ПРОЦЕСС БЕЗОПАСНОЙ РАЗРАБОТКИ

Как его выстроить без драк в переговорных

2022

Алина Новопольцева

«Инфосистемы Джет»

Email: AA.Novopoltseva@jet.su

ЧТО БУДЕМ ОБСУЖДАТЬ

1. В чем отличия процесса разработки от процесса безопасной разработки
2. Как строится жизненный цикл безопасной разработки
3. Часто встречающиеся проблемы
4. Причина возникновения этих проблем
5. Стандарты и лучшие практики
6. Взаимосвязанные процессы и роли
7. Инструменты и их адаптация
8. Оценка эффективности
9. Концептуальный план внедрения процесса

ЖИЗНЕННЫЙ ЦИКЛ БЕЗОПАСНОЙ РАЗРАБОТКИ

			DEV	TEST	PREPROD	PROD
Training	**Requirements**	**Design**	**Development**	**Verification**	**Release**	**Response**
• Анализ компетенций работников в части безопасной разработки • Обучение работников	• Определение требований ИБ к разрабатываемому ПО • Управление рисками ИБ	• Анализ поверхности атаки • Моделирование угроз • Проектирование архитектуры ПО с учетом требований ИБ и выявленных в ходе анализа слабых мест	• Использование стандартов безопасного программирования • Статический анализ кода • Анализ зависимостей	• Функциональное тестирование ИБ • Динамический анализ кода • Тестирование на проникновение • Фаззинг-тестирование	• Финальные проверки ИБ • Верификация релизов	• Управление конфигурацией и изменениями • Мониторинг угроз • Реагирование на инциденты • Регулярные проверки ИБ

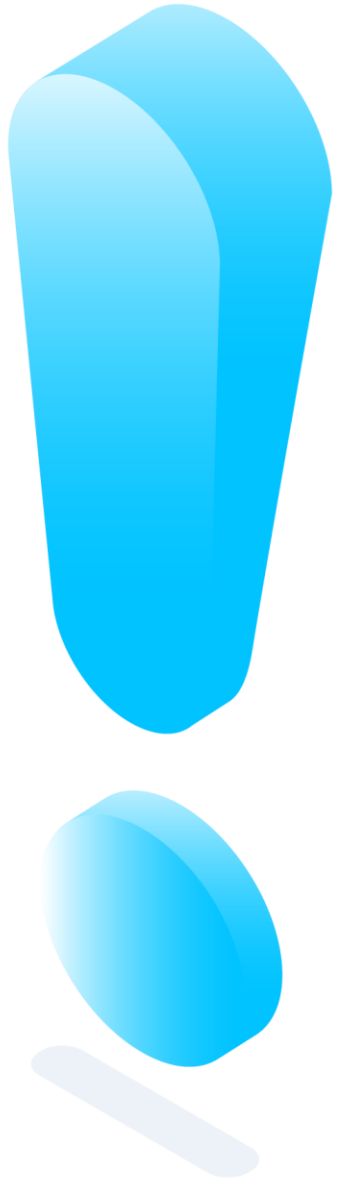
SSDLC (Security Software Development Lifecycle)

ЧАСТО ВСТРЕЧАЮЩИЕСЯ ПРОБЛЕМЫ

- Конфликт интересов разных подразделений
- Увеличение сроков выдачи ценности пользователю
- Специалисты ИБ не воспринимаются как участники процесса разработки
- Бесконечное увеличение тех. долга
- Увеличение стоимости устранения уязвимостей
- Отсутствие понимания, как сделать продукт безопасным

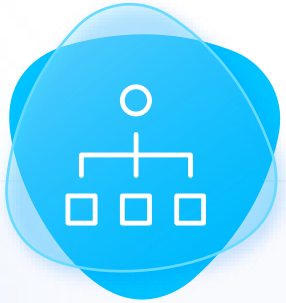
ПРИЧИНЫ ВСЕХ ПРОБЛЕМ

1. Выполнение «в лоб» требований регуляторов
2. Отсутствие видения конечного результата
3. Отсутствие драйверов процесса
4. Не зафиксированы скоуп задач и ответственность участников процесса
5. Попытка сделать все и сразу
6. Отсутствие необходимых ресурсов/компетенций
7. Отсутствие измеримых показателей эффективности
8. ИБ только контролирует, но не содействует



КАК ПРАВИЛЬНО ПОДОЙТИ К ВЫСТРАИВАНИЮ ПРОЦЕССА

Методология



Ресурсы
и зависимости

Автоматизация

Оценка
эффективности

БЛОКИ АКТИВНОСТЕЙ

Методология

Ресурсы
и зависимости

Автоматизация

Оценка
эффективности

ВОПРОСЫ, НА КОТОРЫЕ ДОЛЖНЫ ОТВЕТИТЬ ДОКУМЕНТЫ

Что должно быть сделано?

Основные положения, касающиеся процесса безопасной разработки

Как это должно быть сделано?

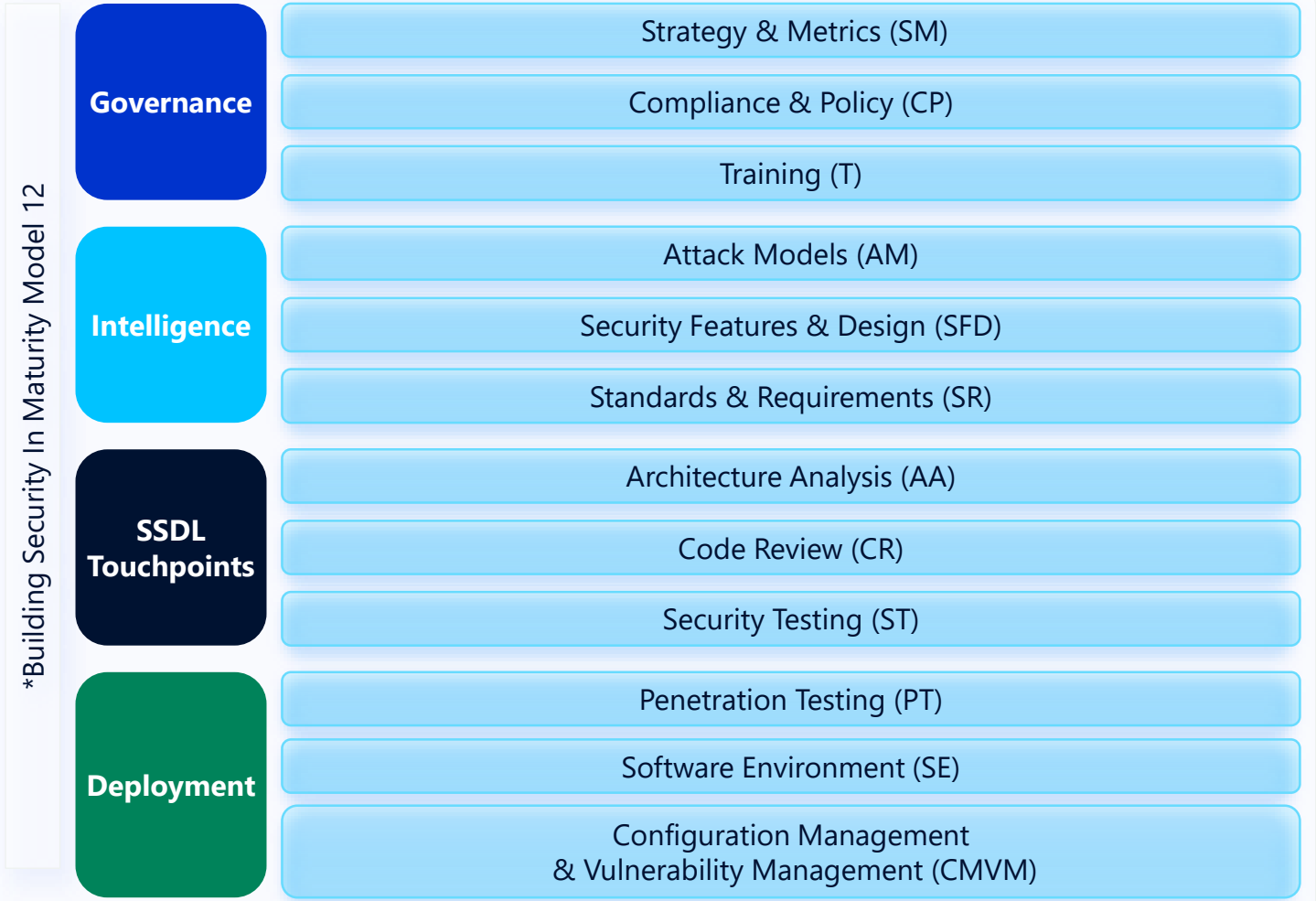
Требования к реализации процесса безопасной разработки

Кем и в какие сроки это должно быть сделано?

Описание функциональных ролей, задач, которые они решают в рамках процесса безопасной разработки, а также схемы бизнес-процессов

ПРЕДЪЯВЛЯЕМЫЕ ТРЕБОВАНИЯ

BEST PRACTICES, BSIMM*



122 — общее количество практик

https://www.bsimm.com

BEST PRACTICES, SAMM*

*OWASP Software Assurance Maturity Model

Governance	Strategy & Metrics	Policy & Compliance	Education & Guidance
Design	Threat Assessment	Security Requirements	Security Architecture
Implementation	Secure Build	Secure Deployment	Defect Management
Verification	Architecture Assessment	Requirements-driven Testing	Security Testing
Operations	Incident Management	Environment Management	Operational Management

90 — общее количество практик

https://owaspsamm.org

БЛОКИ АКТИВНОСТЕЙ

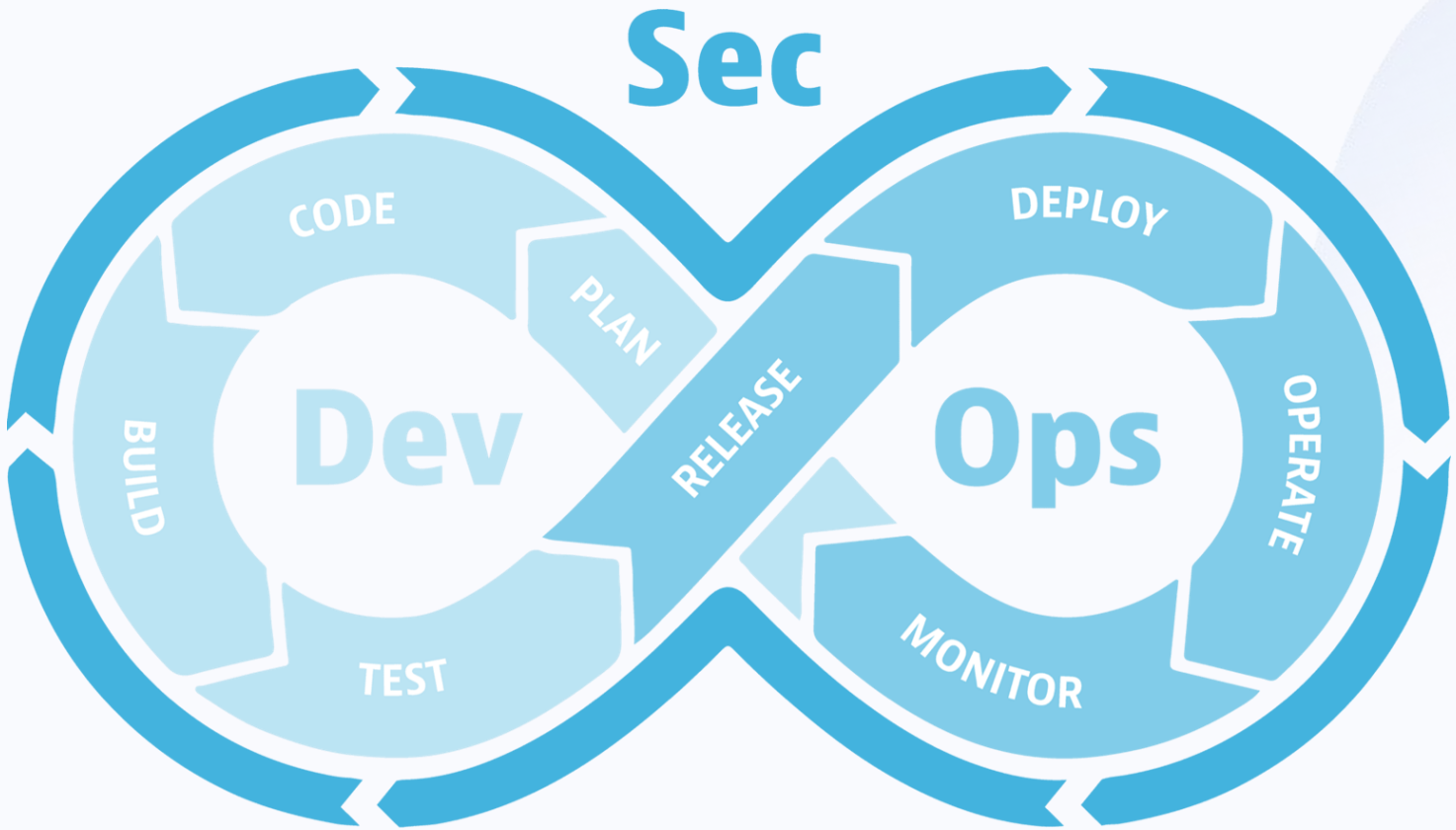
Методология

Ресурсы
и зависимости

Автоматизация

Оценка
эффективности

ЗАВИСИМОСТИ ПРОЦЕССОВ В КОМПАНИИ



ЗАВИСИМОСТИ ПРОЦЕССОВ В КОМПАНИИ

ФУНКЦИОНАЛЬНЫЕ РОЛИ

Бизнес-заказчик

лицо, ответственное за авторизацию выпуска ПО в промышленную эксплуатацию и за соответствие ПО всем функциональным требованиям.

ИТ-архитектор

лицо, ответственное за формирование требований к ИТ-архитектуре, необходимой для разрабатываемого ПО.

Команда разработки

уполномоченные работники компании, отвечающие за разработку ПО, а также подготовку релизов.

Специалист ИБ

лицо, ответственное за формирование требований ИБ к разрабатываемому ПО, проведение функционального тестирования ИБ, анализ безопасности исходного кода, а также контроль выполнения мероприятий по ИБ.

Команда тестирования

уполномоченные работники компании, ответственные за анализ качества разрабатываемого ПО.

Команда эксплуатации

уполномоченные работники компании, ответственные за мониторинг работоспособности ПО.

SECURITY CHAMPION – «ИНТЕРФЕЙС» ИБ В МИРЕ РАЗРАБОТЧИКОВ. И НАОБОРОТ

- **Security Champion – разработчик из команды**, заинтересованный в безопасности
- **Разработчик находится «ближе к продукту», знает его особенности, архитектуру и принципы работы**, что позволяет более эффективно выбирать способы реализации требований по ИБ
- Security Champion выступает в качестве **«интерфейса» взаимодействия с представителями ИБ** по всем вопросам, касающимся информационной безопасности
- **Security Champion объясняет и показывает, как пользоваться инструментами по ИБ**, внедренными в pipeline разработки
- Может проводить **Code Review с точки зрения ИБ**, оказывать **консультации** команде разработки **по вопросам обеспечения ИБ**

RACI-МАТРИЦА

R	**responsible** ответственный за работу	тот, кто непосредственно выполняет задание
A	**accountable** ответственный за результат	тот, кто принимает работу и несёт ответственность за результат
C	**consulted** консультирующий	тот, кто оказывает консультационную помощь
I	**informed** информируемый	тот, кто в курсе принимаемых решений и хода выполнения задачи

RACI-МАТРИЦА (ПРИМЕР)

						
Формирование требований к разрабатываемому ПО	**RA**	**R**	**R**	**I**	**-**	**-**
Разработка ПО	**I**	**C**	**I**	**RA**	**-**	**-**
Устранение уязвимостей, выявленных в ходе статического анализа кода	**I**	**-**	**C**	**RA**	**-**	**-**
Проведение тестирования на проникновение	**I**	**-**	**RA**	**I**	**-**	**I**
Проведение нагрузочного тестирования	**I**	**-**	**-**	**I**	**RA**	**I**
Мониторинг функционирования приложения	**I**	**-**	**I**	**I**	**-**	**RA**

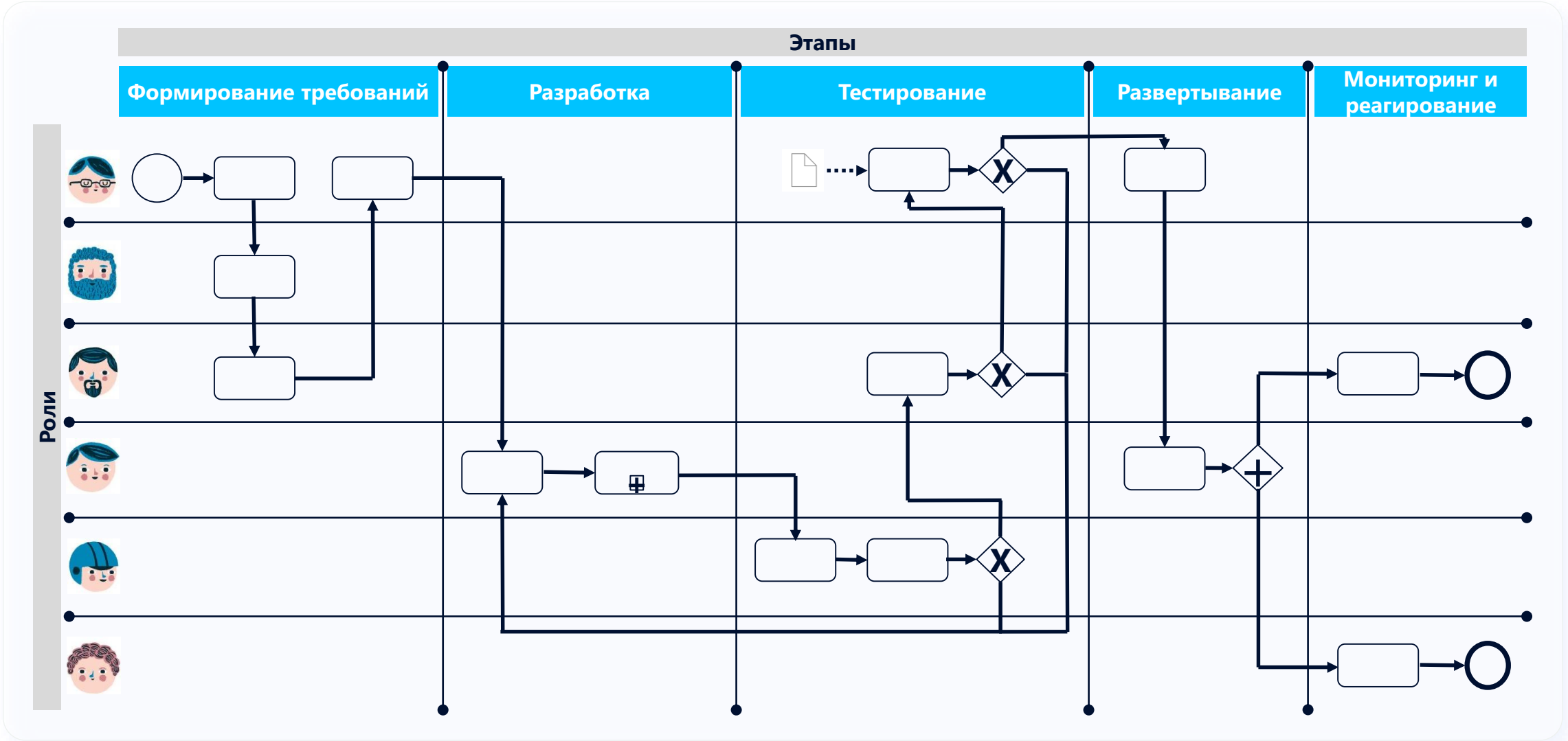
Responsible
выполняет задачу

Accountable
отвечает за результат

Consulted
консультирующий

Informed
информируемый

ПРИМЕР БИЗНЕС-ПРОЦЕССА



БЛОКИ АКТИВНОСТЕЙ

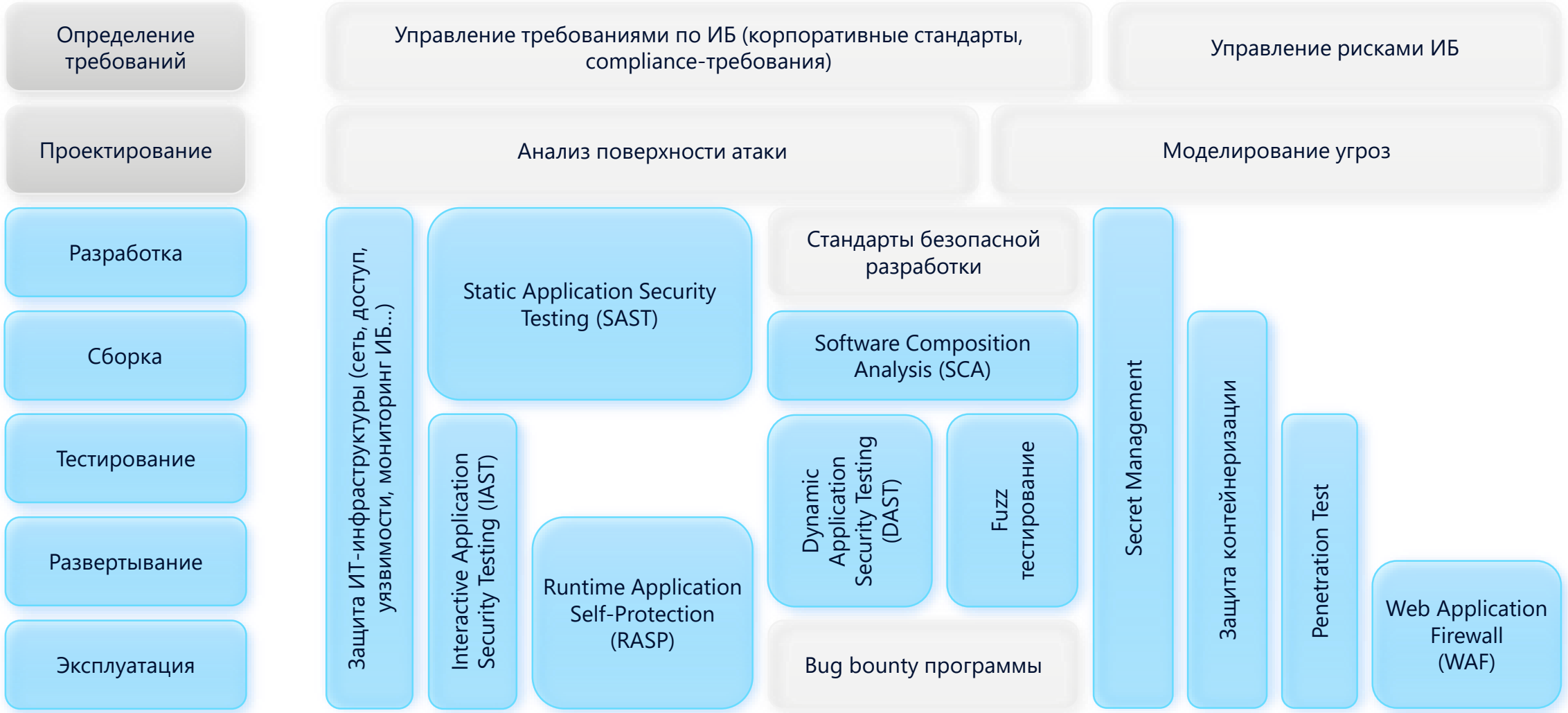
Методология

Ресурсы
и зависимости

Автоматизация

Оценка
эффективности

ЧТО МОЖНО АВТОМАТИЗИРОВАТЬ



АВТОМАТИЗАЦИИ БЫТЬ

Определение ответственных

Подготовка инструкции по эксплуатации

Размещение материалов на доступном для команд ресурсе

Описание бизнес-процесса

Проведение обучения

НА ЧТО ОБРАЩАТЬ ВНИМАНИЕ ПРИ ВЫБОРЕ ИНСТРУМЕНТОВ*?

Не ломайте сборки!
Хотя бы до тех пор, пока не встроите решение корректно и не выстроите вокруг него необходимый процесс

Одно решение – один job!
Это упростит debug, возможна реализация параллельных сканирований

API & CLI = <3
Отсутствие API и/или CLI затрудняет или делает практически невозможным встраивание решения в pipeline

Время – деньги!
Время осуществления проверки не должно превышать 15 минут

Containerize it!
Решения, реализованные в виде контейнера, в значительной степени упрощают возможность использования

Помни о лицензиях!
Выбирайте решения, обладающие минимальными лицензионными ограничениями (например, на параллельное выполнение задач)

Parse everything!
Решения должны предоставлять результат в легко «разбираемом» (parsing) формате, например, xml и/или json

Управление false-positive!
Должна быть реализована возможность подсчета/контроля false positive / false negative

Меньше плагинов!
Потенциальная сложность с масштабированием, недоступность сервиса при обновлении CI/CD и/или самого плагина

Больше информации!
Информируйте разработчиков о причине проблем (например, небольшие сообщения или ссылки на Confluence)

* Согласно материалам презентации «DevSecOps What, Why and How»: https://i.blackhat.com/asia-19/Thu-March-28/bh-asia-Shrivastava-DevSecOps.pdf

БЛОКИ АКТИВНОСТЕЙ

Методология

Ресурсы
и зависимости

Автоматизация

Оценка
эффективности

ЖИЗНЕННЫЙ ЦИКЛ МЕТРИКИ

ПРИМЕРЫ МЕТРИК

Для команд разработки (в общем)		Для конкретной команды разработки
Бизнес-метрики	Метрики процесса	Метрики процесса
• % реализации Программы повышения безопасности приложений • % покрытия критичных приложений Компании проверками по ИБ	• % команд, которые используют решения по анализу безопасности ПО • ТОП-5 наиболее часто встречающихся дефектов ИБ по всем командам • среднее время нахождения дефекта по ИБ в системе управления задачами	• скорость устранения выявленных дефектов по ИБ в программном коде • % false-positive, генерируемых средствами автоматизации проверок ИБ • количество дефектов ИБ на тысячу строк кода (KSLOC)

КОНЦЕПТУАЛЬНЫЙ ПЛАН

	Методология	Ресурсы и зависимости	Автоматизация	Оценка эффективности
1	Обследование			
	Определение ролей, участвующих в процессе, и их задач			
	Разработка нормативной документации Определение набора базовых требований ИБ	Определение взаимозависимых процессов	Определение необходимого стека технологий	Определение необходимых к сбору метрик
	Описание бизнес-процессов			
2				
3				

КОНЦЕПТУАЛЬНЫЙ ПЛАН

	Методология	Ресурсы и зависимости	Автоматизация	Оценка эффективности
1				
2	Разработка технических стандартов	Выделение в команде роли Security Champion	Внедрение инструментов безопасной разработки	Описание собираемых метрик
			Сбор и анализ метрик	
		Проведение обучения для участников процесса		Оценка эффективности реализуемых мероприятий
3	Создание информационного портала, содержащего всю актуальную информацию о процессе безопасной разработки			

КОНЦЕПТУАЛЬНЫЙ ПЛАН

	Методология	Ресурсы и зависимости	Автоматизация	Оценка эффективности
1				
2				
3	Разработка единого фреймворка проведения аудитов, касающихся безопасной разработки	Все участники процесса несут ответственность за безопасность решения	Консолидация отчетности Увеличение покрытия команд инструментами Внедрение дополнительных инструментов ИБ	Решения принимаются на основании собираемых метрик

КОНЦЕПТУАЛЬНЫЙ ПЛАН

	Методология	Ресурсы и зависимости	Автоматизация	Оценка эффективности
1	Обследование			
	Определение ролей, участвующих в процессе, и их задач			
	Разработка нормативной документации Определение набора базовых требований ИБ	Определение взаимозависимых процессов	Определение необходимого стека технологий	Определение необходимых к сбору метрик
	Описание бизнес-процессов			
2	Разработка технических стандартов	Выделение в команде роли Security Champion	Внедрение инструментов безопасной разработки	Описание собираемых метрик
			Сбор и анализ метрик	
		Проведение обучения для участников процесса		Оценка эффективности реализуемых мероприятий
	Создание информационного портала, содержащего всю актуальную информацию о процессе безопасной разработки			
3	Разработка единого фреймворка проведения аудитов, касающихся безопасной разработки	Все участники процесса несут ответственность за безопасность решения	Консолидация отчетности Увеличение покрытия команд инструментами Внедрение дополнительных инструментов ИБ	Решения принимаются на основании собираемых метрик

Актуализация

ВЫВОДЫ

1. Безопасность должна выступать в роли союзника для участников процесса на каждом этапе жизненного цикла ПО, а не в качестве надзирателя
2. Общая цель - сделать продукт не только функциональным, но и безопасным
3. Внедрить все и сразу не получится, нужно подходить к этому поэтапно, развивая каждую активность

ЗАДАВАЙТЕ ВОПРОСЫ

Новопольцева Алина

Инфосистемы Джет

email AA.Novopoltseva@jet.su

2022