

---

## ПРИМЕНЕНИЕ СТАТИСТИЧЕСКИХ МЕТОДОВ КОНТРОЛЯ КАЧЕСТВА В РАЗРАБОТКЕ БАНКОВСКОГО ПО

**Малышев Михаил Олегович,**

Студент магистратуры

2 курс, институт технологий управления

Кафедра управления инновациями

Московский технический университет связи и информатики

email: mix.malyshev2011@yandex.ru

**Чулюков Никита Владимирович,**

Студент магистратуры

2 курс, институт технологий управления

Кафедра управления инновациями

Московский технический университет связи и информатики

email: hipstakripp@gmail.com

**Басыров Ильдар Ильшатович,**

Студент магистратуры

2 курс, факультет «Сети и системы связи»

Кафедра «Сети связи и системы коммутации»

Московский технический университет связи и информатики

e-mail: basyrov.ildar@mail.ru

### Аннотация

---

В данной статье рассматривается возможность интеграции классических методов Statistical Quality Control (и-карт Шухарта, EWMA-наблюдения и дефектных метрик) в высокочастотный DevOps-процесс крупного российского банка. На основе нормативных требований Банка России № 683-П и ISO/IEC 25010 анализируются существующие Dev + QA-процессы, выделяются точки встраивания SQC-инструментов и проводится SWOT-анализ факторов, влияющих на успех внедрения. Показано, что централизованный CI/CD-конвейер и автоматический сбор метрик создают прочную основу для непрерывного статистического контроля, а критические риски связаны с регулярным пересчетом лимитов и уровнем статистической компетенции команд.

---

**Ключевые слова:** Statistical Quality Control, DevOps, банковское программное обеспечение, CI/CD, и-карты Шухарта.

---

## APPLICATION OF STATISTICAL QUALITY CONTROL METHODS IN BANKING SOFTWARE DEVELOPMENT

**Malyshev Mikhail Olegovich,**

Master's student

2nd year, Institute of management technologies

Department of innovation management  
Moscow Technical University of Communications and Informatics

**Chulyukov Nikita Vladimirovich,**

Master's student  
2nd year, Institute of management technologies  
Department of innovation management  
Moscow Technical University of Communications and Informatics

**Basyrov Ildar Ilshatovich,**

Master's student  
2nd year, Faculty of Networks and Communication Systems  
Department of Communication Networks and Switching Systems  
Moscow Technical University of Communications and Informatics

---

**ABSTRACT**

---

This article explores the integration of classical Statistical Quality Control (SQC) techniques – Shewhart u-control charts, EWMA monitoring, and defect-based metrics – into the high-frequency DevOps pipeline of a large Russian bank. Drawing on the regulatory framework of Bank of Russia Directive No. 683-P and the ISO/IEC 25010 quality model, the paper analyses existing Dev + QA processes, identifies insertion points for SQC tools, and presents a SWOT assessment of the factors that determine implementation success. The study demonstrates that a centralized CI/CD pipeline and automated metric collection form a solid basis for continuous statistical control, while the main risks stem from the need for regular recalculation of control limits and the statistical proficiency of engineering teams.

---

**Keywords:** Statistical Quality Control; DevOps; banking software; CI/CD; Shewhart u-charts

---

**Введение**

Надежность цифровых сервисов банка сегодня жестко регулируется Положением № 683-П Банка России, которое требует от кредитных организаций показывать устойчивость и безотказность информационных систем в измеримых показателях, включая процедуры контроля изменений программного обеспечения [1]. Эти нормативные рамки дополняются международной моделью ISO/IEC 25010, которая описывает качество ПО через систему взаимосвязанных характеристик – от функциональной пригодности до безопасности и надежности [2].

АО «Альфа-Банк» обслуживает клиентов в режиме нескольких тысяч релизов в год; внутренние оценки называют порядок 7 000 выпусков, проходящих через CI/CD-конвейер каждые двенадцать месяцев [3]. Такое темпо-развитие сокращает время вывода продукта на рынок, но одновременно усиливает риск накопления дефектов: даже небольшой сбой мгновенно становится событием федерального масштаба.

Методы статистического контроля качества (Statistical Quality Control, SQC), унаследованные от промышленного производства, позволяют фиксировать отклонения процесса до появления критических ошибок в продакшне. Цель данной статьи – обобщить нормативные, методологические и процессные аспекты применения Statistical Quality

Control в высокочастотном DevOps-контуре крупного банка; сопоставить требования Банка России № 683-П и модели ISO/IEC 25010 с реальной инженерной практикой CI/CD; а также с помощью SWOT-анализа выявить ключевые внутренние и внешние факторы, от которых зависит успешная интеграция u-карт, EWMA-мониторинга и других SQC-инструментов в процесс управления качеством банковского программного обеспечения.

Теоретические основы статистического контроля качества в разработке ПО

SQC опирается на идею, что любые свойства программного продукта – от плотности дефектов до времени восстановления после сбоя – подчиняются стохастическим законам и потому поддаются мониторингу средствами математической статистики. Базой для большинства банковских регламентов остается метрика Defect Density (DD) – число подтвержденных ошибок на тысячу строк кода. Ее применяют не только в классических исследовательских работах IBM и NASA, но и в современных руководствах по оценке качества ПО: у Кан и Фугэта она фигурирует как основной индикатор зрелости процессов разработки [4]. Однако сама по себе дефектная плотность не защищает от «утечек» багов в продакшн, поэтому для DevOps-контура банков часто рассчитывают Escape Rate – долю ошибок, обнаруженных после выхода релиза в эксплуатацию. Корпоративные репозитории данных показывают, что скачок показателя выше 15 % почти всегда сопровождается жалобами клиентов и расследованием службы комплаенса [5].

Чтобы отделить устойчивую тенденцию от случайного всплеска, применяют контрольные карты Шухарта. Для потоков, где каждая сборка представляет самостоятельную выборку, рекомендуется Individuals-chart: вариативность оценивается по скользящему диапазону между соседними релизами. Подробное изложение метода и практические примеры размещены в монографии Дугласа Монтгомери, давно ставшей настольной для инженеров-статистиков [5]. В случаях, когда объем изменений заметно колеблется, аналитики переходят к u-chart, нормируя число дефектов на объем кода; при пуассоновском распределении ошибок этот подход дает корректные границы управления даже для коротких спринтов.

Быстро меняющиеся конвейеры CI/CD предъявляют еще более строгие требования к чувствительности: сдвиг процесса может уложиться в одно-два обновления. Поэтому многие авторы предлагают экспоненциально-взвешенные скользящие средние (Exponentially Weighted Moving Average, EWMA). Недавние исследования (например, работа Чоудхури и соавт. по адаптивным EWMA-картам 2024 г.) показывают, что метод выявляет смещение менее чем в 0,5  $\sigma$  уже на шестой-седьмой точке наблюдения, что важно для банковских приложений с суточными релизами [6].

Сигналы контрольных карт становятся операционными решениями после прохождения цикла DMAIC (Define-Measure-Analyze-Improve-Control – определить, измерить, анализировать, улучшить, контролировать), базового для Six Sigma. На этапе Define границы процесса фиксируются от commit-хэша до публикации контейнера; Measure превращает Defect Density, Escape Rate и Mean Time to Recovery в материю, пригодную для статистики; Analyze строит u- или EWMA-карты и тестирует гипотезы о дрейфе; Improve внедряет изменения – скажем, перенос статического анализа на раннюю стадию pipeline; Control удерживает достигнутый результат через автоматический пересчет контрольных лимитов. Эффективность такой интеграции Six Sigma с SDLC детально обсуждают Капур и Басу в обзоре методик Lean Six Sigma для software engineering [7], а в российской литературе примеры привел С. Г. Кириллов, анализируя применение DMAIC к портфелю микросервисов банка среднего размера [8].

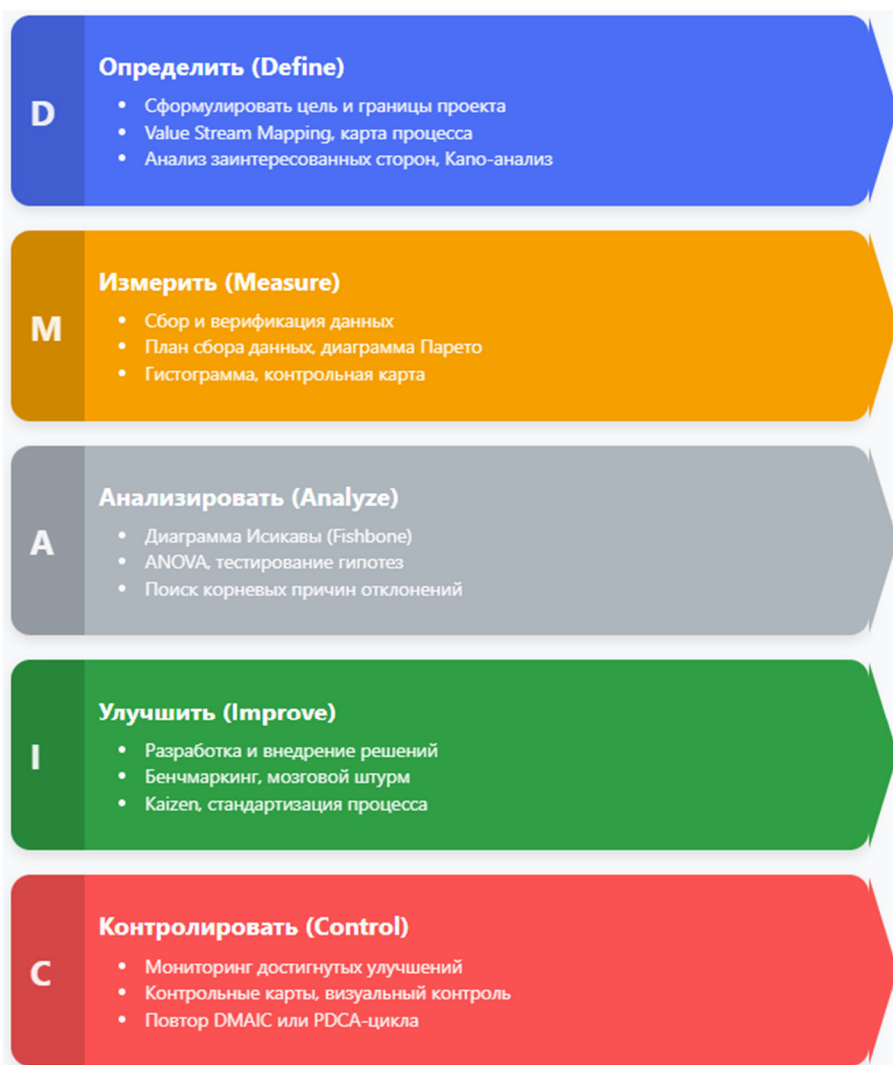


Рисунок 1. Цикл DMAIC и ключевые инструменты фаз [9]

### Особенности Dev + QA-процессов в банковском секторе

В финансовой отрасли требования к непрерывности цифровых сервисов задают не только рыночную конкуренцию, но и надзор: Базельский комитет рассматривает «операционную устойчивость» как способность банка поддерживать критические операции даже при серьезных сбоях, а европейские регуляторы прямо связывают ее с измеримыми целевыми значениями доступности и времени восстановления [10]. Эти нормативы трансформируются во внутренние SLA, где для клиентских каналов недоступность дольше нескольких минут квалифицируется как инцидент высшего приоритета. Высокая цена простоя задает общий тон всей инженерной организации – от проектирования до эксплуатации.

Исторически основная книга транзакций Альфа-Банка обслуживается на мейнфрейме, а пользовательские и партнерские функции уже перешли к облачным микросервисам. Систематическое картирование литературных источников по модернизации легаси-систем показывает, что смешанный («двухконтурный») вариант позволяет сохранить надежность ядра и, одновременно, увеличить пропускную способность фронтов почти на половину за счет независимых развертываний [11]. Однако такая гибридная архитектура резко усложняет трассировку изменений: каждое бизнес-правило проводит два независимых цикла сборки – для мейнфрейм-кода и для микросервиса – что повышает риск рассинхронизации дефектов.

Эти сложности компенсированы комплексной DevOps-платформой. Недавний аналитический обзор взаимодействия DevOps и корпоративной архитектуры в банках

фиксирует, что переход к тысячам выпусков в год возможен только при централизованном CI/CD-канале, где контроль качества и безопасности встроен в слой архитектурных принципов TOGAF [12]. Именно платформа обеспечивает единый оркестратор сборок, управление зависимостями и автоматический откат при нештатном поведении.

Повышенная частота поставок делает традиционную модель «тестирование после разработки» экономически невыгодной, что подтверждено эмпирическим исследованием эффекта shift-left: перенос проверки требований и написания автоматических тестов на ранние фазы снижает совокупные расходы на исправления почти на четверть [13]. В Альфа-Банке подобная практика закреплена регламентом Definition of Ready: ни одна задача не принимается в работу без заранее описанных критериев приемки и набора автотестов, которые сразу же включаются в конвейер.

Столь же формальным стал и блок нормативной безопасности. Концепция compliance-as-code, описанная в работе Томпсона и Мейера, переводит требования регуляторов и внутренних политик в исполняемые правила, автоматически проверяющиеся на стадии сборки. Авторитетные расчеты показывают, что автоматизированная валидация снижает затраты на ежегодные аудиторские проверки почти на треть и резко сокращает число ручных исключений [14].

Наконец, высокая рыночная волатильность заставляет банки вкладываться в наблюдаемость. Специальное исследование по интеграции observability-практик в финансовых ИТ-сервисах выявило сокращение среднего времени обнаружения инцидента практически вдвое, когда телеметрия приложений, журналирование и трассировки транзакций объединены с DevOps-конвейером в единую систему событий [15]. Для операционных команд это означает ранний статистический сигнал о дрейфе показателей, в то время как для регулятора – документальное подтверждение управляемости процессов.

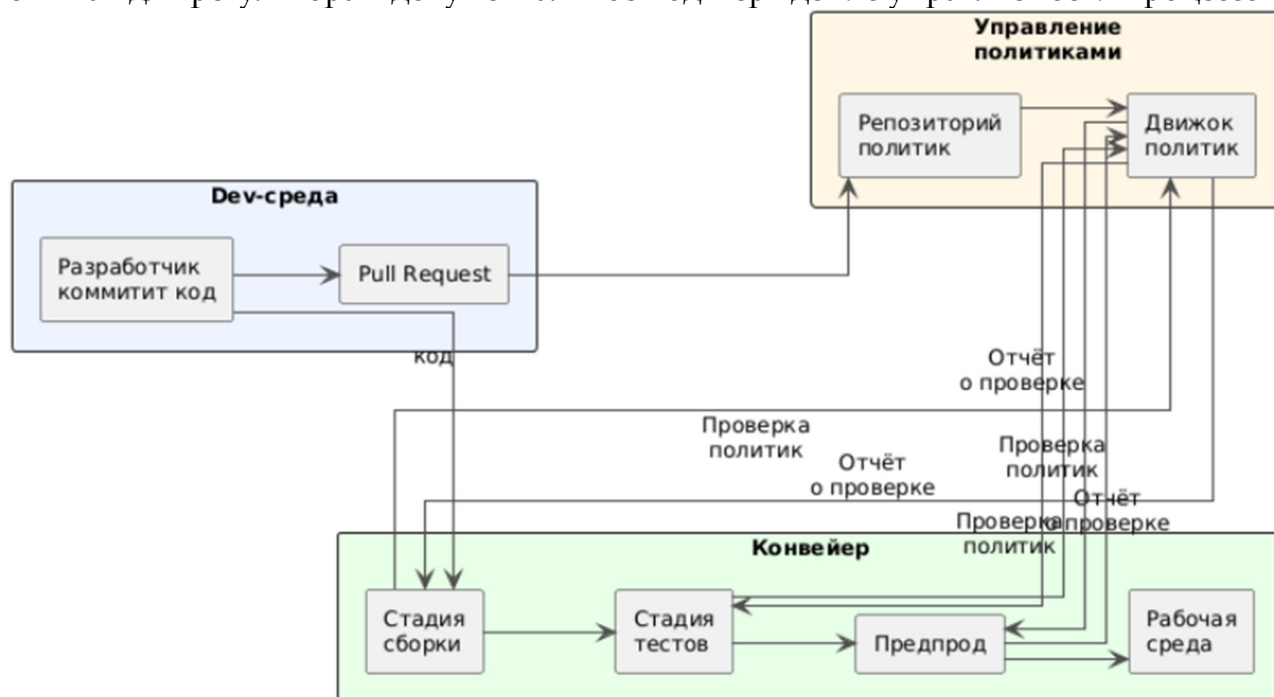


Рисунок 2. Архитектура DevOps-конвейера с автоматизированным шлюзом комплаенса [16]

#### SWOT-анализ внедрения SQC-подхода в банковском DevOps

В целях комплексной оценки инициативы по внедрению Statistical Quality Control (SQC) в DevOps-контур банка был выполнен SWOT-анализ, позволяющий сопоставить внутренние и внешние факторы, влияющие на эффективность проекта.

	Сильные стороны (Strengths)	Слабые стороны (Weaknesses)
Внутренняя среда	<ul style="list-style-type: none"> <li>• Метрики <i>Defect Density</i> и <i>Escape Rate</i> уже собираются автоматически в CI/CD.</li> <li>• Централизованный пайплайн обеспечивает единые точки контроля.</li> <li>• Поддержка руководства — регламент DoR закреплён приказом СТО.</li> </ul>	<ul style="list-style-type: none"> <li>• Недостаточная статистическая подготовка части команд.</li> <li>• Данные из Jira, SonarQube и Prometheus ещё не полностью синхронизированы.</li> <li>• Контрольные лимиты обновляются нерегулярно.</li> </ul>
Внешняя среда	<ul style="list-style-type: none"> <li>• Готовая Auto-ML-платформа позволяет прогнозировать дефекты.</li> <li>• Результаты SQC можно использовать как доказательство управляемости для Банка России.</li> <li>• Возможность связать SQC-метрики с бизнес-метриками (конверсия, NPS).</li> </ul>	<ul style="list-style-type: none"> <li>• По-жесткому регулятору могут понадобиться новые метрики.</li> <li>• Рост транзакций увеличит нагрузку на систему метрик и storage.</li> <li>• Кадровая текучесть рискует снизить компетенции по SQC.</li> </ul>

Рисунок 3. SWOT-анализ внедрения SQC-подхода в банковском DevOps

SWOT-карта на рисунке 3 показывает, что техническая база (автоматический сбор метрик и централизованный CI/CD) и организационная поддержка уже создают прочную «платформу» для полноценного статистического контроля качества. Ключевые риски находятся не в технологиях, а в процессах: регулярность пересчета контрольных лимитов и рост компетенций команд. При минимизации этих слабых сторон банк сможет превратить SQC не просто в инструмент обнаружения дефектов, а в аргумент для регуляторов и конкурентное преимущество на рынке.

### Заключение

Представленный обзор показал, что классические методы Statistical Quality Control – и-карты Шухарта, EWMA-наблюдение и связанные с ними дефектные метрики – органично вписываются в архитектуру высокочастотного DevOps-конвейера крупного банка. Их внедрение позволяет превратить требования Банка России № 683-П и модель ISO/IEC 25010 из формального «чек-листа» в живую часть CI/CD-процесса: контрольные границы автоматически пересчитываются, а сигналы о дрейфе качества поступают операционным командам еще до выхода релиза. SWOT-анализ подтвердил, что технологическая база (централизованный пайплайн и автоматический сбор метрик) уже создает устойчивый фундамент для SQC, а организационная поддержка на уровне СТО снижает барьер изменений.

При этом ключевые риски лежат не в инструментах, а в процессах и компетенциях: регулярность обновления лимитов, полнота консолидированных данных и статистическая грамотность инженеров. Именно их минимизация позволит превратить SQC-подход из точечного контроля дефектов в стратегический актив банка – основание для доказательной комплаенс-отчетности и дополнительное конкурентное преимущество на рынке цифровых финансовых сервисов. Дальнейшее развитие видится в расширении набора мониторинговых индикаторов до бизнес-метрик, интеграции прогнозных ML-моделей и формировании единого дашборда, отражающего взаимосвязь качества кода, клиентского опыта и нормативных показателей.

### Список литературы:

1. Положение 683-П // [regulhub.kaspersky.ru](https://regulhub.kaspersky.ru) URL: [regulhub.kaspersky.ru/decrees/polozhenie-683-p](https://regulhub.kaspersky.ru/decrees/polozhenie-683-p) (дата обращения: 28.04.2025).
2. ISO/IEC 25010 // [iso25000.com](https://iso25000.com) URL: [iso25000.com/en/iso-25000-standards/iso-25010](https://iso25000.com/en/iso-25000-standards/iso-25010) (дата обращения: 28.04.2025).

3. Как оседлать хаос // Хабр URL: [habr.com/ru/companies/alfa/articles/764518/](https://habr.com/ru/companies/alfa/articles/764518/) (дата обращения: 28.04.2025).
4. Software Quality Metrics Overview // [www.pearsonhighered.com URL: www.pearsonhighered.com/assets/samplechapter/0/2/0/1/0201729156.pdf](https://www.pearsonhighered.com/URL:www.pearsonhighered.com/assets/samplechapter/0/2/0/1/0201729156.pdf) (дата обращения: 28.04.2025).
5. Montgomery D. C. Introduction to Statistical Quality Control. 6th ed. Hoboken: Wiley, 2009. 754 p.
6. Alduais, Fuad & Khan, Zahid. (2023). EWMA Control Chart for Rayleigh Process With Engineering Applications. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3240660.
7. Implementing Lean Six Sigma In Software Development // [www.sixsigmaonline.org URL: www.sixsigmaonline.org/six-sigma-for-software-development/](https://www.sixsigmaonline.org/URL:www.sixsigmaonline.org/six-sigma-for-software-development/) (дата обращения: 28.04.2025).
8. Анализ современного статистического программного обеспечения / Т. А. Першина, О. Э. Башина, Л. А. Давлетшина, Е. А. Долгих // Вестник Академии. – 2020. – № 4. – С. 7-16. – DOI 10.51409/2073-9621-2020-4-7-16. – EDN QDCGUY.
9. Define, Measure, Analyze, Improve, Control (DMAIC) Methodology as a Roadmap in Quality Improvement // [pmc.ncbi.nlm.nih.gov URL: pmc.ncbi.nlm.nih.gov/articles/PMC10229001/](https://pmc.ncbi.nlm.nih.gov/URL:pmc.ncbi.nlm.nih.gov/articles/PMC10229001/) (дата обращения: 28.04.2025).
10. European Banking Authority. Risk Assessment Report – July 2024. Chapter “Operational risks and resilience”. Luxembourg: EBA, 2024. 78 p.
11. Almeida N. R., Campos G. N., Moraes F. R., Affonso F. J. Modernization of Legacy Systems to Microservice Architecture: A Tertiary Study // Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024). Vol. 2. Angers (France): SCITEPRESS – Science and Technology Publications, 2024. P. 581-592.
12. Cui J., Herryanto N. A., Luong L., Pham N. N., Nguyen M. H., Dilnutt R. How DevOps Impacts Enterprise Architecture in the Banking and Financial Services Industry // Enterprise Architecture Professional Journal (EAPJ). – Melbourne: The University of Melbourne, 2024. – 21 p.
13. Vaddadi, Srinivas & Thatikonda, Ramya & Padthe, Adithya & Arnepalli, Pandu Rangarao. (2023). Shift-Left Testing Paradigm Process Implementation for Quality of Software Based on Fuzzy. 10.21203/rs.3.rs-2845536/v1.
14. Metha S. Automated Code Review for Secure Banking Applications: Harnessing AI for Security, Performance, and Regulatory Compliance in Financial Software Engineering // Journal of Information Systems Engineering and Management. – 2025. – Vol. 10, No. 31s. – P. 465–476. – DOI 10.52783/jisem.v10i31s.5100.
15. Mahida, A. Integrating Observability with DevOps Practices in Financial Services Technologies: A Study on Enhancing Software Development and Operational Resilience / A. Mahida // International Journal of Advanced Computer Science and Applications. – 2024. – Vol. 15, No. 7. – DOI 10.14569/ijacsa.2024.0150701. – EDN UCAUCI.
16. Ramreddy Gouni (2025). Automating Compliance In Devops Pipelines. International Journal of Computational and Experimental Science and Engineering, 11(2).

**References:**

1. Regulation 683-P // regulhub.kaspersky.ru URL: regulhub.kaspersky.ru/decrees/polozhenie-683-p (date of access: 28.04.2025).
2. ISO/IEC 25010 // iso25000.com URL: iso25000.com/en/iso-25000-standards/iso-25010 (date of access: 28.04.2025).
3. How to saddle chaos // Habr URL: habr.com/ru/companies/alfa/articles/764518/ (date of access: 28.04.2025).
4. Software Quality Metrics Overview // www.pearsonhighered.com URL: www.pearsonhighered.com/assets/samplechapter/0/2/0/1/0201729156.pdf (access date: 04/28/2025).
5. Montgomery D. C. Introduction to Statistical Quality Control. 6th ed. Hoboken: Wiley, 2009. 754 p.
6. Alduais, Fuad & Khan, Zahid. (2023). EWMA Control Chart for Rayleigh Process With Engineering Applications. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3240660.
7. Implementing Lean Six Sigma In Software Development // www.sixsigmaonline.org URL: www.sixsigmaonline.org/six-sigma-for-software-development/ (date of access: 28.04.2025).
8. Analysis of modern statistical software / T. A. Pershina, O. E. Bashina, L. A. Davletshina, E. A. Dolgikh // Bulletin of the Academy. - 2020. - No. 4. - P. 7-16. - DOI 10.51409/2073-9621-2020-4-7-16. - EDN QDCGUY.
9. Define, Measure, Analyze, Improve, Control (DMAIC) Methodology as a Roadmap in Quality Improvement // pmc.ncbi.nlm.nih.gov URL: pmc.ncbi.nlm.nih.gov/articles/PMC10229001/ (access date: 04/28/2025).
10. European Banking Authority. Risk Assessment Report – July 2024. Chapter “Operational risks and resilience”. Luxembourg: EBA, 2024. 78 p.
11. Almeida N. R., Campos G. N., Moraes F. R., Affonso F. J. Modernization of Legacy Systems to Microservice Architecture: A Tertiary Study // Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024). Vol. 2. Angers (France): SCITEPRESS – Science and Technology Publications, 2024. P. 581-592.
12. Cui J., Herryanto N. A., Luong L., Pham N. N., Nguyen M. H., Dilnutt R. How DevOps Impacts Enterprise Architecture in the Banking and Financial Services Industry // Enterprise Architecture Professional Journal (EAPJ). – Melbourne: The University of Melbourne, 2024. – 21 p.
13. Vaddadi, Srinivas & Thatikonda, Ramya & Padthe, Adithya & Arnepalli, Pandu Rangarao. (2023). Shift-Left Testing Paradigm Process Implementation for Quality of Software Based on Fuzzy. 10.21203/rs.3.rs-2845536/v1.
14. Metha S. Automated Code Review for Secure Banking Applications: Harnessing AI for Security, Performance, and Regulatory Compliance in Financial Software Engineering // Journal of Information Systems Engineering and Management. – 2025. – Vol. 10, No. 31s. – P. 465–476. – DOI 10.52783/jisem.v10i31s.5100.
15. Mahida, A. Integrating Observability with DevOps Practices in Financial Services Technologies: A Study on Enhancing Software Development and Operational Resilience /

A. Mahida // International Journal of Advanced Computer Science and Applications. – 2024. – Vol. 15, No. 7. – DOI 10.14569/ijacsa.2024.0150701. – EDN UCAUCI.

16. Ramreddy Gouni (2025). Automating Compliance In Devops Pipelines. International Journal of Computational and Experimental Science and Engineering, 11(2)