



ИРКУТСКИЙ ПОЛИТЕХ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ



Методы безопасной разработки ПО

Руководитель лаборатории
сетевых систем и ИТ-инфраструктуры
институт ИТиАД ИРНИТУ:
Аношко Алексей Федорович
Telegram: @anoshko_education



Методология разработки – это...

Методология разработки ПО — это система принципов, методов и подходов, используемых для управления процессом создания программного обеспечения.

Это «карта», по которой команда движется от идеи до готового продукта.

Следование методологии обеспечивает:

- 1.Повышение предсказуемости:** Мы понимаем, когда будет готов результат.
- 2.Улучшение качества продукта:** Системный подход снижает количество багов.
- 3.Оптимизация сроков и бюджета:** Ресурсы распределяются эффективно.
- 4.Упрощение коммуникации:** Все участники (заказчик, разработчик, аналитик) говорят на одном языке. ГОСТ 33707-2016 (ISO/IEC 2382:2015)





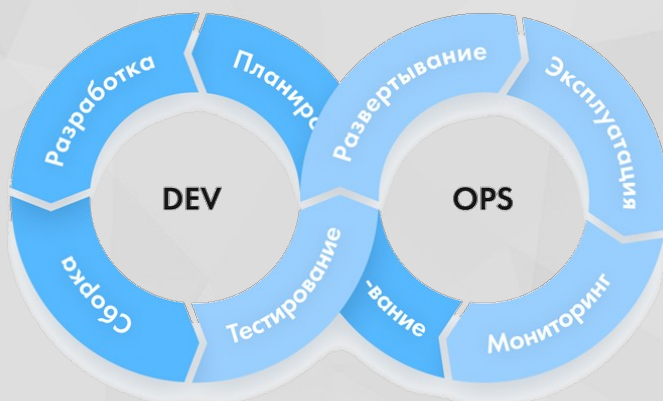
- Waterfall
- Agile, Scram, Kanban
- DevOps
- DevSecOps



Waterfall, Agile, DevOps...

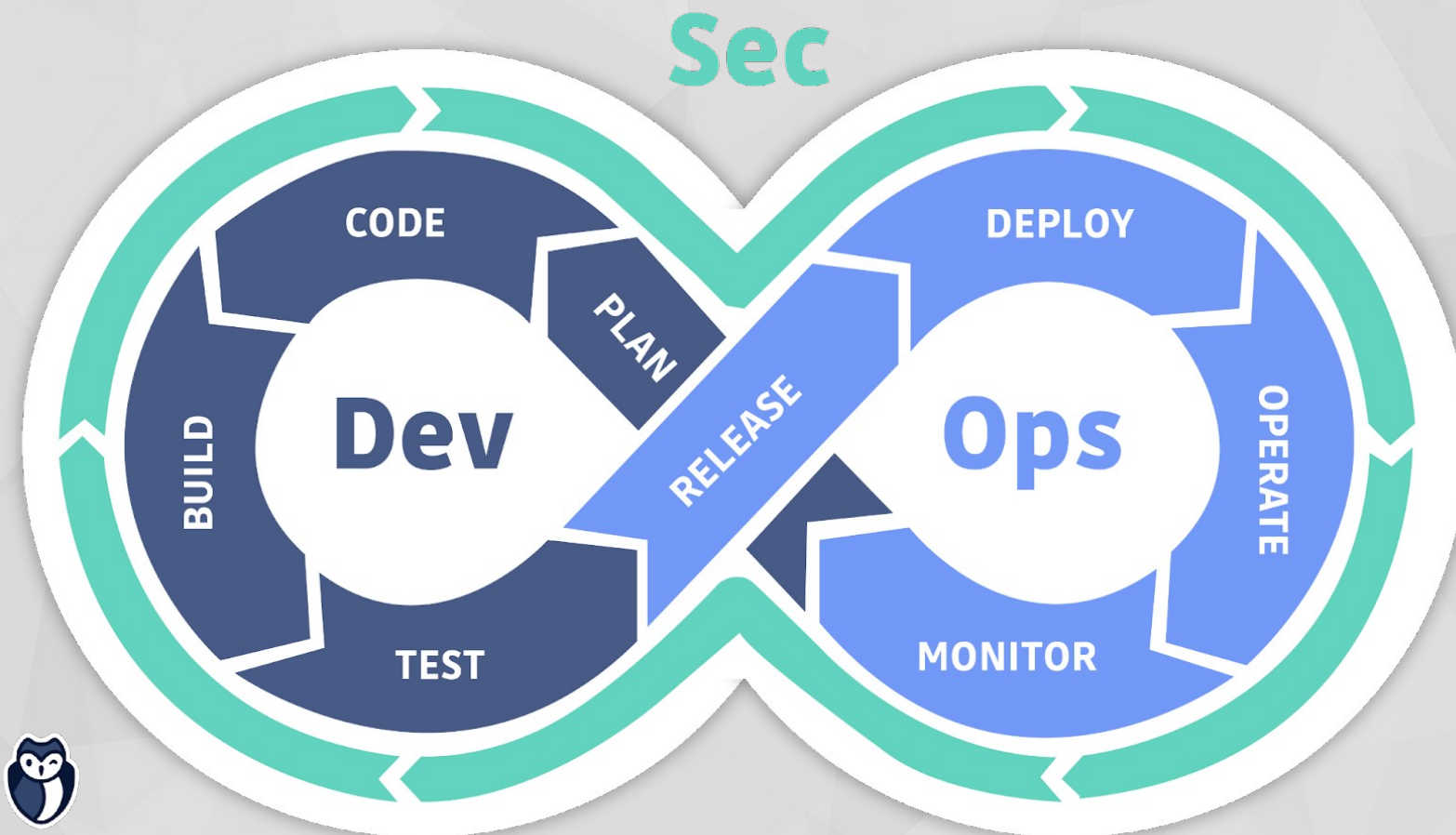


Agile методология





DevSecOps – общий контроль!





Agile – это...

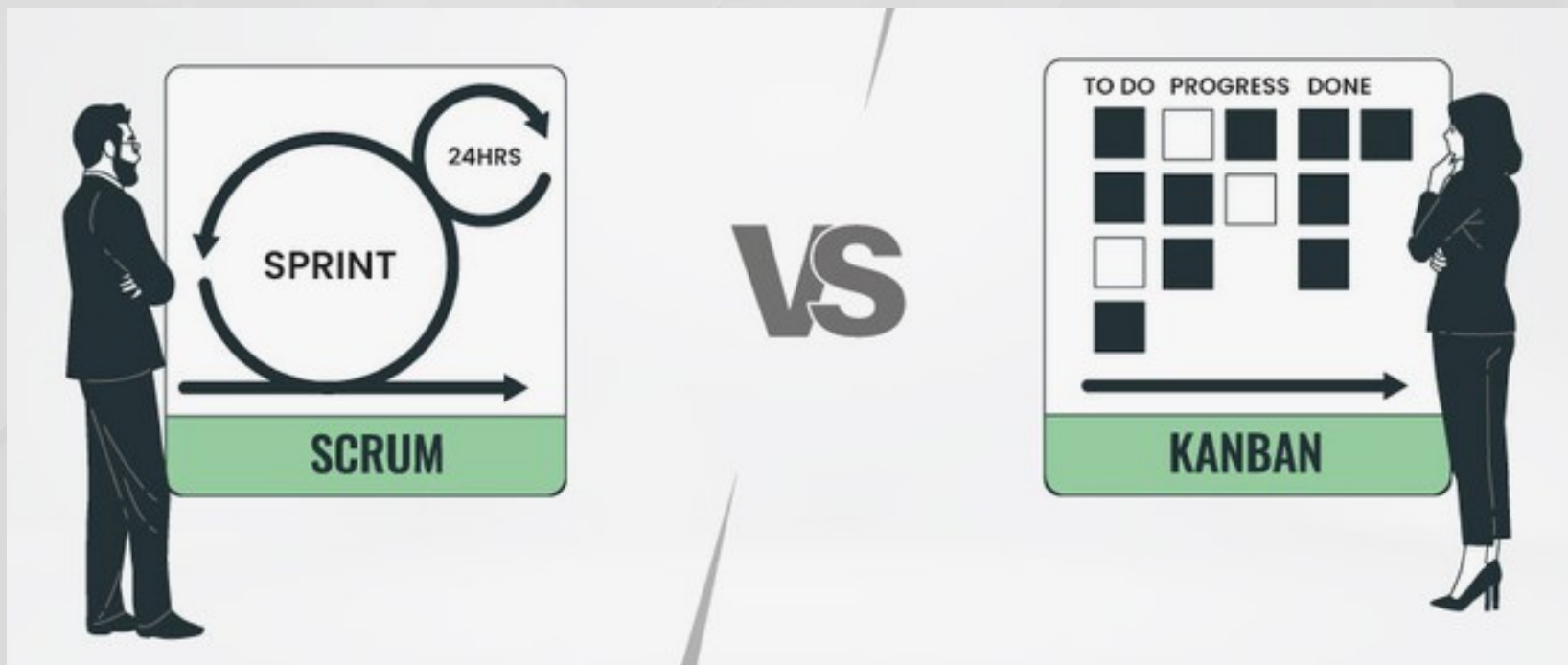
Agile (от англ. «гибкий», «подвижный») — это гибкий подход к управлению проектами, который фокусируется на итеративной разработке, постоянном взаимодействии с заказчиком и быстрой адаптации к изменениям.

Agile методология





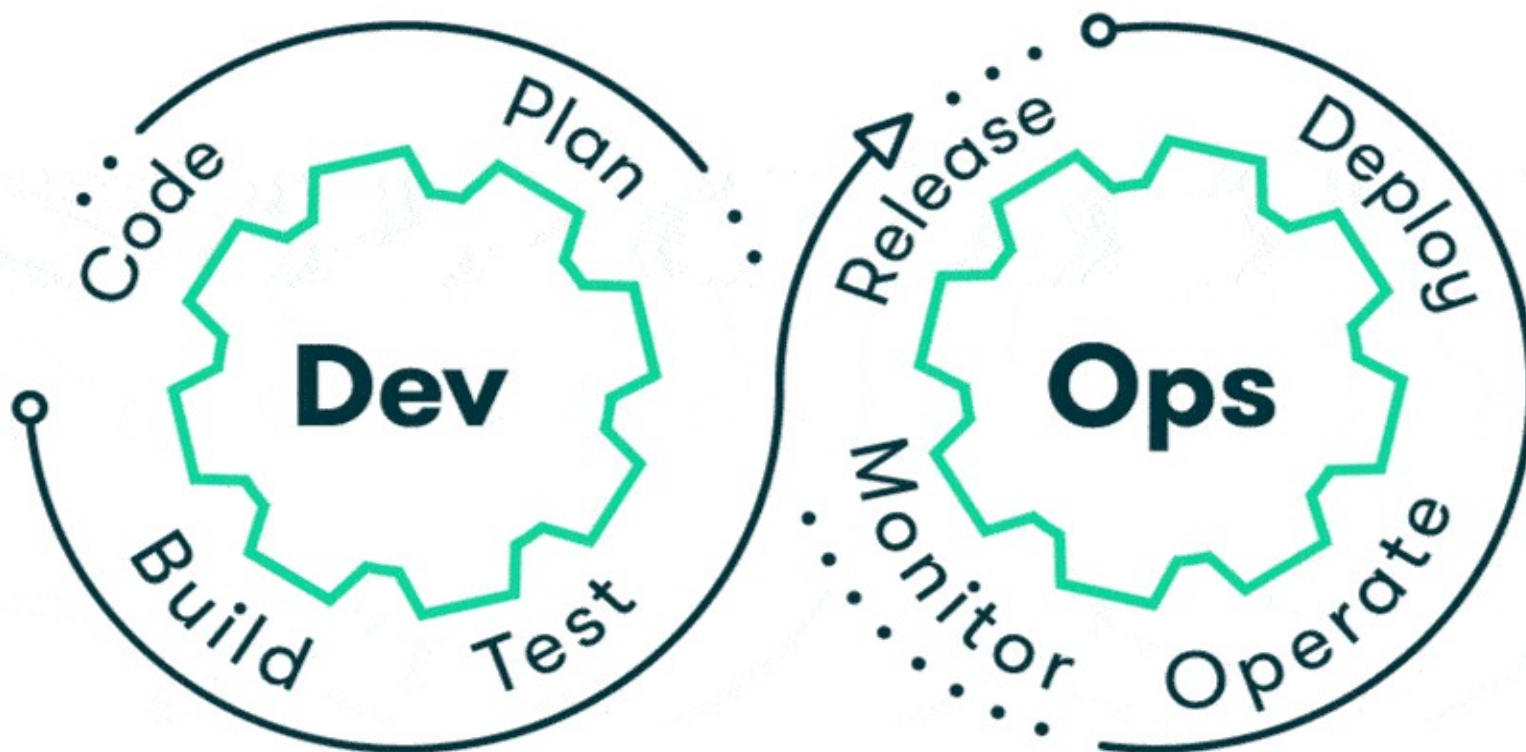
Scrum, Kanban





DevOps – это...

DevOps ([акроним](#) от [англ.](#) *development & operations*) — методология автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения/





Стандарты в разработке ПО (ГОСТ)

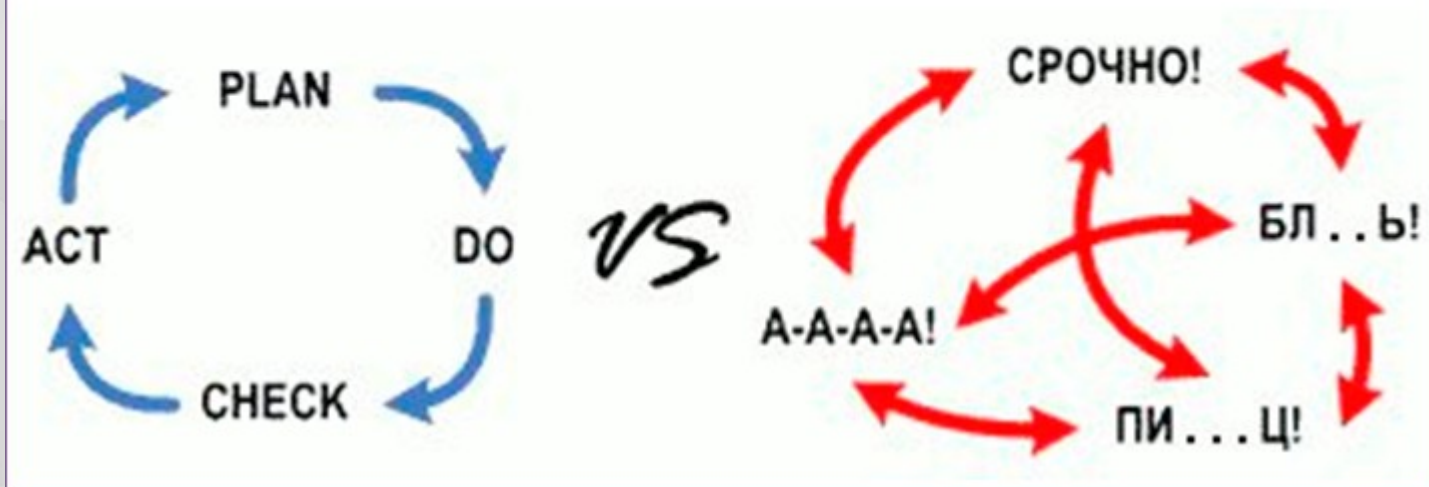
Взаимосвязь стандартов ГОСТ в жизненном цикле разработки ПО и их применение в современных методологиях:

1. ГОСТ 33707-2016 — задаёт терминологическую основу.
2. ГОСТ 34.602-2020 — формализует требования в техническом задании.
3. ГОСТ Р 56939-2024 — встраивает безопасность в жизненный цикл разработки.
4. ГОСТ Р ИСО 9241-110-2016 — задаёт принципы правильного взаимодействия человека с системой.



Принципы обратной связи

Цикл Деминга





Стандарты в разработке ПО (ГОСТ)

В переводе с делового канцелярского языка:

- ГОСТ 33707 - отвечает на вопрос «на каком языке мы описываем систему»;
- ГОСТ 34.602 — «что именно мы должны создать»;
- ГОСТ Р 56939 — «как создать это безопасно»;
- ГОСТ Р ИСО 9241-110 — «как сделать взаимодействие с системой понятным, устойчивым и удобным».



ГОСТ 33707-2016 как фундамент терминологии

ГОСТ 33707-2016 «Информационные технологии.

- Словарь» нужен не для формальности, а для устранения неоднозначности.
- В ИТ одни и те же слова часто употребляются в разных смыслах: «система», «интерфейс», «данные», «сервис», «процесс», «аутентификация», «ресурс», «защита», «ошибка», «отказ».
- Если эти термины не унифицированы, документы начинают противоречить друг другу.





Роль этого стандарта в жизненном цикле разработки можно описать так:

- создаёт единый понятийный аппарат;
- снижает риск неправильной интерпретации требований;
- помогает согласовывать документацию;
- позволяет корректно формулировать требования в ТЗ, документации, архитектурных решениях и тестах.





Предположим, в проекте есть требование: «Система должна обеспечивать контроль доступа к данным». Чтобы это требование стало пригодным для реализации, необходимо уточнить:

- кто является субъектом доступа;
- какие объекты защищаются;
- по каким правилам доступ разрешается или запрещается;
- как фиксируются события доступа;
- чем контроль доступа отличается от аутентификации и авторизации.

Именно терминологический стандарт помогает начать этот разговор профессионально и без путаницы.

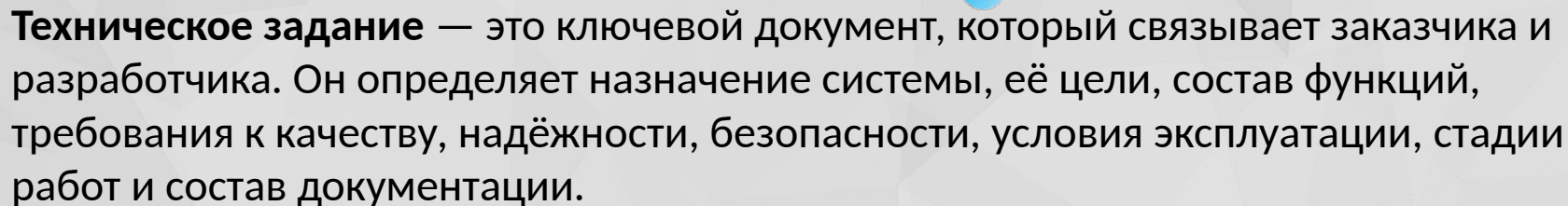


«Информационные технологии.

Комплекс стандартов на автоматизированные системы.

Техническое задание на создание автоматизированной системы» —

национальный стандарт РФ, вступивший в
силу 1 января 2022 года и заменивший
версию 1989 года





ГОСТ 34.602-2020 - переход от определений к требованиям

1. Как объяснил
заказчик



2. Как описал аналитик
в документации



3. Как спроектировал
архитектор



4. Как реализовал
разработчик



5. Как внедрили



6. Что хотел заказчик





В ТЗ должны быть отражены:

- функциональные требования;
- требования к информационному обеспечению;
- требования к защите информации;
- требования к интерфейсу и условиям эксплуатации;
- требования к составу и содержанию работ;
- критерии приёмки.





Если в ТЗ написано: «Система должна обеспечивать безопасную регистрацию и аутентификацию пользователей», этого недостаточно.

Для практической разработки надо уточнить:

- 1) какие механизмы аутентификации допустимы;
- 2) каковы требования к паролям или иным факторам;
- 3) как обрабатываются ошибки входа;
- 4) какие сообщения видит пользователь;
- 5) что происходит при множественных неудачных попытках;
- 6) как ведётся журналирование;
- 7) как интерфейс предотвращает ошибки и социальную инженерию.

Здесь уже видно, что одно требование из ТЗ тянет за собой и безопасность разработки, и эргономику диалога, и корректную терминологию.



Допустимые механизмы аутентификации

ГОСТ 34.602-2020 · разд. 4.6.4.8 — требования к защите информации от НСД

Парольная

Логин + пароль.
Обязательный базовый метод. Требования по сложности фиксируются в ТЗ.

Многофакторная (MFA)

ОТР/TOTP-коды, SMS, push-уведомления. Второй фактор для критичных ролей и операций.



Сертификаты / СКЗИ

PKI / КЭП — eToken, Рутокен. Рекомендуется при работе с ГИС и защищёнными данными.



Биометрия

Отпечаток, распознавание лица — дополнительный фактор. Допустимо при наличии регламента.

SSO / LDAP

Единый вход через корпоративный каталог (AD, LDAP, Kerberos). Снижает число точек входа.

Аппаратные токены

USB/NFC-токены, смарт-карты. Перечень допустимых устройств указывается в ТЗ на АС.



ГОСТ Р 56939-2024: безопасность не после разработки, а внутри неё

ГОСТ Р 56939-2024 «Защита информации. Разработка безопасного программного обеспечения. Общие требования»

В соответствии с Приказом ФСТЭК РФ от 01.12.2023 N 240,
"Об утверждении Порядка проведения сертификации процессов безопасной разработки программного обеспечения средств защиты информации"

Его особенно важно рассматривать в контексте современных методологий, потому что он описывает не отдельную функцию «проверить безопасность», а системную организацию процессов.

Связь с ТЗ

Очень важно: требования безопасной разработки не могут возникнуть из ниоткуда. Их нужно заложить ещё в техническом задании.

Если в ТЗ нет требований к защите информации, журналированию, разграничению доступа, обработке инцидентов, требованиям к коду и проверкам, то дальше разработчики **будут трактовать безопасность по-своему** или не будут учитывать её системно.



Для команды разработки это означает, что безопасность должна быть встроена:

- в анализ требований;
- в проектирование архитектуры;
- в разработку кода;
- в тестирование;
- в управление изменениями;
- в поставку и сопровождение.





- ГОСТ 34.602 фиксирует требования;



- ГОСТ Р 56939 задаёт, как эти требования безопасно реализовывать.



Единая логика взаимосвязи стандартов

Представим, что университет заказывает разработку автоматизированной системы управления учебным процессом.

Шаг 1. Использование ГОСТ 33707-2016

Команда определяет значения терминов:

- пользователь;
- роль;
- аутентификация;
- интерфейс;
- информационная система;
- ошибка;
- событие безопасности;
- данные;
- доступ.



Это позволяет исключить неоднозначные формулировки в документах.



Использование ГОСТ 34.602-2020

Формируется ТЗ, в котором указано:

- *цель системы;*
- *состав подсистем;*
- *функции пользователей;*
- *требования к безопасности;*
- *требования к интерфейсу;*
- *стадии разработки;*
- *виды испытаний.*



Например, в ТЗ можно зафиксировать, что система должна поддерживать разграничение прав для студентов, преподавателей, сотрудников деканата и администраторов.



Использование ГОСТ Р 56939-2024

На этапе проектирования и реализации команда:

- проводит анализ угроз;
- определяет защищаемые активы;
- вводит практики безопасного кодирования;
- проверяет изменения в репозитории;
- использует статический и динамический анализ;
- контролирует уязвимости библиотек;
- документирует решения по безопасности.





Использование ГОСТ Р ИСО 9241-110-2016

При разработке интерфейса личного кабинета:

- кнопки и действия делают однозначными;
- сообщения об ошибках — понятными;
- критичные операции — подтверждаемыми;
- навигацию — предсказуемой;
- формы ввода — устойчивыми к ошибкам;
- интерфейс — удобным для обучения и повседневного использования.





Единая логика взаимосвязи стандартов

В DevSecOps интерфейс тоже должен рассматриваться как поверхность атаки и как механизм предотвращения ошибок. Непонятное сообщение, ошибочная логика подтверждения, избыточно подробный текст ошибки, небезопасный ввод данных — всё это уже не только UX-проблема, но и проблема информационной безопасности.

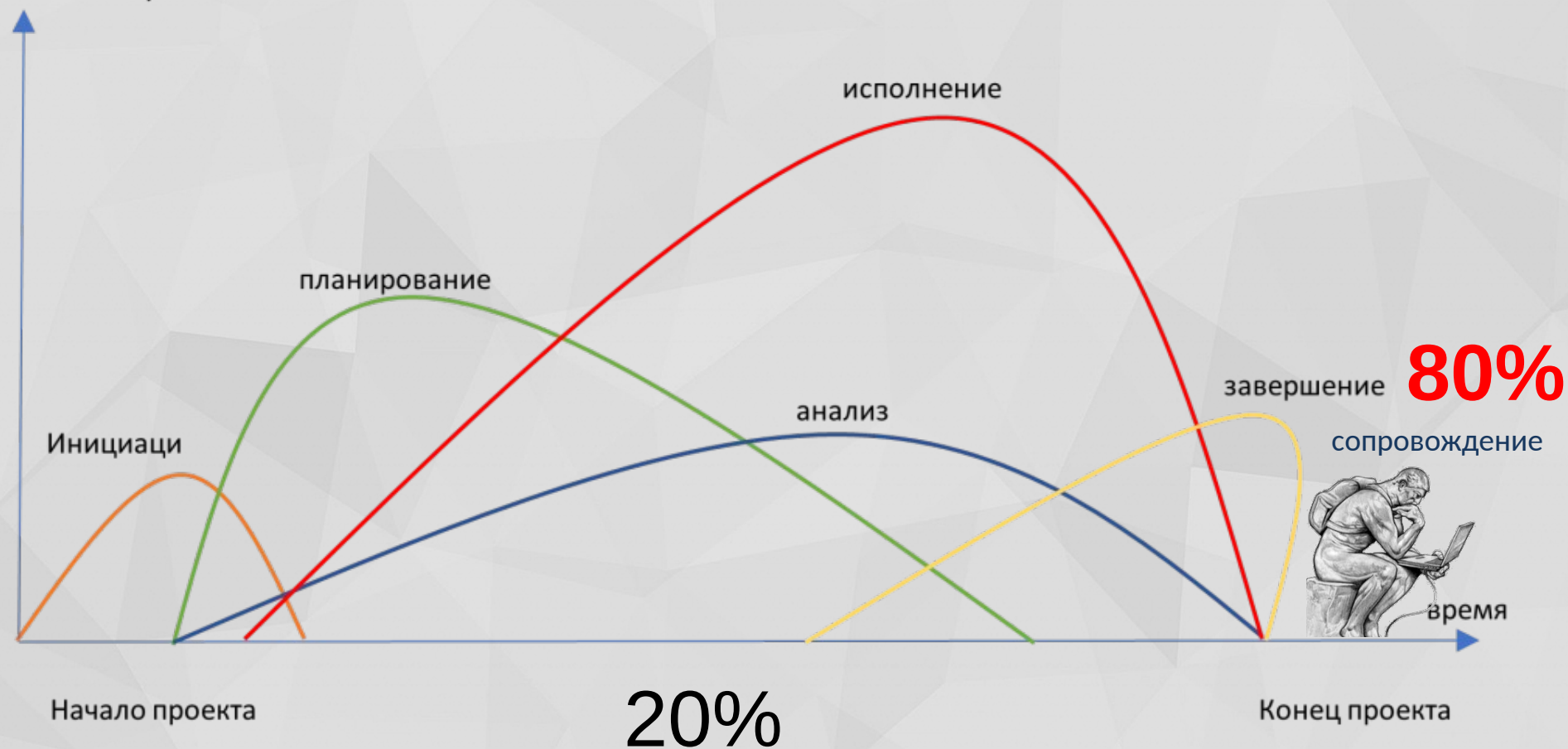


Таким образом, DevSecOps — это не только про сканеры и CI/CD. Это про согласованную инженерную систему, где требования, безопасность и эргономика работают совместно.



Цикл

Величина усилий





Вопросы

