as Code, DocOps, инструменты разработчиков, практики разработчиков, система контроля версий, система управления проектами, система управления требованиями, управление содержанием, управление требованиями

Цитирование: Лобзов А.В. Использование инструментов и практик разработчиков для управления содержанием проекта // Вестник проектного управления. 2025. Т. 1, № 3. С. 21-28

Аннотация

Объектом исследования, представленного в работе, является деятельность по управлению проектами в сфере информационных технологий (далее – ИТ), предметом – инструменты, используемые для управления содержанием таких проектов. Под содержанием в первую очередь понимаются требования к продукту, сервису или иному результату проекта. Обозначены проблемы, с которыми сталкиваются организации при использовании наиболее популярных систем управления требованиями – локальных электронных документов в связке с почтовыми системами, а также систем типа wiki. К таковым относятся, в частности, противоречивость и неконсистентность требований. В качестве одного из решений рассмотренных проблем выделены подход Docs as Code (англ. Documentation as Code – документация как программный код) и практики DocOps (англ. Documentation Operations – операции с документацией), которые вместе определяют набор инструментов и практик разработчиков для управления содержанием проектов в сфере ИТ.



Project scope management using developers' tools and practices



Alexey V. Lobzov^{1,2}

MBA, Competency Leader for Systems Analysis¹, Mentor of the Master's Programme "Systems Analysis and Data-Driven Management"²

ORCID: 0000-0002-7146-1715

e-mail: aleks.lobzov@gmail.com

¹Joint Stock Company "Alfa-Bank", Moscow, Russia

²Russian Presidential Academy of National Economy and Public Administration, Moscow, Russia

Keywords: Docs as Code, DocOps, developers' tools, developers' practices, version control system, project management system, requirements management system, scope management, requirements management

For citation: Lobzov A.V. (2025) Project scope management using developers' tools and practices. Vestnik proektnogo upravleniya, v. 1, no. 3, pp. 21-28.

Abstract

The article focuses on project management activities in the field of information technology (hereinafter referred to as IT) which is the object of the study, and the tools used to manage the scope of such projects (the subject). Scope primarily refers to requirements for a product, service, or other project deliverable. The article identifies the challenges organisations face when using the most popular requirements management systems – local electronic documents in conjunction with email systems as well as wiki-type systems. These include, in particular, contradiction and inconsistency in requirements. The Docs as Code approach and DocOps (Documentation Operations) practices are highlighted as solutions to these challenges, defining a set of developers' tools and practices to manage the scope of IT projects.



ВВЕДЕНИЕ

Что такое содержание проекта с точки зрения проектного управления? С одной стороны, это множество работ, которые необходимо выполнить, чтобы создать продукт, сервис или иной результат с заданными характеристиками. С другой – это множество самих характеристик продукта, сервиса или иного результата, который должен быть создан в процессе выполнения проекта. Фактически первое является производным от второго – множество работ проекта во многом определяется множеством характеристик продукта проекта.

Для управления работами создан целый ряд специализированных инструментов – системы управления проектами и задачами. В качестве примеров популярных иностранных систем можно выделить Microsoft Project and Portfolio Management и Atlassian Jira. Среди отечественных аналогов выделяются ПМ «Форсайт» и АСУ-Инвест от группы компаний «Проектная ПРАКТИКА», а также «Яндекс Трекер».

Характеристики продукта, сервиса или иного результата, создаваемого в процессе выполнения проекта в сфере информационных технологий (далее – ИТ), как правило, выражаются в виде требований. Существует множество специализированных систем управления требованиями¹. Однако, несмотря на большое разнообразие, едва ли можно говорить об их широком применении в индустрии [Горелиц, Кильдишев, Хорошилов, 2019].

Практика показывает, что, в зависимости от типа разработки, – заказная или внутренняя (англ. in-house) – наиболее популярными системами управления требованиями являются локальные электронные документы в связке с почтовыми системами (например, Microsoft Word + Microsoft Outlook), а также wiki-системы (например, Atlassian Confluence и «Яндекс Вики»). Однако использование этих инструментов сопряжено с рядом проблем. Рассмотрим некоторые из них.

ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ ЭЛЕКТРОННЫХ ДОКУМЕНТОВ В СВЯЗКЕ С ПОЧТОВЫМИ СИСТЕМАМИ

Предположим, бизнес-аналитик оформил требования в виде электронного документа. Он должен быть согласован тремя экспертами (ex_1 , ex_2 , ex_3). Согласование проходит по электронной почте.

Спустя некоторое время после отправки на согласование бизнес-аналитик получает ответное письмо от ex_1 . В нем он рекомендует внести ряд правок в предложенные требования. Бизнес-аналитик вносит изменения в соответствии с полученными предложениями и направляет электронный документ на повторное согласование.

Ex_2 видит на почте второе письмо от бизнес-аналитика. Он понимает, что в электронный документ с требованиями

были внесены изменения, не тратит время на первую версию, приступает к согласованию второй.

В свою очередь, ex_3 отвечает на письма последовательно. Он просматривает первую версию электронного документа, его все устраивает. Ex_3 пишет ответ бизнес-аналитику – документ согласован.

Чуть позже ex_3 доходит до второго письма бизнес-аналитика с измененной по предложениям ex_1 версией электронного документа. Эксперт не согласен с внесенными изменениями. Он пишет ответ бизнес-аналитику – документ не согласован.

Бизнес-аналитик ранее получил согласование ex_3 . Теперь его внимание переключается на получение согласования двух других экспертов. В результате если бизнес-аналитик пропускает второй ответ ex_3 , то дальше он будет работать с противоречивыми (требования ex_1 противоречат требованиям ex_3).

Данная проблема вызвана отсутствием единого места хранения требований, с которым могли бы работать все заинтересованные стороны, – бизнес-аналитик и три эксперта – что является одним из ограничений системы управления требованиями в виде локальных электронных документов в связке с почтовыми системами. Перечень иных ограничений представлен в классических трудах по управлению требованиями к программному обеспечению (далее – ПО) [Wiegers, Beatty, 2013].

Одним из решений описанной выше проблемы является переход на управление требованиями с использованием wiki-систем. Однако данный класс систем также имеет свои недостатки.

ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ WIKI-СИСТЕМ

Предположим, бизнес-аналитик оформил требования в виде wiki-страницы. Документ должен быть согласован тремя экспертами. Согласование проходит на wiki-странице с использованием комментариев – эксперты оставляют их в случае несогласия с текущими формулировками, бизнес-аналитик вносит изменения в соответствии с оставленными комментариями.

Требования и комментарии к ним находятся в едином месте – на wiki-странице. Каждый участник согласования имеет доступ только к одной версии – текущей, отображаемой на странице. Все замечания и рекомендации экспертов накапливаются в комментариях и являются дополнительным источником данных для проверки требований на непротиворечивость.

Пусть добавление каждого требования rq_i приводит к созданию новой версии v_i wiki-страницы. Тогда в первой версии v_1 будет сохранено rq_1 , во второй v_2 – rq_1 и rq_2 , а в третьей v_3 – rq_1 , rq_2 и rq_3 .

Предположим, бизнес-аналитик выявил, что требование rq_2 устарело, его необходимо исключить. Как это сделать с использованием wiki-системы?

Одна из распространенных ловушек при работе с требованиями – неадекватная система контроля версий.

Одним из возможных вариантов являются удаление rq_2 из текущей версии (v_3) и создание в результате выполнения этой операции новой версии (v_4), включающей только rq_1 и rq_3 . Удаление требования будет производиться бизнес-аналитиком вручную. В результате возникает риск нарушения консистентности всех требований по причине того, что они могут быть взаимосвязаны.

Второй возможный вариант состоит в откате к v_1 , добавлении rq_3 и создании новой версии. Откат к предыдущей версии обычно выполняется автоматически средствами wiki-системы, а добавление требования – вручную бизнес-аналитиком. В результате снова возникает риск нарушения консистентности всех требований.

Одной из распространенных ловушек при работе с требованиями является неадекватная система контроля версий². Обойти ее позволят применение подхода Docs as Code (англ. Documentation as Code – документация как программный код) и практик DocOps (англ. Documentation Operations – операции с документацией). Рассмотрим, в чем они состоят.

ПОДХОД DOCS AS CODE И ПРАКТИКИ DOCOPS

Под Docs as Code принято понимать подход к написанию документации с использованием тех же инструментов, что и при написании кода³, в том числе систем управления задачами, систем контроля версий, языков текстовой разметки и пр. В качестве примеров систем управления задачами можно назвать Atlassian Jira и «Яндекс Трекер». Примерами систем контроля версий являются Atlassian Bitbucket и GitFlic от «Группы Астра». Популярными языками текстовой разметки – Markdown и AsciiDoc.

В свою очередь, под DocOps понимается множество практик для интеграции команды разработки и команды сопровождения в части процесса разработки документации, в том числе его автоматизации (ревью, тестирование, сборка и публикация и пр.)⁴. Вместе подход Docs as Code и практики DocOps помогают решить проблемы, присущие системам управления требованиями в виде локальных электронных документов в связке с почтовыми системами, а также wiki-систем.

Предположим, бизнес-аналитик оформил требования в виде электронного документа в формате Markdown. Он размещен в системе контроля версий. В первой версии v_1 сохранено rq_1 , во второй v_2 – rq_1 и rq_2 , а в третьей v_3 – rq_1 , rq_2 и rq_3 . Каждое требование создавалось в отдельной ветке (br_1 , br_2 , br_3), которая впоследствии сливалась с главной. Перед бизнес-аналитиком стоит задача исключить устаревшее требование rq_2 .

Для решения поставленной задачи бизнес-аналитик может откатиться на v_1 главной ветки, содержащей rq_1 . Затем

создать запрос на слияние с ней br_3 , содержащей rq_3 . Далее, разрешив конфликты и осуществив слияние средств системы контроля версий, получить новую версию требований, содержащую rq_1 и rq_3 и не содержащую rq_2 .

Применение подхода Docs as Code вместе с практиками DocOps позволяет сократить количество ручных операций, выполняемых при работе с требованиями и, как следствие, количество ошибок, вызванных человеческим фактором. Более того, подход Docs as Code позволяет минимизировать финансовые риски организации [Лобзов, 2023].

Вернемся к тезису о том, что множество работ проекта фактически является производным от множества требований. Рассмотрим, как организовать связь элементов двух множеств.

СВЯЗЬ МЕЖДУ ТРЕБОВАНИЯМИ И РАБОТАМИ ПРОЕКТА

Рассмотрим случай внутренней (in-house) разработки. Бизнес-аналитик получает задачу на разработку требований. Она ведется с использованием системы управления задачами Atlassian Jira.

Пусть в организации используется система контроля версий Atlassian Bitbucket. Оба рассматриваемых инструмента интегрированы между собой – из задачи в Atlassian Jira бизнес-аналитик создает новую ветку (branch) в Atlassian Bitbucket и сохраняет (commit) в ней исходники спецификации требований в виде электронного документа в формате Markdown. В завершение бизнес-аналитик открывает запрос на слияние (pull request) с веткой, в которой ведется базовая версия требований. Открытый запрос ожидает согласования (ревью).

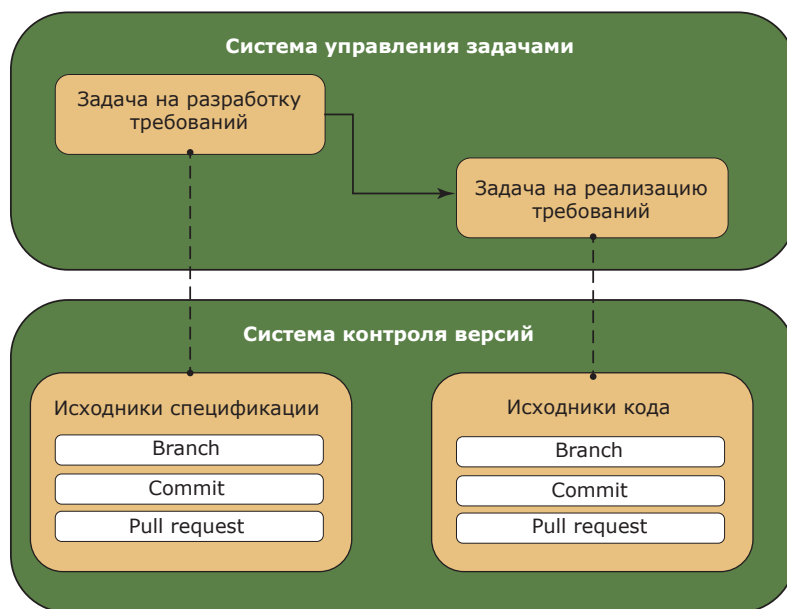
Предположим, в согласовании участвует лишь одно заинтересованное лицо – разработчик. Его задача – убедиться, что требования понятны, непротиворечивы и он сможет их реализовать.

Пусть разработчик согласовал запрос на слияние, а бизнес-аналитик слил исходники спецификации требований с базовой версией. Тогда базовая версия обновится, а в задаче на разработку требований в Atlassian Jira напротив ссылки на запрос отобразится соответствующий статус – «слито» (merged).

Таким образом, бизнес-аналитик выполнил задачу на разработку требований. Ее можно закрыть, переведя в финальный статус, – «завершено» (done).

Теперь разработчик получает задачу на реализацию требований. В ней приведена ссылка на задачу по разработке требований со ссылкой на согласованный ранее самим разработчиком запрос на слияние. Специалист начинает трудиться над исходниками кода аналогично тому, как ранее бизнес-аналитик трудился над исходниками спецификации.

Подход Docs as Code позволит обойти неадекватную систему контроля версий.



Составлено автором по материалам исследования
Рис. 1. Модель связи требований и работ проекта

Описанная выше модель связи требований и работ проекта приведена на рис. 1. Стоит отметить, что данная модель не является универсальной и приведена в качестве одного из возможных вариантов, которые могут быть реализованы при условии применения подхода Docs as Code и практик DocOps.

Остается вопрос: насколько широко используются подход Docs as Code и практики DocOps в отечественных компаниях? Постараемся найти на него ответ.

ТРЕНД НА ИСПОЛЬЗОВАНИЕ ПОДХОДА DOCS AS CODE И ПРАКТИК DOCS OPS В РОССИЙСКОЙ ФЕДЕРАЦИИ

Тренд на использование подхода Docs as Code и практик DocOps в деятельности отечественных компаний возник почти 10 лет назад. В контексте разработки требований одними из первых о них публично заговорили эксперты «Лаборатории Касперского», выступив с докладом на восьмой Международной конференции по системному и бизнес-анализу Analyst Days⁵.

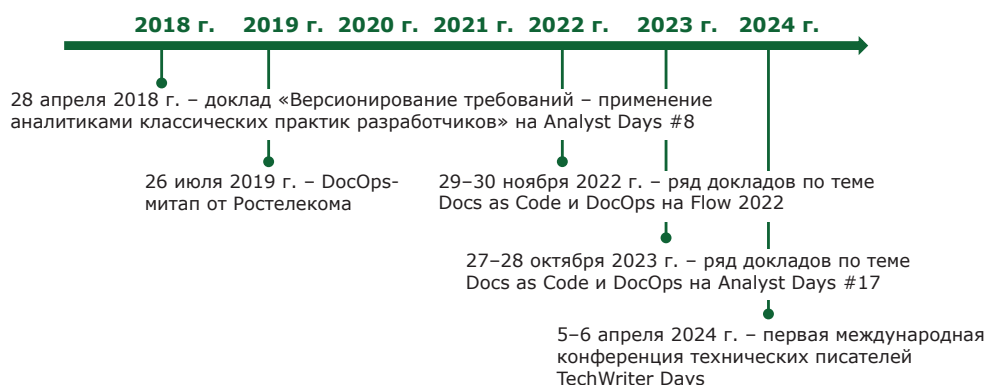
В 2019 г. сотрудниками Ростелекома был организован и успешно проведен один из первых митапов по теме DocOps. После чего случилась эпидемия COVID-19, очные мероприятия были отменены, компаниям пришлось адаптироваться к новой реальности.

В 2022 JUG Ru Group решила запустить новую профессиональную конференцию по системному и бизнес-анализу Flow. На конференции было представлено сразу несколько докладов по теме Docs as Code и DocOps⁶. Там же в процессе одного из обсуждений была приведена оценка, что почти половина российских ИТ-компаний использует подход Docs as Code для разработки документации⁷.

В 2023 г. похожая ситуация наблюдалась на семнадцатой конференции Analyst Days – многие спикеры выступили с докладами на тему Docs as Code и DocOps. В результате организаторы решили запустить отдельную конференцию для обсуждения вопросов и обмена опытом в части разработки технической документации. Так, в 2024 г. прошла первая международная конференция технических писателей TechWriter Days.

Таким образом, можно проследить семилетнее становление тренда на использование подхода Docs as Code и практик DocOps в отечественных организациях (рис. 2).

Однако на рынке существует еще один тренд – повсеместное внедрение искусственного интеллекта (далее – ИИ), в частности генеративного, в деятельность организаций. Данный тренд отлично сочетается с подходом Docs as Code и практиками DocOps для управления требованиями и, следовательно, содержанием проекта.



Составлено автором по материалам источника⁸
Рис. 2. Становление тренда на применение подхода Docs as Code и практик DocOps

ПОЛЬЗА ГЕНЕРАТИВНОГО ИИ

Данная технология уже используется в управлении проектами. Создаются ИИ-агенты – автономные программы, способные выполнять конкретные задачи проектного управления [Гузенко, Балябин, 2025]. Ведется поиск новых областей применения.

В части управления содержанием проекта, в частности требованиями, первые публичные результаты появились в 2022 г.⁹ Было продемонстрировано, как можно использовать ChatGPT (англ. generative pre-trained transformer – генеративный предобученный трансформер) для формирования бэклога продукта и проработки его элементов.

ChatGPT выдает ответы в текстовом виде. Причем взаимодействующий с ним эксперт (в том числе ИИ-агент) имеет возможность задать желаемый формат ответа. Например, в виде одного из популярных языков текстовой разметки (Markdown, AsciiDoc).

Возможность задавать нужный формат ответа позволяет связать результаты работы генеративного ИИ с подходом Docs as Code. Так, например, в «Альфа-Банке» уже разработано решение (ИИ-агент), позволяющее по исходному коду генерировать документацию на сервис в формате AsciiDoc и размещать ее в корпоративной системе контроля версий¹⁰. Ведутся исследования и по другим направлениям с целью повысить производительность сотрудников в частности и эффективность деятельности организации в целом.

ЗАКЛЮЧЕНИЕ

Когда мы говорим об управлении содержанием проекта, важно держать в голове два его аспекта. С одной стороны, это множество работ по созданию продукта, сервиса или иного результата с заданными характеристиками. С другой – это множество самих характеристик продукта, сервиса или иного результата проекта.

Инструменты управления работами разнообразны и имеют глубокое проникновение в деятельность современных организаций. Что касается инструментов управления характе-

ристиками или, вернее сказать, требованиями к продукту, сервису или иному результату проекта, то, несмотря на достойный выбор, проникновение могло бы быть больше.

Практики DocOps также являются эффективной стратегией для обхода неадекватной системы контроля версий.

Практика показывает, что в большинстве случаев для управления требованиями используются локальные электронные документы в связке с почтовыми системами либо системы типа wiki. Данные инструменты имеют ряд недостатков, а их применение приводит к возникновению проблем, связанных, например, с противоречивостью и/или неконсистентностью требований.

Для решения данных проблем отечественные компании активно внедряют инструменты и практики разработчиков для управления требованиями и, как следствие, содержанием проекта. Данный тренд возник почти 10 лет назад, а с учетом современного тренда на внедрение генеративного ИИ получил новый импульс, благодаря которому открылись дополнительные направления исследований с целью повысить производительность и эффективность деятельности отечественных организаций.

Перечень упомянутых в статье языков разметки, систем, множеств практик и подходов приведен в приложении.

ПРИЛОЖЕНИЕ. ГЛОССАРИЙ

В таблице приведен глоссарий, содержащий перечень языков разметки, систем, множеств практик и подходов, упомянутых в статье, а также их определения.

Таблица. Глоссарий

Термин	Определение
AsciiDoc	Легковесный язык текстовой разметки, созданный в первую очередь для написания технической документации
Atlassian Bitbucket	Сервис для хостинга кода и совместной работы на основе Git от Atlassian
Atlassian Confluence	Сервис от Atlassian для совместной работы, представляющий собой единое рабочее пространство для команд, где они могут создавать и совместно редактировать документы, обмениваться знаниями, управлять проектами и поддерживать порядок в информации
Atlassian Jira	Сервис от Atlassian для управления рабочим процессом команд создателей ПО, которые хотят систематизировать и отслеживать свой труд
DocOps	Множество практик для интеграции команды разработки и команды сопровождения в части процесса разработки документации, в том числе его автоматизации (ревью, тестирование, сборка и публикация и пр.)
Docs as Code	Подход к написанию документации с использованием тех же инструментов, что и при написании кода, в том числе системы управления задачами, системы контроля версий, языков текстовой разметки и пр.
GitFlic	Платформа для работы с кодом от «Группы Астра»

Термин	Определение
Markdown	Легковесный язык текстовой разметки, созданный с целью добавления форматирования в простой текст с максимальным сохранением его читаемости человеком
Microsoft Outlook	Почтовый клиент для работы с электронной почтой, входящий в пакет программ Microsoft Office
Microsoft Project and Portfolio Management	Комплексное корпоративное решение от корпорации Microsoft для управления проектами, программами и портфелями проектов, позволяющее организациям централизованно отслеживать, анализировать и оптимизировать инвестиции в проекты в соответствии со своими стратегическими целями
Microsoft Word	Мощный текстовый процессор от корпорации Microsoft, предназначенный для создания, редактирования, просмотра и форматирования текстовых документов
АСУ-Инвест	Платформа разработки прикладных решений для автоматизации процессов инвестиционной, портфельной и проектной деятельности крупных государственных и коммерческих организаций от группы компаний «Проектная ПРАКТИКА»
ПМ «Форсайт»	Комплексное решение от группы компаний «Проектная ПРАКТИКА», позволяющее развивать проектное управление в организации системно. В ПМ «Форсайт» включена определенная методология проектного управления, поддерживаемая ИТ-решением, а для его внедрения предусмотрен комплекс организационных мер. Наличие в ПМ «Форсайт» такого набора элементов позволяет построить работающую, не отторгаемую компанией систему управления, учитывающую зрелость проектной деятельности и ее специфику
«Яндекс Вики»	Сервис от «Яндекса» для создания корпоративной базы знаний, которую наполняют и обновляют пользователи – сотрудники компании
«Яндекс Трекер»	Сервис для управления проектами и процессами от «Яндекса»

Окончание таблицы

СПИСОК ЛИТЕРАТУРЫ

Горелиц Н.К., Кильдишев Д.С., Хорошилов А.В. Управление требованиями к ответственным системам. Обзор решений. Труды ИСП РАН. 2019;1(31):25–48. [https://doi.org/10.15514/ISPRAS-2019-31\(1\)-2](https://doi.org/10.15514/ISPRAS-2019-31(1)-2)

Гузенко О.И., Балябин Д.С. ИИ-агенты в управлении проектами: возможности, ограничения и стратегии интеграции. Вестник проектного управления. 2025;2(1):71–78. <https://doi.org/10.26425/3034-6916-2025-1-2-71-78>

Лобзов А.В. Использование подхода docs as code для минимизации финансовых рисков. Управление финансовыми рисками. 2023;1(73):72–78. <https://doi.org/10.36627/2221-7541-2023-1-1-72-78>

Wiegiers, K., & Beatty, J. (2013). *Software requirements* (3rd ed.). Microsoft Press.

REFERENCES

Gorelits, N. K., Kildishev, D. S., & Khoroshilov, A. V. (2019). Requirements management for safety-critical systems. Overview of solutions. *Proceedings of ISP RAS*, 31(1), 25–48. (In Russian). [https://doi.org/10.15514/ISPRAS-2019-31\(1\)-2](https://doi.org/10.15514/ISPRAS-2019-31(1)-2)

Guzenko, O. I., & Baliabin, D. S. (2025). AI agents in project management: opportunities, limitations, and integration strategies. *Journal of Project Management*, 1(2), 71–78. (In Russian). <https://doi.org/10.26425/3034-6916-2025-1-2-71-78>

Lobzov, A. V. (2023). Application of docs as code approach for financial risk minimisation. *Financial risk management journal*, 1(73), 72–78. (In Russian). <https://doi.org/10.36627/2221-7541-2023-1-1-72-78>

Wiegiers, K., & Beatty, J. (2013). *Software requirements* (3rd ed.). Microsoft Press.

СПИСОК ИНЫХ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

¹ Маркина Т. Какую систему управления требованиями выбрать: обзор инструментов. Режим доступа: <https://habr.com/ru/companies/pt/articles/821643/> (дата обращения: 20.08.2025).

² Wiegiers K.E. Karl Wiegiers describes 10 requirements traps to avoid. Режим доступа: <http://users.csc.calpoly.edu/~csturner/courses/300f06/readings/reqtraps.pdf> (дата обращения: 20.08.2025).

³ Holscher E. Docs as Code. Режим доступа: <https://www.writethedocs.org/guide/docs-as-code/> (дата обращения: 20.08.2025).

⁴ Putrino J. DocOps. Режим доступа: <https://www.writethedocs.org/guide/doc-ops/> (дата обращения: 20.08.2025).

- ⁵ Грановская Ю., Маркелов И. Версионирование требований – применение аналитиками классических практик разработчиков. Режим доступа: <https://analystdays.ru/ru/talk/58553> (дата обращения: 20.08.2025).
- ⁶ Поташников Н.М. DocOps на Flow 2022. Режим доступа: <https://habr.com/ru/companies/jugru/articles/708338/> (дата обращения: 20.08.2025).
- ⁷ Валеев К., Новикова Л., Факторович С., Волынкин Н., Поташников Н., Беянина (Горбатенко) А. DocOps в работе системного аналитика. Режим доступа: <https://flowconf.ru/archive/2022/talks/e01ce8c8594242f58260d6a71b26de95/> (дата обращения: 20.08.2025).
- ⁸ Лобзов А. Использование инструментов и практик разработчиков в создании технической документации. Режим доступа: https://www.tadviser.ru/images/f/fe/15._Лобзов.pdf (дата обращения: 20.08.2025).
- ⁹ Куприянов Ю. Я попросил ChatGPT написать требования, и вот что получилось. Режим доступа: <https://habr.com/ru/news/704392/> (дата обращения: 20.08.2025).
- ¹⁰ Шелапутов И. Генерация документации на API-сервис рядом с кодом. Режим доступа: https://vk.com/video-215425037_456239552 (дата обращения: 20.08.2025).