

Применение DevOps в корпоративных информационных системах: организационные условия, технологические решения и этические ограничения

М.А. Хлыстун, В.Ю. Чертовикова, И.В. Перелехов, Е.В. Лим

Аннотация — В статье рассматриваются организационные, технологические и этические аспекты применения DevOps в корпоративных информационных системах. Цель исследования – разработка модели выбора и оценки DevOps-практик, связывающей ограничения релизного процесса с соответствующими управленческими, технологическими и контрольными мерами. Используются сравнительный, системный и структурно-логический анализ, а также сценарное расчетное моделирование традиционного и DevOps-подходов. Предложена матрица, сопоставляющая выявленное ограничение, применяемое решение, условие контроля и измеримый результат. В рамках заданных параметров применение DevOps характеризовалось сокращением релизного цикла и времени восстановления, ростом частоты развертываний, уровня автоматизации, контрольного покрытия и прослеживаемости. При неизменном числе неуспешных развертываний их относительная доля снижалась за счет увеличения общего числа выпусков. Результативность DevOps зависит от согласования организационных изменений, технологической автоматизации, этических требований и контрольных процедур. Результаты исследования имеют расчетный характер и требуют проверки на эксплуатационных данных.

Ключевые слова — CI/CD, DevOps, DevSecOps, автоматизация релизного цикла, корпоративные информационные системы, прослеживаемость изменений.

I. ВВЕДЕНИЕ

Распространение облачных и микросервисных архитектур и рост числа взаимосвязанных цифровых сервисов повышают требования к скорости, надежности и безопасности разработки корпоративного программного обеспечения (ПО). DevOps обеспечивает интеграцию процессов разработки и эксплуатации, однако его результативность определяется зрелостью CI/CD, контейнеризации и автоматизации

инфраструктуры, а также распределением ответственности и встроенными механизмами контроля.

Актуальность темы подтверждается научными исследованиями. Автоматизация релизного цикла с использованием CI/CD, контейнеризации и Infrastructure as Code (IaC) сокращает время выпуска ПО, но устойчивость полученного эффекта зависит от компетенций персонала и качества межфункционального взаимодействия [1], [2]. В исследованиях DevOps и DevSecOps подчеркивается необходимость встраивания защиты данных, распределения ответственности и контрольных процедур непосредственно в процессы разработки, поставки и эксплуатации ПО [3], [4]. Вместе с тем организационные, технологические и этические аспекты DevOps преимущественно рассматриваются отдельно, что обуславливает необходимость их комплексного анализа.

Научная новизна исследования заключается в разработке модели выбора и оценки DevOps-практик, связывающей ограничения корпоративной информационной системы с организационными и технологическими решениями, проверяемыми условиями этического контроля и измеримыми показателями релизного процесса. Цель исследования – разработать подход к выбору и оценке DevOps-практик, обеспечивающий согласование организационных изменений, архитектурных решений и этических ограничений с показателями скорости, надежности и безопасности выпуска ПО. Гипотеза заключается в том, что целевая DevOps-конфигурация, сочетающая кросс-функциональное взаимодействие, стандартизированную автоматизацию и встроенные контрольные механизмы, характеризуется более благоприятным соотношением скорости, устойчивости и контролируемости релизного процесса по сравнению с традиционной моделью.

II. МАТЕРИАЛЫ И МЕТОДЫ

В исследовании применялись методы сравнительного, системного и структурно-логического анализа. Они использовались для сопоставления традиционной модели раздельной работы подразделений разработки и эксплуатации с DevOps-подходом, а также для установления взаимосвязей между организационными, технологическими и этическими условиями его

Статья получена 29 июля 2026.

М.А. Хлыстун, HPC Systems Administrator, Fred DeMatteis School of Engineering and Applied Science, Hofstra University, Хемпстед, США (e-mail: mikhail.khlystun.publications@gmail.com).

В.Ю. Чертовикова, аспирант, кафедра психологии труда и инженерной психологии, факультет психологии МГУ им. М.В. Ломоносова, Москва, Россия (e-mail: v.chertovikova@mail.ru).

И.В. Перелехов, старший веб-разработчик, ООО «Оджетто Веб», Таганрог, Россия (e-mail: ivanperelekhov@gmail.com).

Е.В. Лим, независимый исследователь, Москва, Россия (e-mail: e.lim@ro.ru).

применения. Анализ охватывал кросс-функциональное взаимодействие, распределение ответственности, CI/CD, IaC, контейнеризацию, DevSecOps, наблюдаемость, защиту данных, прослеживаемость изменений и контроль критических операций.

Практическая часть основана на сценарном расчетном моделировании релизного цикла условной корпоративной информационной системы. Сопоставлялись традиционная организация разработки и целевая DevOps-модель при одинаковых периоде наблюдения, количестве запланированных изменений и архитектуре цифрового сервиса. Оценка проводилась по продолжительности релизного цикла, частоте развертываний, доле неуспешных изменений, среднему времени восстановления после неуспешного развертывания, уровню автоматизации, покрытию обязательными проверками и полноте прослеживаемости изменений.

III. РЕЗУЛЬТАТЫ

Логика представления результатов строится на последовательном анализе организационных условий применения DevOps, технологического контура DevSecOps и связанных с ним этических ограничений. На этой основе формируется практическая модель выбора и оценки DevOps-практик, внутренняя согласованность которой и потенциальные различия между сценариями оцениваются посредством расчетного моделирования.

А. Организационные условия применения DevOps в корпоративной среде

В корпоративной среде DevOps охватывает деятельность ИТ-команд, управление архитектурой, информационную безопасность и взаимодействие с бизнес-подразделениями [5], [6]. Масштаб распространения данного подхода отражается в отраслевой динамике: по оценке Fortune Business Insights, объем мирового рынка решений и услуг DevOps в 2026 г. оценивается в 24,3 млрд долл. США (рис. 1).

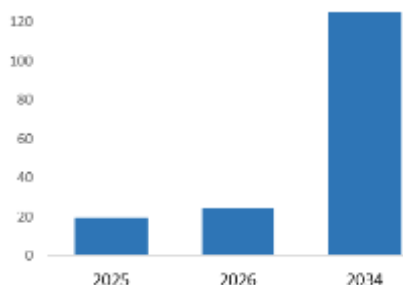


Рис. 1. Оценка и прогноз объема мирового рынка решений и услуг DevOps в 2025-2034 гг., млрд долл. США, по данным Fortune Business Insights

Рост рынка сопровождается увеличением инвестиций в автоматизацию разработки, однако внедрение инструментов не устраняет сопротивление изменениям, недостаточную поддержку руководства, дефицит компетенций и длительное последовательное согласование релизов [7], [8]. Поэтому результативность

DevOps определяется технологической зрелостью и распределением ответственности между разработкой, эксплуатацией и владельцами цифрового продукта.

Масштабирование кросс-функциональных команд требует сочетания операционной автономности с едиными инфраструктурными и контрольными стандартами. По данным отчета Perforce Software State of Platform Engineering 2026, основанного на опросе 820 технологических специалистов, высокую зрелость автоматизации управленческих и контрольных процедур отметили 79% организаций с развитой платформенной инженерией и 14% организаций с низкой зрелостью платформенного подхода. Распространенность платформенного подхода подтверждается исследованием Google Cloud DORA 2025 г., охватившим почти 5 тыс. технологических специалистов: внутренние платформы используют 90% организаций, 76% организаций имеют специализированные платформенные команды, а 29% организаций эксплуатируют несколько платформ. Это указывает на переход от разрозненного набора инструментов к стандартизированным сервисам самообслуживания. Основные организационные условия DevOps и показатели их оценки представлены в таблице 1.

Таблица 1. Организационные условия результативного применения DevOps

Организационное условие	Практическая реализация	Контролируемый показатель
Сквозная ответственность за продукт	Закрепление владельцев продукта и сервисов, участие команды в эксплуатации.	Количество межфункциональных передач, время согласования, доля инцидентов, устраненных внутри команды.
Стандартизация инструментального контура	Единые шаблоны CI/CD, мониторинга и управления артефактами.	Число используемых инструментов, частота переключений между ними, полнота прослеживаемости изменений.
Организация контрольных функций	Встраивание проверок безопасности и аудита в релизный процесс.	Доля проверенных изменений, число обходов контрольных процедур.

Организационную результативность DevOps целесообразно оценивать по времени согласования, количеству межфункциональных передач, доступности платформенных сервисов и способности команды самостоятельно сопровождать цифровой продукт.

В. Технологические и этические аспекты DevOps и DevSecOps

Технологический контур DevOps объединяет автоматизацию разработки, поставки и эксплуатации ПО [9]. Развитие данного подхода в направлении DevSecOps предполагает встраивание требований безопасности во все стадии CI/CD-конвейера, включая контроль исходного кода, зависимостей, инфраструктурных конфигураций, контейнерных образов, секретов, программных интерфейсов (API) и итоговых артефактов. Основные элементы

технологического контура представлены в таблице 2.

Таблица 2. Технологический контур DevSecOps в корпоративной информационной системе

Элемент контура	Объект контроля	Основные механизмы	Результат
Управление исходным кодом	Код, ветви разработки, история изменений.	Контроль версий, code review, проверка правил слияния.	Прослеживаемость изменений.
Непрерывная интеграция	Сборка и тестирование программных компонентов.	Автоматическая сборка, модульные и интеграционные тесты.	Раннее выявление дефектов.
Анализ безопасности	Код, API и конфигурации.	SAST, DAST, проверка секретов и политик.	Блокирование небезопасных изменений.
Контроль зависимостей	Сторонние библиотеки и пакеты.	Software Composition Analysis, проверка лицензий, SBOM.	Выявление уязвимых и неподтвержденных компонентов.
Управление инфраструктурой	Конфигурации IaC и контейнерные образы.	Проверка шаблонов, сканирование образов, policy as code.	Воспроизводимость и соответствие инфраструктуры требованиям.
Непрерывное развертывание	Изменения в производственной среде.	Canary release, blue-green deployment, feature flags, автоматический откат.	Ограничение последствий неуспешного релиза.
Наблюдаемость	Состояние сервисов и пользовательских операций.	Метрики, журналы, трассировка и оповещения.	Выявление отклонений и сокращение времени восстановления.

Необходимость сквозного контроля подтверждается данными Verizon DBIR 2026, основанными на анализе инцидентов, произошедших с 1 ноября 2024 г. по 31 октября 2025 г. Эксплуатация уязвимостей стала начальным вектором доступа в 31% утечек данных. Только 26% уязвимостей, включенных в каталог CISA Known Exploited Vulnerabilities, были полностью устранены, при этом медианное время их исправления увеличилось с 32 до 43 дней. Дополнительные риски связаны с цепочкой поставки ПО: в 2025 г. Sonatype выявила более 454,6 тыс. новых вредоносных пакетов, а их совокупное количество превысило 1,233 млн. Эти

данные обосновывают применение SCA, SBOM, доверенных реестров и проверки происхождения артефактов непосредственно в CI/CD-конвейере.

Техническая управляемость DevOps-процессов не исключает организационных и этических последствий автоматизации [10]. Высокая частота релизов и перераспределение функций между участниками процесса повышают требования к ответственности, прозрачности операций, конфиденциальности данных и защите интересов пользователей. Основные этические аспекты применения DevOps представлены в таблице 3.

Таблица 3. Этические аспекты применения DevOps в корпоративной среде [11], [12]

Этический аспект	Основной риск	Механизм обеспечения
Ответственность за изменения	Размывание ответственности между разработкой, эксплуатацией и автоматизированными средствами	Назначение владельца сервиса, фиксация автора и согласующего лица
Приоритет безопасности и надежности	Ускорение релизов в ущерб качеству и устойчивости системы.	Обязательные проверки, критерии допуска, запрет обхода контрольных этапов.
Прозрачность операций	Невозможность восстановить последовательность принятия и выполнения решений.	Журналирование, трассировка изменений, хранение истории развертываний.
Конфиденциальность данных	Раскрытие персональных, коммерческих или служебных данных.	Разграничение доступа, маскирование данных, управление секретами.
Человеческий контроль	Автоматическое выполнение критичных операций без проверки последствий.	Ручное подтверждение высокорисковых изменений и возможность отката.
Учет интересов пользователей	Выпуск изменений, ухудшающих доступность или корректность цифрового сервиса.	Тестирование критических пользовательских маршрутов, мониторинг SLO.
Соразмерность мониторинга персонала	Избыточный сбор данных о действиях разработчиков и оценка работников по отдельным метрикам.	Минимизация данных, ограничение доступа, комплексная оценка результатов.

Этические требования должны преобразовываться в проверяемые условия DevOps-процесса: закрепление ответственности, прослеживаемость изменений, разграничение доступа, обязательное выполнение контрольных процедур и подтверждение высокорисковых операций. Это позволяет соотносить скорость поставки ПО с требованиями надежности, безопасности и защиты интересов пользователей.

С. Практическая модель применения DevOps в корпоративной среде

Предлагаемая модель предназначена для выбора и оценки DevOps-практик применительно к ограничениям корпоративной информационной системы. Объектом анализа выступает сквозной процесс изменения программного продукта – от формирования бизнес-

требования до эксплуатации выпущенной версии. Модель сопоставляет выявленное ограничение, применяемое решение, обязательное условие контроля и измеримый результат, связывая организационные, технологические и этические аспекты с характеристиками релизного процесса. Логика ее применения представлена на рисунке 2.



Рис. 2. Практическая модель выбора и оценки DevOps-практик в корпоративной среде

Представленная последовательность используется как аналитическая рамка: для каждого ограничения определяются соответствующая DevOps-практика, условие ее безопасного применения и показатель результативности. Это позволяет оценивать влияние организационных и технологических решений на управляемость, надежность и безопасность поставки ПО. Основные взаимосвязи представлены в таблице 4.

Таблица 4. Матрица выбора и оценки DevOps-практик

Выявленное ограничение	Применяемое решение	Контрольное условие	Оцениваемый результат
Длительное межфункциональное согласование	Закрепление владельца сервиса, расширение автономности команды.	Сохранение полномочий для разделения критических операций.	Время согласования, количество передач задачи.
Низкая воспроизводимость развертываний	CI/CD и IaC	Проверка конфигураций и журналирование изменений.	Продолжительность релизного цикла, доля ручных операций.
Высокая доля неуспешных релизов	Автоматизированное тестирование и поэтапное развертывание.	Блокирование релиза при критических дефектах.	Доля неуспешных изменений, время восстановления.
Недостаточная прослеживаемость	Связь требований, кода, тестов и артефактов.	Сохранение полной истории изменений.	Полнота прослеживаемости.
Риск нарушения безопасности и конфиденциальности	DevSecOps, разграничение доступа, управление секретами.	Блокирование релиза при наличии неисправленных критических уязвимостей или подтвержденных нарушений доступа.	Покрытие проверками, число неисправленных критических нарушений на момент выпуска.
Ухудшение цифрового сервиса после релиза	Мониторинг критического пользовательского маршрута.	Возможность автоматического или ручного отката.	Успешность операций, доступность сервиса, число обращений.

Расчетная апробация выполнена путем сценарного сопоставления традиционной организации разработки и целевого DevOps-процесса в условной корпоративной информационной системе. Цель моделирования состояла не в прогнозировании фактических результатов конкретной организации, а в проверке внутренней согласованности предложенной модели и количественной оценке различий между двумя заранее сформированными сценариями.

В обоих сценариях период наблюдения составлял 90 календарных дней, число изменений – 24, функциональный состав и архитектура сервиса были одинаковыми. Традиционная модель предполагала пакетные релизы и последовательное согласование,

DevOps – сокращение межэтапных ожиданий, автоматизацию проверок и более частый выпуск небольших изменений. Релизные циклы могли перекрываться, поэтому их суммарная длительность не приравнивалась к периоду наблюдения.

Для оценки сценариев использовались показатели, частично основанные на подходе DORA: частота развертываний, доля неуспешных изменений и время восстановления после неуспешного развертывания. Продолжительность релизного цикла, уровень автоматизации, покрытие обязательными проверками и полнота прослеживаемости рассматривались как дополнительные авторские показатели (таблица 5).

Таблица 5. Условия и заданные параметры сценарного моделирования

Параметр	Традиционный сценарий	DevOps-сценарий
Период наблюдения, календарных дней	90	90
Количество запланированных изменений	24	24
Согласование изменения, рабочих дней	4,0	1,0
Подготовка сборки и среды, рабочих дней	2,0	0,5
Тестирование, рабочих дней	3,0	1,5
Проверка безопасности, рабочих дней	2,0	0,5
Развертывание, рабочих дней	1,0	0,5
Количество производственных развертываний	6	18
Количество неуспешных развертываний	2	2
Совокупное время восстановления, ч	16	5
Автоматизированные операции / общее число операций	8 / 20	17 / 20
Выполненные проверки / обязательные проверки	6 / 9	9 / 9
Прослеживаемые изменения / общее число изменений	10 / 24	24 / 24

Для сопоставления традиционного и целевого сценариев использовались единые перечни из 20 типовых

операций релизного цикла и 9 обязательных проверок, включая code review, модульное и интеграционное

тестирование, SAST, DAST, анализ зависимостей, проверку секретов, валидацию IaC и сканирование контейнерных образов. Различия определялись способом выполнения операций и полнотой включения проверок в релизный процесс. Все параметры фиксировались до расчетов, не отражали данные конкретной организации и использовались для оценки скорости, устойчивости, автоматизации, контрольного покрытия и прослеживаемости.

Продолжительность релизного цикла определялась как сумма длительности его этапов:

$$LT = \sum_{i=1}^n t_i, \quad (1)$$

где LT – продолжительность релизного цикла, t_i – суммарное время выполнения и ожидания на i -м этапе, n – количество этапов.

Частота производственных развертываний рассчитывалась по формуле:

$$DF_m = \frac{30N_{dep}}{T_d}, \quad (2)$$

где DF_m – частота развертываний в месяц, N_{dep} – количество производственных развертываний, T_d – продолжительность периода наблюдения в календарных днях.

Доля неуспешных изменений определялась следующим образом:

$$CFR = \frac{N_{fail}}{N_{dep}} \times 100\%, \quad (3)$$

где CFR – доля неуспешных изменений, N_{fail} – количество развертываний, вызвавших отказ, откат или необходимость дополнительного исправления, N_{dep} – общее количество производственных развертываний.

Среднее время восстановления после неуспешного развертывания рассчитывалось по формуле:

$$FDRT = \frac{\sum_{j=1}^{N_{fail}} t_{recovery,j}}{N_{fail}}, \quad (4)$$

где $FDRT$ – среднее время восстановления, $t_{recovery,j}$ – продолжительность восстановления после j -го инцидента, N_{fail} – количество неуспешных развертываний.

Уровень автоматизации определялся как доля автоматизированных операций в релизном цикле, покрытие контрольными процедурами – как доля выполненных обязательных проверок, полнота прослеживаемости – как доля изменений, связанных с требованием, кодом, тестами, артефактом и производственным развертыванием.

Рассчитываемые показатели были разделены на две группы. К первой относились показатели скорости и устойчивости релизного процесса: продолжительность релизного цикла, частота производственных развертываний, доля неуспешных изменений и среднее время восстановления. Ко второй – нормированные характеристики сценария: уровень автоматизации, покрытие обязательными проверками и полнота прослеживаемости. Показатели второй группы рассчитывались как относительные величины на основе заданного состава операций и изменений и характеризовали различия в организации рассматриваемых сценариев.

Результаты расчета представлены в таблице 6.

Таблица 6. Расчетные показатели традиционного и DevOps-сценариев

Показатель	Традиционный сценарий	DevOps-сценарий	Изменение
Продолжительность релизного цикла, рабочих дней	12,0	4,0	снижение на 66,7%
Частота развертываний в месяц	2,0	6,0	увеличение в 3 раза
Доля неуспешных изменений, %	33,3	11,1	снижение на 22,2 п.п.
Среднее время восстановления, ч	8,0	2,5	снижение на 68,8%
Уровень автоматизации, %	40,0	85,0	увеличение на 45,0 п.п.
Покрываемость обязательными проверками, %	66,7	100,0	увеличение на 33,3 п.п.
Полнота прослеживаемости, %	41,7	100,0	увеличение на 58,3 п.п.

При заданных параметрах продолжительность релизного цикла составила 12 рабочих дней в традиционном сценарии и 4 рабочих дня в DevOps-сценарии, что соответствует расчетному снижению на 66,7%. Частота производственных развертываний увеличилась с двух до шести в месяц.

В обоих сценариях было задано по два неуспешных развертывания. Поэтому при увеличении общего числа развертываний с 6 до 18 расчетная доля неуспешных изменений снизилась с 33,3 до 11,1%. Данный результат не означает эмпирически установленного снижения количества отказов, а показывает, что в рассматриваемом сценарии повышение частоты поставки не сопровождалось увеличением их абсолютного числа. Среднее время восстановления, рассчитанное на основе заданного совокупного времени устранения последствий двух неуспешных развертываний, составило соответственно 8 и 2,5 ч.

Уровень автоматизации составил 40% в традиционном сценарии и 85% в DevOps-сценарии. Покрытие обязательными проверками увеличилось с 66,7 до 100%, а полнота прослеживаемости – с 41,7 до 100%. Эти показатели характеризуют заданную организацию сценариев и используются для количественного сопоставления традиционного и целевого релизных процессов.

Сценарное моделирование подтвердило внутреннюю

согласованность подхода и выявило потенциальные различия между традиционным и целевым релизными процессами. Улучшение расчетных показателей скорости и устойчивости обусловлено принятыми допущениями, а уровни автоматизации, контрольного покрытия и прослеживаемости отражают заданные параметры сценариев. Практическая результативность модели требует проверки на эксплуатационных данных.

IV. ОБСУЖДЕНИЕ

Результативность DevOps в корпоративной среде определяется согласованностью организационных и технологических преобразований [13], [14]. Автоматизация CI/CD, IaC и управляемого развертывания дает устойчивый эффект при наличии кросс-функционального взаимодействия, распределенной ответственности, автономности продуктовых команд и единых корпоративных стандартов. Это соответствует исследованиям, связывающим зрелость DevOps с развитием компетенций, эффективным формированием команд и устранением межфункциональных барьеров [15], [16].

Этические аспекты применения DevOps связаны с ответственностью за изменения, прозрачностью автоматизированных операций, защитой данных и недопустимостью приоритета скорости выпуска над надежностью и интересами пользователей. Предложенная модель переводит эти требования в проверяемые условия релизного процесса: прослеживаемость изменений, разграничение доступа и контроль критических операций. Ограничением исследования является сценарный характер апробации, поэтому практическая результативность модели требует проверки на данных реальных корпоративных информационных систем.

V. ЗАКЛЮЧЕНИЕ

Применение DevOps в корпоративной среде следует рассматривать как комплексное преобразование организационных, технологических и этико-контрольных механизмов. Его результативность определяется сочетанием кросс-функционального взаимодействия, распределенной ответственности, автономности продуктовых команд, автоматизации CI/CD, IaC, DevSecOps и наблюдаемости. Ускорение поставки ПО должно сопровождаться прослеживаемостью изменений, защитой данных, прозрачностью операций и контролем критических решений.

Предложенная модель связывает ограничения корпоративной информационной системы с соответствующими DevOps-практиками, условиями их безопасного применения и измеримыми показателями результата. Сценарное моделирование показало, что при принятых допущениях целевой DevOps-сценарий характеризуется меньшей продолжительностью релизного цикла, меньшим средним временем восстановления и более высокой частотой развертываний, а также повышенными значениями автоматизации, контрольного покрытия и

прослеживаемости. При одинаковом заданном количестве неуспешных развертываний увеличение общего числа выпусков обусловило снижение их относительной доли. Гипотеза исследования получила частичную расчетную поддержку в части показателей скорости и восстановления, тогда как влияние предложенных решений на критические операционные и информационные риски требует проверки на эксплуатационных данных реальных корпоративных систем.

БИБЛИОГРАФИЯ

- [1] Т. О. Миронов и Н. Н. Середенко, "Построение модели расчета времени выпуска доработок программного обеспечения после внедрения системы управления релизным циклом," *Научный результат. Информационные технологии*, Т. 9, № 3, С. 43-54, 2024. DOI: 10.18413/2518-1092-2024-9-3-0-5.
- [2] M. H. Tanzil, M. Sarker, G. Uddin, and A. Iqbal, "A mixed method study of DevOps challenges," *Information and Software Technology*, vol. 161, art. 107244, 2023. DOI: 10.1016/j.infsof.2023.107244.
- [3] L. M. Amugongo, A. Kriebitz, A. Boch, and C. Lütge, "Operationalising AI ethics through the agile software development lifecycle: A case study of AI-enabled mobile health applications," *AI and Ethics*, vol. 5, pp. 227-244, 2025. DOI: 10.1007/s43681-023-00331-3.
- [4] А. В. Блинов и С. В. Беззатеев, "Защита DevOps-пайплайнов: автоматизация безопасности в рамках DevSecOps," *Проблемы информационной безопасности. Компьютерные системы*, № 2, С. 21-34, 2025. DOI: 10.15827/0236-235X.151.524-533.
- [5] П. С. Цынгальев, "Применение методологии DevOps при разработке приложений для высокопроизводительных вычислений," *Программные продукты и системы*, Т. 38, № 3, С. 524-533, 2025. DOI: 10.15827/0236-235X.151.524-533.
- [6] А. В. Блинов и С. В. Беззатеев, "DevSecOps: объединение процессов разработки и безопасности," *Вопросы кибербезопасности*, № 2 (66), С. 78-89, 2025. DOI: 10.21681/2311-3456-2025-2-78-89.
- [7] А. С. Ермаков, "Проблемы применения методологии DevOps в рамках российского бизнеса," *Вестник НГУЭВ*, № 2, С. 166-181, 2023. DOI: 10.34020/2073-6495-2023-2-166-181.
- [8] А. В. Уткин и Ю. Д. Козлова, "Особенности тестирования программного обеспечения в рамках методологий разработки Waterfall, Agile и DevOps," *Парадигма*, № 4-2, С. 146-154, 2025.
- [9] R. Amaro, R. Pereira, and M. M. da Silva, "Mapping DevOps capabilities to the software life cycle: A systematic literature review," *Information and Software Technology*, vol. 177, art. 107583, 2025. DOI: 10.1016/j.infsof.2024.107583.
- [10] D. Ahuja, "DevOps and ethical AI: Ensuring responsible deployment," *Sarcouncil Journal of Multidisciplinary*, vol. 5, no. 6, pp. 1-14, 2025.
- [11] R. Alidoosti, P. Lago, M. Razavian, and A. Tang, "Exploring ethical values in software systems: A systematic literature review," *Journal of Systems and Software*, vol. 226, art. 112430, 2025. DOI: 10.1016/j.jss.2025.112430.
- [12] L. Prates and R. Pereira, "DevSecOps practices and tools," *International Journal of Information Security*, vol. 24, art. 11, 2025. DOI: 10.1007/s10207-024-00914-z.
- [13] J. Díaz, J. Pérez, I. Alves, F. Kon, L. A. F. Leite, P. Meirelles, and C. Rocha, "Harmonizing DevOps taxonomies: A grounded theory study," *Journal of Systems and Software*, vol. 208, art. 111908, 2024. DOI: 10.1016/j.jss.2023.111908.
- [14] J. Jayakody and W. Wijayanayake, "Framework for DevOps success factors in software development," *International Journal of Information Systems and Change Management*, vol. 15, no. 2, pp. 187-205, 2025. DOI: 10.1504/IJISCM.2025.148633.
- [15] Е. А. Власова, "Разработка модели автоматизированной системы формирования проектных кросс-функциональных команд на предприятии," *Известия Самарского научного центра Российской академии наук*, Т. 27, № 3, С. 58-66, 2025. DOI: 10.37313/1990-5378-2025-27-3-58-66.
- [16] H. E. Korkmaz and M. N. Aydin, "An empirical study on performance comparisons of different types of DevOps team formations," *Frontiers in Computer Science*, Vol. 7, art. 1554299, 2025. DOI: 10.3389/fcomp.2025.1554299.

Application of DevOps in corporate information systems: organizational conditions, technological solutions, and ethical constraints

M. Khlystun, V.Yu. Chertovikova, I. Perelekhov, E. Lim

Abstract — The article examines the organizational, technological, and ethical aspects of applying DevOps in corporate information systems. The study aims to develop a model for selecting and evaluating DevOps practices that links release-process constraints with corresponding managerial, technological, and control measures. Comparative, systems, and structural-logical analyses were employed, together with scenario-based quantitative modeling of traditional and DevOps approaches. A matrix was proposed that matches each identified constraint with an applicable solution, a control condition, and a measurable outcome. Under the predefined parameters, the application of DevOps was characterized by shorter release-cycle and recovery times, higher deployment frequency, greater automation, broader control coverage, and more complete traceability. With the number of failed deployments held constant, their relative share decreased as the total number of releases increased. DevOps effectiveness depends on the alignment of organizational changes, technological automation, ethical requirements, and control procedures. The study results are computational in nature and require validation using operational data.

Keywords — CI/CD, change traceability, corporate information systems, DevOps, DevSecOps, release-cycle automation.

REFERENCES

- [1] T. O. Mironov and N. N. Seredenko, "Building a model for calculating the release time of software improvements after the implementation of the release cycle management system," *Research Result. Information Technologies*, vol. 9, no. 3, pp. 43-54, 2024. DOI: 10.18413/2518-1092-2024-9-3-0-5.
- [2] M. H. Tanzil, M. Sarker, G. Uddin, and A. Iqbal, "A mixed method study of DevOps challenges," *Information and Software Technology*, vol. 161, art. 107244, 2023. DOI: 10.1016/j.infsof.2023.107244.
- [3] L. M. Amugongo, A. Kriebitz, A. Boch, and C. Lütge, "Operationalising AI ethics through the agile software development lifecycle: A case study of AI-enabled mobile health applications," *AI and Ethics*, vol. 5, pp. 227-244, 2025. DOI: 10.1007/s43681-023-00331-3.
- [4] A. V. Blinov and S. V. Bezzateev, "Protection of DevOps pipelines: Automation of security within DevSecOps," *Information Security Problems. Computer Systems*, no. 2, pp. 21-34, 2025. DOI: 10.48612/jisp/nr14-x1nu-r6t9.
- [5] P. S. Tsyngalev, "Developing applications for high-performance computing: DevOps methodology," *Software & Systems*, vol. 38, no. 3, pp. 524-533, 2025. DOI: 10.15827/0236-235X.151.524-533.
- [6] A. V. Blinov and S. V. Bezzateev, "DevSecOps: Unifying development and security processes," *Cybersecurity Issues*, no. 2 (66), pp. 78-89, 2025. DOI: 10.21681/2311-3456-2025-2-78-89.
- [7] A. S. Ermakov, "Problems of using DevOps methodology in the framework of Russian business," *Vestnik NSUEM*, no. 2, pp. 166-181, 2023. DOI: 10.34020/2073-6495-2023-2-166-181.
- [8] A. V. Utkin and I. D. Kozlova, "Features of software testing in the framework of Waterfall, Agile and DevOps development methods," *Paradigma*, no. 4, pt. 2, pp. 146-154, 2025.
- [9] R. Amaro, R. Pereira, and M. M. da Silva, "Mapping DevOps capabilities to the software life cycle: A systematic literature review," *Information and Software Technology*, vol. 177, art. 107583, 2025. DOI: 10.1016/j.infsof.2024.107583.
- [10] D. Ahuja, "DevOps and ethical AI: Ensuring responsible deployment," *Sarcouncil Journal of Multidisciplinary*, vol. 5, no. 6, pp. 1-14, 2025.
- [11] R. Alidoosti, P. Lago, M. Razavian, and A. Tang, "Exploring ethical values in software systems: A systematic literature review," *Journal of Systems and Software*, vol. 226, art. 112430, 2025. DOI: 10.1016/j.jss.2025.112430.
- [12] L. Prates and R. Pereira, "DevSecOps practices and tools," *International Journal of Information Security*, vol. 24, art. 11, 2025. DOI: 10.1007/s10207-024-00914-z.
- [13] J. Díaz, J. Pérez, I. Alves, F. Kon, L. A. F. Leite, P. Meirelles, and C. Rocha, "Harmonizing DevOps taxonomies: A grounded theory study," *Journal of Systems and Software*, vol. 208, art. 111908, 2024. DOI: 10.1016/j.jss.2023.111908.
- [14] J. Jayakody and W. Wijayanayake, "Framework for DevOps success factors in software development," *International Journal of Information Systems and Change Management*, vol. 15, no. 2, pp. 187-205, 2025. DOI: 10.1504/IJISCM.2025.148633.
- [15] E. A. Vlasova, "The development of a model for an automated system of formation of project cross-functional teams at the enterprise," *Izvestiya of Samara Scientific Center of the Russian Academy of Sciences*, vol. 27, no. 3, pp. 58-66, 2025. DOI: 10.37313/1990-5378-2025-27-3-58-66.
- [16] H. E. Korkmaz and M. N. Aydin, "An empirical study on performance comparisons of different types of DevOps team formations," *Frontiers in Computer Science*, vol. 7, art. 1554299, 2025. DOI: 10.3389/fcomp.2025.1554299.