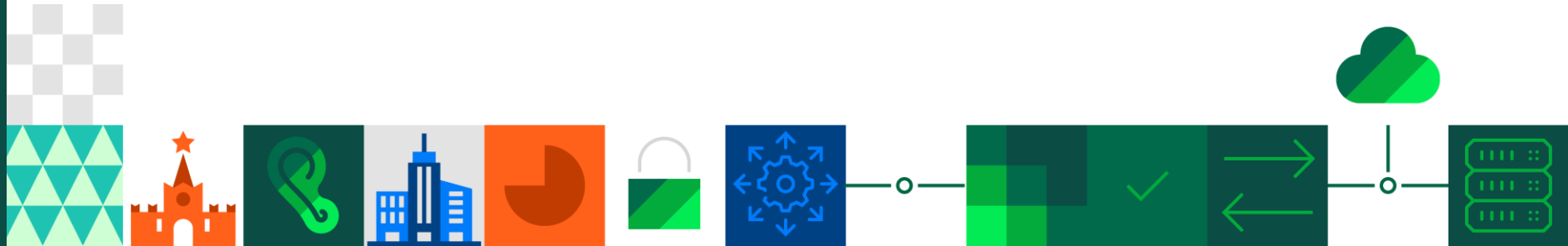
КОД безопасности

Безопасная разработка средств сетевой безопасности

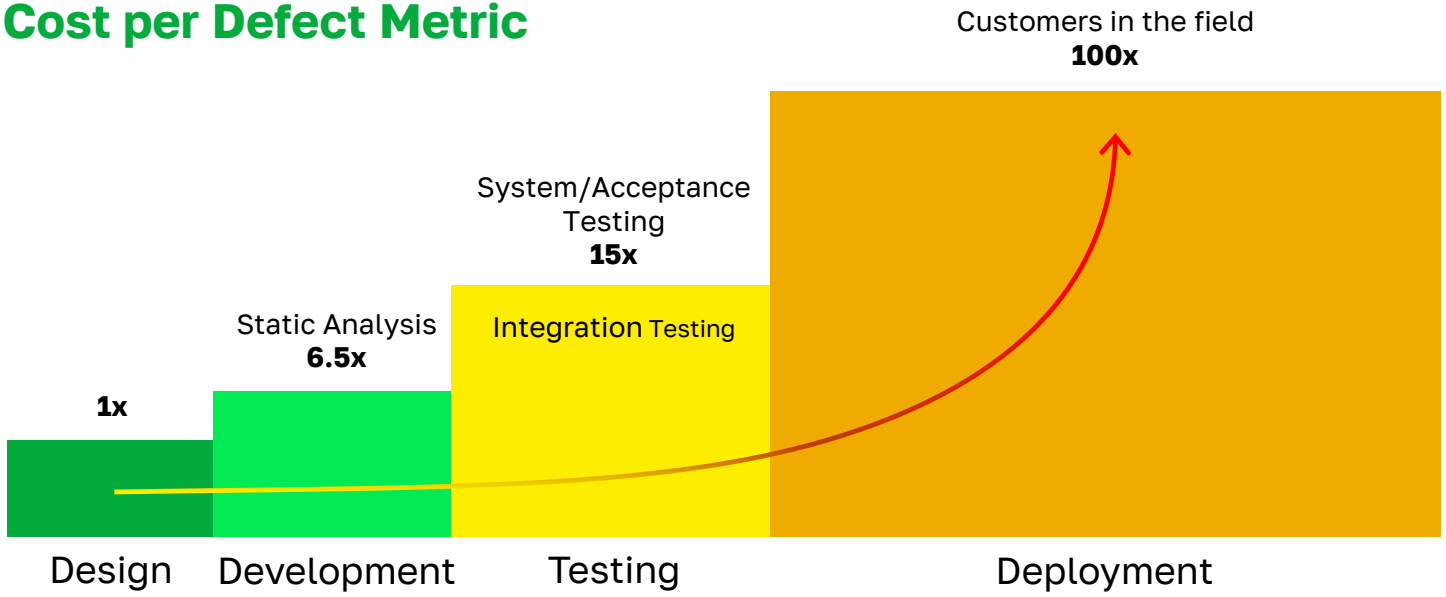


Опасения связанные с уровнями доверия

Это бессмысленно! **Это долго!** **Это дорого!**

Как растет стоимость бага на различных этапах жизненного цикла ПО

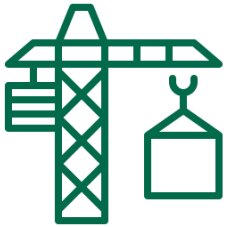
Cost per Defect Metric



Source:
IBM, 2009

Ключевые этапы цикла безопасной разработки

Подготовка



Разработка

Поддержка

Ключевые компоненты КБ SDL

Подготовка

- Компетентные сотрудники
- Безопасная инфраструктура

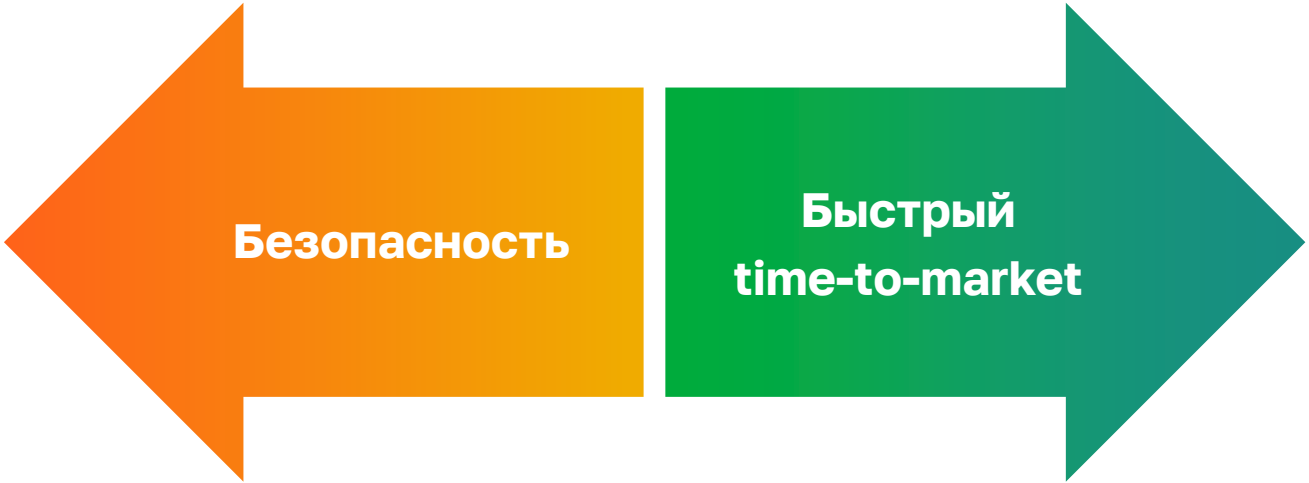
Разработка

- Управление требованиями
- Ответственное проектирование
- Безопасная реализация
- Независимая верификация ИБ

Поддержка

- Поддержка
- Устранение уязвимостей

Парадокс разработчика СЗИ



Решение:

- Короткие релизы
- Автоматизация
- Интеграция практик безопасной разработки в жизненный цикл продукта

Безопасная реализация - процессы

- Трекинг требований и задач
- Защищенное хранение кода
- Процедуры Code review
- Автоматизированный статический анализ кода
- Continuous Integration
- Автоматизированное Unit тестирование
- Функциональное тестирование
  (ручное и автоматизированное)

Ключевые проблемы: Кадровый вопрос

Разработчик для средств ИБ с навыками SDL – птица редкая

Изменения = стресс

Ключевые проблемы: Open Source

Open Source – рабочий способ выпускать новые релизы в разумные сроки

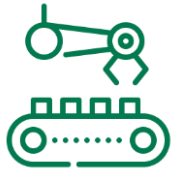
Полноценная замена невозможна

Единоличный контроль безопасности таких компонентов - тоже

Ключевые проблемы: Инструментарий

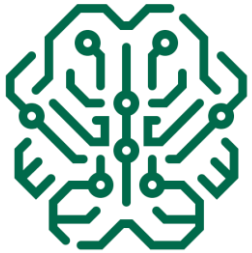
Исходный материал для анализа иногда слишком сложен

Набор используемых языков для современных программных продуктов часто, значительно шире поддерживаемого



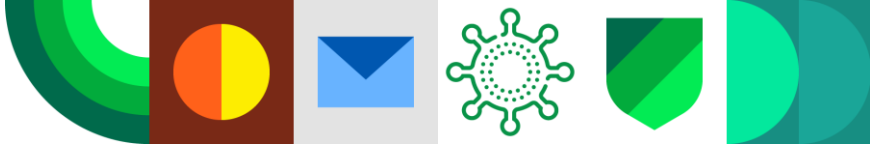
Промышленное применение инструмента – повышенные требования к качеству

Перспектива



Artificial Intelligence?

КОД безопасности



Спасибо за внимание!
Вопросы?

Денис Копылов
D.kopylov@securitycode.ru