Ай-ТИ Фреш

АНАЛИТИЧЕСКИЙ РАЗБОР

CI/CD с нуля: релизы без простоев и человеческих ошибок

Почему ручной деплой — скрытый риск для бизнеса и что даёт конвейер автоматической поставки

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Релиз в вашей компании — это событие: разработчик собирает продукт вручную, админ по SSH копирует файлы на сервер, все скрещивают пальцы. Каждая выкатка — это простой сервиса и риск человеческой ошибки, откат занимает часы, а о проблемах вы узнаёте от клиентов. Пока изменения доходят до продакшена неделями, конкуренты выпускают обновления ежедневно — и забирают ваш рынок.

Почему это важно бизнесу

- Ручной деплой = простой сервиса: клиенты не могут оплатить, записаться или оформить заказ именно в момент выкатки
- Час простоя ИТ-систем обходится даже компании из 50 сотрудников примерно в 40 тысяч рублей прямых потерь
- Медленные релизы тормозят бизнес: нужная клиентам функция ждёт выкатки неделями — рынок занимают конкуренты
- Зависимость от одного «человека, который умеет деплоить» — риск при его отпуске, болезни или увольнении
- Ошибка при ручном обновлении систем с персональными данными — это ещё и риск по 152-ФЗ, а не только простой

СТАТИСТИКА

Проблема в цифрах

$460 млн

потерял Knight Capital за 45 минут из-за ошибки ручного деплоя на биржевые серверы

Источник: SEC, разбор инцидента, 2013

66–80%

инцидентов простоя ИТ-систем связаны с человеческим фактором

Источник: Uptime Institute, Outage Analysis 2025

40 000 ₽

в час — прямые потери компании из 50 сотрудников при простое ИТ-систем (оценка)

Источник: Киберпротект, 2026

5%

неудачных релизов у элитных ИТ-команд; деплой по требованию, восстановление — до часа

Источник: DORA State of DevOps Report, 2024

8,5 млн

компьютеров по всему миру вывело из строя одно обновление CrowdStrike без полноценных тестов

Источник: CNN Business, 2024

50%

российских DevOps-команд внедряют проверки безопасности на ранних этапах CI/CD-конвейера

Источник: State of DevOps Russia (Экспресс 42), 2025

КЕЙС 1 ИЗ 3

Ручной деплой на 8 серверов: минус $460 млн за 45 минут

Ситуация

Knight Capital Group — крупнейший маркетмейкер американского фондового рынка

Как развивались события

1. Инженер вручную обновлял ПО на восьми торговых серверах и пропустил один — на нём остался «мёртвый» код 2003 года
2. 1 августа 2012 при открытии биржи старый код ожил: система начала неконтролируемо скупать акции — около 4 млн сделок по 154 бумагам
3. 45 минут никто не мог понять, что происходит, и остановить торговлю
4. Убыток свыше $460 млн; через четыре месяца компанию поглотил конкурент

ПОСЛЕДСТВИЯ

Прямой убыток свыше $460 млн за 45 минут — компания потеряла независимость и была продана. Причина — не «плохой код», а отсутствие автоматизации выкатки: ни проверки одинаковости версий на серверах, ни автоматического отката, ни алертов на аномальное поведение системы.

ГЛАВНАЯ ОШИБКА / ВЫВОД

Ручное копирование файлов не масштабируется даже на 8 серверов. Автоматический конвейер деплоит одинаково на все машины, сверяет версии и умеет мгновенно откатываться.

Источник: SEC Release 34-70694; case study Henrico Dolfing

КЕЙС 2 ИЗ 3

Команда не на том сервере: потеря базы и 5 нерабочих бэкапов

Ситуация

GitLab — платформа хранения кода и CI/CD, сотни тысяч пользователей по всему миру

Как развивались события

1. 31 января 2017 инженер, устраняя отставание реплики, выполнил команду очистки каталога базы данных не на реплике, а на боевом сервере
2. Стёрто около 300 ГБ данных продакшена, сервис остановлен
3. Все пять механизмов резервного копирования оказались нерабочими: pg_dump молча не запускался, письма об ошибках отбрасывались почтовым фильтром
4. Восстанавливались ~18 часов из случайного снапшота шестичасовой давности; данные 5000 проектов и 700 новых аккаунтов потеряны безвозвратно

ПОСЛЕДСТВИЯ

Почти сутки простоя публичного сервиса и безвозвратная потеря части пользовательских данных за 6 часов работы. Спасла лишь случайная ручная копия, сделанная незадолго до аварии. Компания опубликовала постмортем, ставший хрестоматийным.

ГЛАВНАЯ ОШИБКА / ВЫВОД

Ручные операции на проде под нагрузкой — прямой путь к катастрофе. Бэкапы, восстановление из которых никто не репетировал, — это не бэкапы, а иллюзия защиты.

Источник: Официальный постмортем GitLab, 2017

КЕЙС 3 ИЗ 3

Непротестированное обновление положило 8,5 млн компьютеров

Ситуация

CrowdStrike — мировой вендор кибербезопасности; пострадали авиакомпании, банки и клиники по всему миру

Как развивались события

1. 19 июля 2024 вендор разослал всем клиентам сразу обновление конфигурации, не прошедшее полноценное тестирование
2. Ошибка в файле вызывала «синий экран» Windows: 8,5 млн машин ушли в циклическую перезагрузку
3. Остановились аэропорты, банки, больницы; Delta Air Lines отменила около 7000 рейсов за 5 дней
4. Ущерб только компаний Fortune 500 оценён в $5,4 млрд; Delta подала иск более чем на $500 млн

ПОСЛЕДСТВИЯ

Крупнейший ИТ-сбой в истории: одно обновление, выкаченное всем сразу без поэтапного раската, остановило критическую инфраструктуру по всему миру. Восстановление заняло дни — каждую машину чинили вручную. Ущерб Fortune 500 — $5,4 млрд.

ГЛАВНАЯ ОШИБКА / ВЫВОД

Любое обновление — даже «маленький конфиг» — обязано проходить тесты и поэтапную выкатку: сначала малая доля систем, потом остальные. Правило одинаково для вендора и для внутренних систем компании.

Источник: CNN Business; Wikipedia, 2024

КАК ДЕЛАТЬ НЕ НАДО

Типовые ошибки

✕ **Деплой руками по SSH**
Копирование файлов на сервер вручную — каждая выкатка уникальна и невоспроизводима; серьёзная ошибка — лишь вопрос времени

✕ **Тесты «когда-нибудь потом»**
Конвейер без автотестов просто быстрее доставляет баги в продакшен. Начните с 20–30 unit-тестов критичной бизнес-логики

✕ **Нет отработанного отката**
Если откат — это «найти старую версию и вспомнить, как ставили», простой растянется на часы вместо секунд

✕ **Секреты в коде и скриптах**
Пароли БД и API-ключи в репозитории — готовая утечка; храните их в защищённом хранилище секретов CI-системы

✕ **Нет тестового контура**
Без staging-среды изменения впервые встречаются с реальными данными уже на ваших клиентах

✕ **Миграции базы вручную**
SQL «руками на проде» нельзя повторить и откатить; миграции должны версионироваться и выполняться через конвейер

✕ **Один незаменимый «релизчик»**
Все знания о выкатке — в голове одного человека: его отпуск, болезнь или увольнение парализуют релизы

✕ **Игнорирование проверок безопасности**
Сканеры кода и зависимостей ловят уязвимости до продакшена; для систем с данными под 152-ФЗ это вопрос ещё и штрафов

РЕКОМЕНДАЦИИ И ПРАКТИКИ

Как правильно

МИНИМУМ

- Версионируйте всё в Git: код, скрипты деплоя, конфигурации
- Опишите деплой одним скриптом — одинаковым для теста и прода
- Настройте автосборку и запуск тестов на каждое изменение кода
- Заведите проверенный на практике откат на предыдущую версию

НОРМАЛЬНО

- Контейнеризируйте приложения (Docker) — одинаковая среда везде
- Разверните staging-контур и выкатывайте туда автоматически
- Встройте сканирование кода и зависимостей в конвейер
- Секреты — только в хранилище CI-системы, не в коде

ХОРОШО

- Blue-green или canary деплой: релизы без остановки сервиса
- Автооткат по метрикам: вырос уровень ошибок — версия откатилась сама
- Feature flags: включение функций без выкатки и на часть клиентов
- Миграции БД через Flyway/Liquibase, только обратно совместимые

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Можете ли вы выкатить обновление в рабочее время без остановки сервиса для клиентов?
- [ ] Сможете ли откатиться на предыдущую версию быстрее чем за 15 минут?
- [ ] Запускаются ли автотесты при каждом изменении кода, а не «перед большим релизом»?
- [ ] Сможет ли новый сотрудник выполнить релиз по инструкции, без «того самого админа»?
- [ ] Хранятся ли пароли и ключи вне кода — в защищённом хранилище секретов?
- [ ] Есть ли тестовый контур, где изменения проверяются до продакшена?
- [ ] Проверяли ли вы восстановление из резервной копии за последние 3 месяца?
- [ ] Узнаёте ли вы о сбое после релиза из мониторинга, а не из звонков клиентов?
- [ ] Сканируются ли код и зависимости на уязвимости до выхода в продакшен?
- [ ] Есть ли у вас план на случай недоступности зарубежных сервисов разработки (GitHub, Docker Hub)?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Аудит процесса релизов: как вы выкатываете сейчас, где риски простоя и потери данных — отчёт с планом работ
- Внедрение CI/CD под ключ: GitLab CI или российские платформы (GitFlic, GitVerse) — тесты, сканеры, откаты
- Настройка деплоя без простоя: контейнеры, staging-контур, blue-green, автоматический откат по метрикам
- Сопровождение конвейера: мониторинг пайплайнов, обновление сканеров безопасности, обучение вашей команды

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

НА ЧЁМ ОСНОВАН РАЗБОР

Источники

01 SEC Release 34-70694 по инциденту Knight Capital (sec.gov — 2013)

02 Case Study: The $440 Million Software Error at Knight Capital (henricodolfing.com — 2019)

03 Postmortem of database outage of January 31 (about.gitlab.com — 2017)

04 2024 CrowdStrike-related IT outages (en.wikipedia.org — 2024)

05 CrowdStrike outage: what caused it and how much it cost (cnn.com — 2024)

06 Accelerate State of DevOps Report (DORA) (dora.dev — 2024)

07 Annual Outage Analysis (uptimeinstitute.com — 2025)

08 Цена ИТ-простоев для российского МСБ (cyberprotect.ru — 2026)

09 К2Тех: простой ИТ-систем стал дороже для 39% компаний (tass.ru — 2026)

10 State of DevOps Russia (Экспресс 42) (express42.com — 2025)

11 Платформы-аналоги GitHub и GitLab в России (trends.rbc.ru — 2022)

Все данные пересказаны по открытым источникам; точность на дату публикации разбора.