Как обеспечить кибербезопасность компании в условиях Agile и DevOps? От теории к практике

The better the question. The better the answer.
The better the world works.

Building a better working world

Перспективные бизнес-модели, цифровые продукты и современные технологии размывают периметр безопасности и приносят новые риски

Современные индустриальные тренды показывают, что применение подхода Agile & DevOps становится одним из наиболее существенных аспектов для получения конкурентных преимуществ, таких как быстрая доставка продуктов и услуг потребителям, предложение уникальных продуктов и услуг одновременно с возможностью быстрого восстановления услуг в случае сбоев.

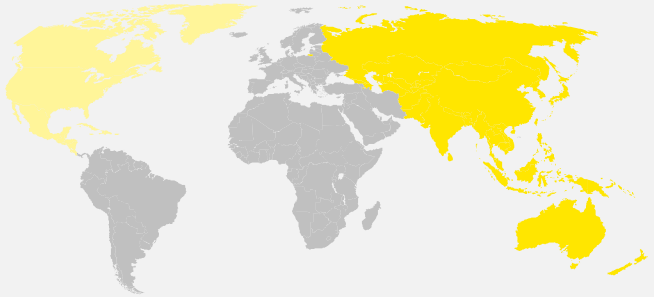
Основные преимущества

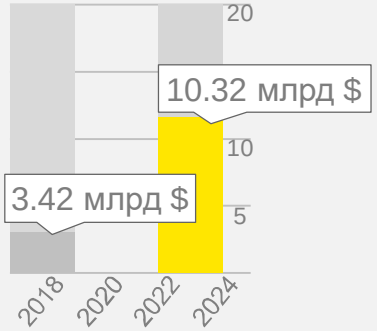
- Быстрая доставка услуг и продуктов клиентам, используя автоматизированные инструменты DevOps
- Оптимизация использования ресурсов за счёт автоматизации ежедневных рутинных рабочих задач
- Высокий уровень удовлетворенности заказчиков благодаря их интенсивному взаимодействию с разработчиками в модели процессов Agile
- Быстрое восстановление услуг в случае отказов

Основные риски

- Нарушения безопасности и соответствия регуляторным требованиям
- Разнородная сложно контролируемая инфраструктура
- Трудности с наймом высококвалифицированных и опытных инженеров DevOps и специалистов по безопасной разработке ПО
- Невозможность адаптации современного подхода к легаси-системам без их фундаментального рефакторинга

Ожидается, что объём рынка DevOps вырастет с 3.42 миллиардов долларов в 2018 до 10.31 миллиардов долларов к 2023, при совокупном годовом темпе роста (CAGR) 24,7% в течение прогнозируемого периода. При этом самый высокий темп роста ожидается в азиатско-тихоокеанском регионе, в то время как Северная Америка демонстрирует средние показатели*





*Источник: MarketsandMarkets Analysis

Лидирующие ИТ-компании повышают скорость разработки и безопасность кода за счет автоматизации и тесного взаимодействия с функцией кибербезопасности

Компания 1
США
Онлайн продажа товаров и услуг

- “You build it, you run it”. Отсутствует разделение полномочий разработчиков и специалистов по эксплуатации приложений в промышленном окружении.
- Каждая команда разработчиков полностью поддерживает один отдельный сервис.
- Применение практик безопасного кодирования и использование проверенных компонентов контролируется на стадии инспекции кода.

Компания 2
США
Онлайн провайдер медиаконтента

- “From gates to guardrails”. Проверка требований безопасности осуществляется после развертывания приложения в промышленном окружении, а не на этапах разработки и тестирования.
- Разработкой и поддержкой разработанного ПО занимается инженер «полного цикла».
- No Ops. Вся инфраструктура реализована на основе облачных технологий и является для разработчиков по умолчанию безопасной.

Компания 3
США
Онлайн продажа товаров

- Акцент в обеспечении безопасности приложений смещён на оперативное устранение уязвимостей в промышленном окружении, а не на их выявление на разных этапах разработки.
- Приоритет отдаётся уменьшению time-to-market с целью максимально быстрого устранения возможных уязвимостей.
- Следование принципам MVP (minimum viable product) при поставке функциональности в промышленное окружение.
- Используются разные реализации DevOps pipeline для систем разной критичности (отличаются в том числе набором security gates).

Компания 4
США
Социальная сеть

- Сканирование кода на безопасность осуществляется по инициативе разработчиков с помощью предоставляемого им сервиса.
- Результаты сканирования кода публикуются на общедоступном внутреннем дашборде, на котором разработчики могут акцептовать или отклонить выявленную уязвимость.
- Реализуется внешняя программа bug-bounty.

Компания 5
США
Разработка ПО и облачных сервисов

- Поддержка важности роли безопасности при разработке на уровне CEO.
- Автоматизирован процесс проверки следования практик безопасности на каждом этапе жизненного цикла.
- Security Engineering – отдельная структура, включающая в себя в том числе программистов-экспертов в кибербезопасности, помогает ставить процессы SDLC в новых командах.
- Гибкий баланс между зонами ответственности Security Engineer, разработчиками и другими участниками процесса, достигается за счет постоянной обратной связи.

Интеграция компонентов Sec в процесс DevOps осуществляется поэтапно с учетом уровня зрелости и культуры

PM 1 Интеграция требований ИБ в требования к продукту
Разработка методики формирования требований безопасности

BA 2 Идентификация и оценка рисков (evil user stories)
Разработка методики управления рисками информационной безопасности ПО

DEV 3 Анализ безопасности архитектуры
Разработка стандартных требований к безопасности архитектуры ПО

DEV 4 Контроль безопасности сторонних компонент
Внедрение шлюза качества SCA и определение критериев его прохождения

DEV 5 Статический анализ кода
Внедрение шлюза качества SAST и определение критериев его прохождения

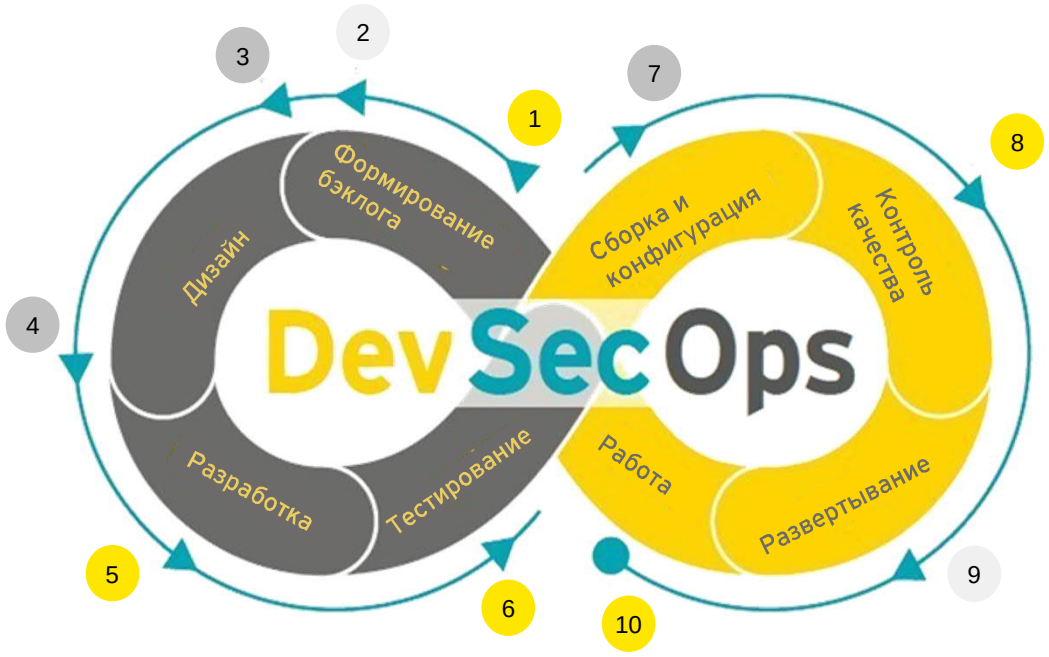
DEV 6 Динамический анализ приложения
Внедрение шлюза качества DAST и определение критериев его прохождения

OPS 7 Применение стандартов безопасной конфигурации инфраструктуры/облака
Разработка стандартных требований к конфигурации среды

BA 8 Контроль безопасности при приемке
Разработка методики контроля безопасности ПО

OPS 9 Непрерывный мониторинг
Внедрение процессов мониторинга и реагирования на инциденты ПО

PM 10 Тестирование на проникновение
Разработка регламента тестирования на проникновение и требований по устранению выявленных уязвимостей



1 Этап запуска
2 Этап развития
3 Этап совершенствования

PM Менеджер проекта
BA Бизнес-аналитик
DEV Разработчик
OPS Администратор

Evil User Stories в качестве сценариев для оценки киберрисков (примеры для ритейла)

Как спамер я хочу получить базу данных контактов покупателей строительных материалов для рекламы товаров и услуг в сфере строительства и ремонта

Как злонамеренный сотрудник магазина я хочу зарабатывать деньги на продаже скидок по программам лояльности

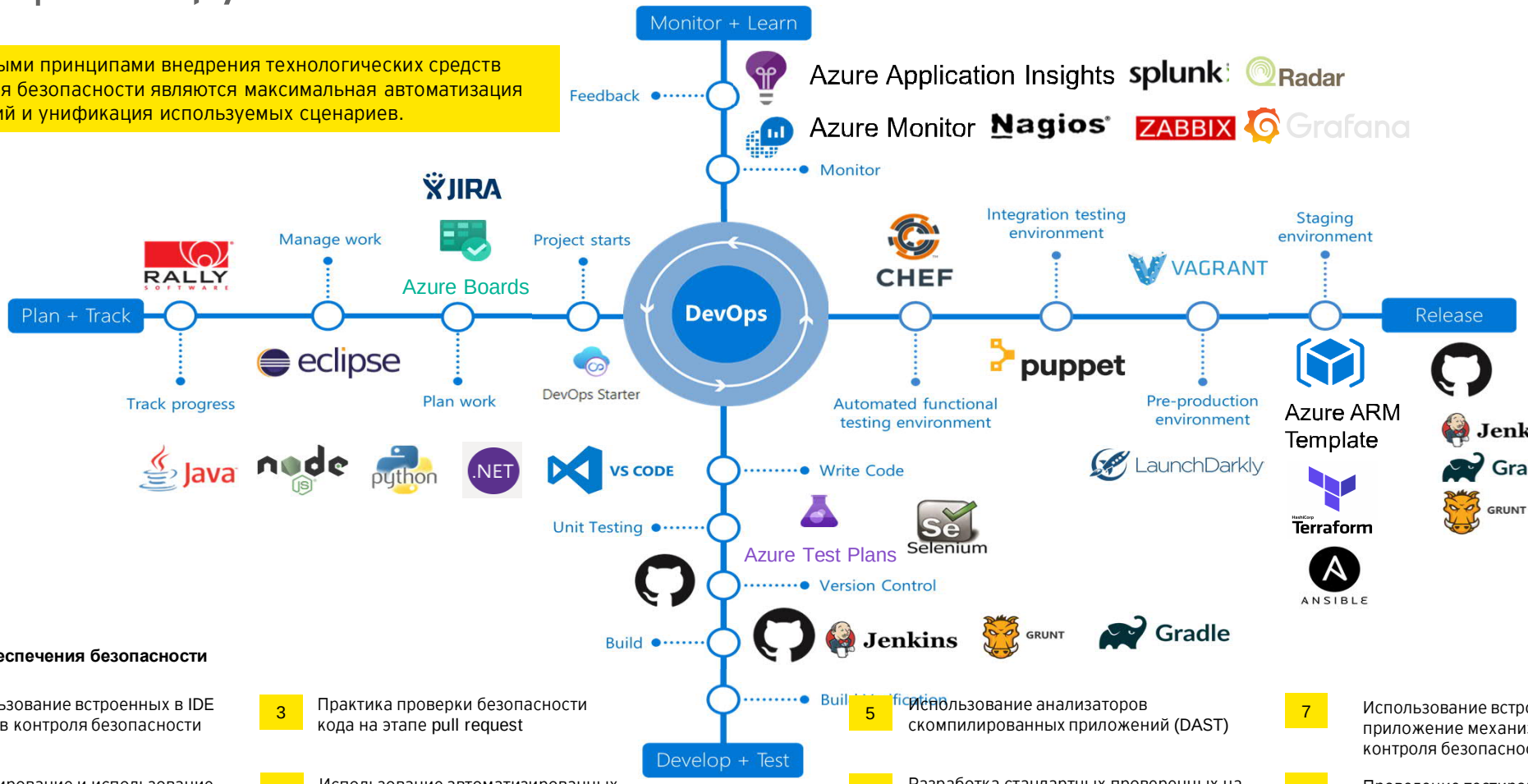
Как поставщик сервисов я хочу использовать контакты покупателей, чтобы продавать им свои услуги напрямую, без уплаты комиссии

Как злонамеренный сотрудник склада я хочу подменить данные в системе учета, чтобы совершить хищение товара

Как интернет-мошенник я хочу получить доступ к базе с карточными данными покупателей, чтобы совершать хищения их средств

DevOps инструменты

Основными принципами внедрения технологических средств контроля безопасности являются максимальная автоматизация операций и унификация используемых сценариев.



Практики обеспечения безопасности

1. Использование встроенных в IDE средств контроля безопасности
2. Формирование и использование внутренних репозиториев проверенных программных компонентов
3. Практика проверки безопасности кода на этапе pull request
4. Использование автоматизированных анализаторов исходного кода (SAST)
5. Использование анализаторов скомпилированных приложений (DAST)
6. Разработка стандартных проверенных на безопасность образов для разворота инфраструктуры
7. Использование встроенных в приложение механизмов контроля безопасности (RASP)
8. Проведение тестирования на проникновение

Функция кибербезопасности требует адаптации под потребности продуктовых команд и особенности инфраструктуры

Группа безопасности приложений

4. Контроль безопасности сторонних компонент
5. Статический анализ кода
6. Динамический анализ приложения
10. Тестирование на проникновение

Группа безопасности инфраструктуры

7. Применение стандартов безопасной конфигурации инфраструктуры (системы, средства виртуализации, ЦОД, облака)

Группа мониторинга

9. Непрерывный мониторинг

Группа сопровождения Agile-команд

1. Интеграция требований ИБ в требования к продукту
2. Идентификация и оценка рисков (evil user stories)
3. Анализ безопасности архитектуры
8. Контроль безопасности при приемке

Способ и объем взаимодействия специалистов кибербезопасности с продуктовыми командами может варьироваться в зависимости от сложности и критичности продукта и области его развертывания

Первые шаги по адаптации функции кибербезопасности и интеграции DevSecOps

1

Анализ процессов разработки ПО и связанных рисков кибербезопасности

Проведение оценки безопасности процесса разработки ПО с использование методологии BSIMM и сравнение с мировыми государственными и публичными организациями, а также идентификация ключевых рисков и риск-факторов, связанных с недостатками процесса.

Результат: доведение до всех заинтересованных сторон информации о критичных рисках и мерах, необходимых для их снижения

2

Разработка рекомендаций по развитию архитектуры кибербезопасности и процесса безопасной разработки

Определение ключевых точек процесса разработки, на которых необходимо включение кибербезопасности. Разработка дорожной карты на два-три года по развитию архитектуры кибербезопасности и процесса безопасной разработки. Разработка рекомендаций по функционально-техническим требованиям к системам статического анализа исходного кода и динамического анализа приложений.

Результат: определение плана развития на ближайшие годы

3

Разработка рекомендаций по развитию процедур и методик безопасной разработки, тестирования и мониторинга

Разработка рекомендаций по обеспечению и контролю безопасности ПО, тестированию на проникновение и устранению выявленных уязвимостей

Результат: повышение стабильности работы и снижение количества ошибок и уязвимостей в ПО

Подход EY был успешно использован для улучшения процесса Secured Agile & DevOps в крупной организации

Область: Российская компания, ведущая самостоятельную активную разработку и развитие ПО

Задача: Оценка процесса Secured Agile & DevOps и разработка дорожной карты улучшений на 2 года

Этапы проекта	Описание этапа	Результаты этапа
Оценка	Оценка текущего процесса разработки ПО с использованием фреймворков BSIMM и SAMM по 12 параметрам	Определён базовый уровень для дальнейшего улучшения
Идентификация рисков	Выявление рисков текущего состояния и ключевых моментов для улучшения	Определена целевая модель процесса на основе наиболее критичных рисков текущего состояния.
Анализ несоответствий	Определение действий, необходимых для устранения пробелов в проблемных областях бизнеса Клиента.	Ряд инициатив и мероприятий разработан на основе несоответствий между двумя состояниями процесса.
Рекомендации	Анализ лучшего и худшего международного опыта, адаптация сценария реализации инициатив с учётом особенностей Клиента	Точные сценарии реализации определены и подробно описаны
Дорожная карта	Приоритизация основных активностей, установление взаимосвязей и планирование реализации	Разработана дорожная карта реализации инициатив, у Клиента есть полное понимание действий, которые необходимо выполнить для улучшения процесса.

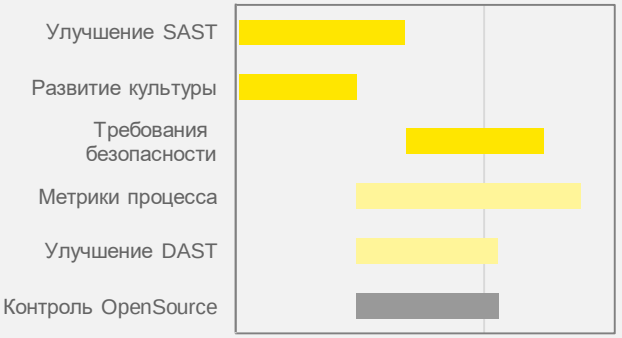
Пример результатов проекта по совершенствованию практик безопасной разработки

Оценка | Идентификация рисков | Анализ несоответствий | Рекомендации | Дорожная карта

Анализ несоответствий между текущим и целевым состояниями, разработка ряда активностей:

- Настройка инструментов автоматического тестирования безопасности (* AST)
- Внедрение риск-ориентированного подхода в процедурах тестирования безопасности
- Улучшение культуры обеспечения безопасности ПО
- Интеграция средств управления решениями с открытым исходным кодом
- Установка показателей качества процесса Agile & DevOps

Глобальные инициативы планируется реализовать в течение нескольких лет:

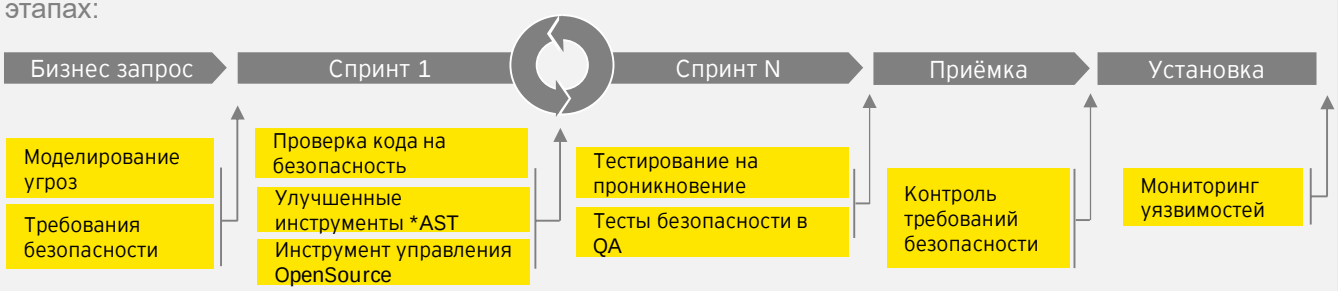


Сильные и слабые стороны процесса Secured Agile&DevOps определяются в соответствии с уровнями BSIMM и производится оценка рисков

Низкий уровень → Идентификация и уменьшение рисков

Высокий уровень → Действий не требуется

Общий процесс изменяется путем добавления действий по обеспечению безопасности на разных этапах:



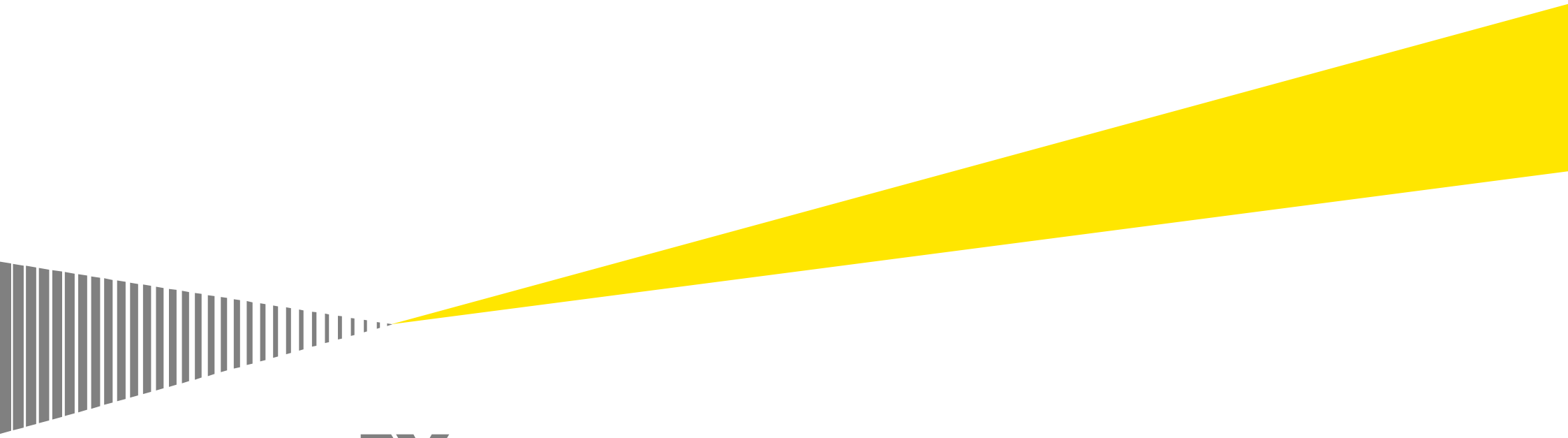
При совершенствовании процессов безопасной разработки рекомендуется учесть опыт известных ошибок из мировой практики

Ошибка	Последствие
Направить все ресурсы на поиск дефектов и не проработать процесс их устранения	Выявленные дефекты игнорируются командой, что, в свою очередь, отрицательно влияет на мотивацию специалистов, которые их выявляют
Использовать сложную для восприятия документацию при взаимодействии с разработчиками (политики, RACI-матрицы, фреймворки) без обучения	Документация не используется в работе и быстро становится неактуальной
Не учитывать среду, в которой работают команды (цели, сроки показатели эффективности)	Команды не выполняют функции по кибербезопасности из-за того, что они негативно влияют на достижение их целей
Требовать от команд скорость устранения дефектов, которую они не в состоянии обеспечить	Команда практически полностью перестает устранять дефекты
Не оценивать воздействие на бизнес при оценке критичности дефекта	Задачи по устранению дефектов, в том числе, критичных, получают низкий приоритет, поскольку команда не понимает последствий его эксплуатации
Не уделять достаточное внимание стабильности работы, производительности и масштабируемости инструментов, а также неготовность к быстрому росту нагрузки (например, при резком увеличении количества подключенных команд)	Возникновение проблем у команд в их собственных процессах из-за технических сбоев инструментов безопасности и, как следствие, негативное восприятие всего процесса
Проводить автоматизированное тестирование лишнего кода (например, статический анализ кода подключаемых компонентов)	Неприемлемая для команд продолжительность тестирования
Не уделять достаточного внимания снижению количества ложных срабатываний	Негативное восприятие процесса со стороны разработчиков, выполняющих триаж дефектов, и команды в целом
Проводить моделирование угроз без учета специфики приложения (отсутствие документации, слишком сложные потоки данных, недостаток информации по legacy-системам)	Невозможность построения модели угроз по выбранной методике и, как следствие, негативное восприятие практики со стороны команды
Привлекать команды к процессу без достаточной его проработки и тестирования	Дискредитация процесса в глазах разработчиков и трудности с внедрением его в будущем
Не делиться информацией по процессам безопасной разработки	Негативное отношение команд к процессам безопасной разработки из-за их непрозрачности

Спасибо за внимание

Сергей Машошин
Менеджер

Тел: +7 (495) 755-9700
Sergei.Mashoshin@ru.ey.com



EY
Совершенствуя бизнес,
улучшаем мир