# Безопасная разработка при ограниченном бюджете на DevSecOps

RTM GROUP

Объединяем IT,
право и безопасность

Далее

# ЭКСПЕРТЫ

## Фёдор Музалевский

- Директор технического департамента RTM Group
- Экспертная работа с 2010 года. Педагогический стаж с 2012 года
- Кандидат физико-математических наук.
- Участник ТК №122 при Банке России.

## Артём Семёнов

- Консультант по информационной безопасности RTM Group
- Специалист по тестированию на проникновение и анализу защищённости

# О КОМПАНИИ RTM GROUP

**RTM Group** — ведущая консалтинговая компания в области информационной безопасности, судебной экспертизы и ИТ-права.

**Информационная безопасность:**

Аудиты, пентест, КИИ, защита ПДн, оценка защищённости и др.

**Компьютерно-технические экспертизы:**

44-ФЗ, расследования инцидентов, ИТ-контракты и др.

**ИТ-право:**

Судебное и досудебное урегулирование, сопровождение и др.

# О ЧЕМ ПОЙДЕТ РЕЧЬ

01 **Что такое конвейер.**

02 **Конвейер в** DevSecOps.

03 **Контроль версии** и его значимость на этапе разработки ПО.

04 **Как сократить риски** которые могут быть в зависимостях?

05 **Как тестировать приложение** на наличие уязвимостей?

06 **Безопасность на этапе** развёртывания приложения.

# ВОПРОСЫ

Задавайте вопросы на вкладке «**Вопросы**»,

мы обязательно ответим на них.

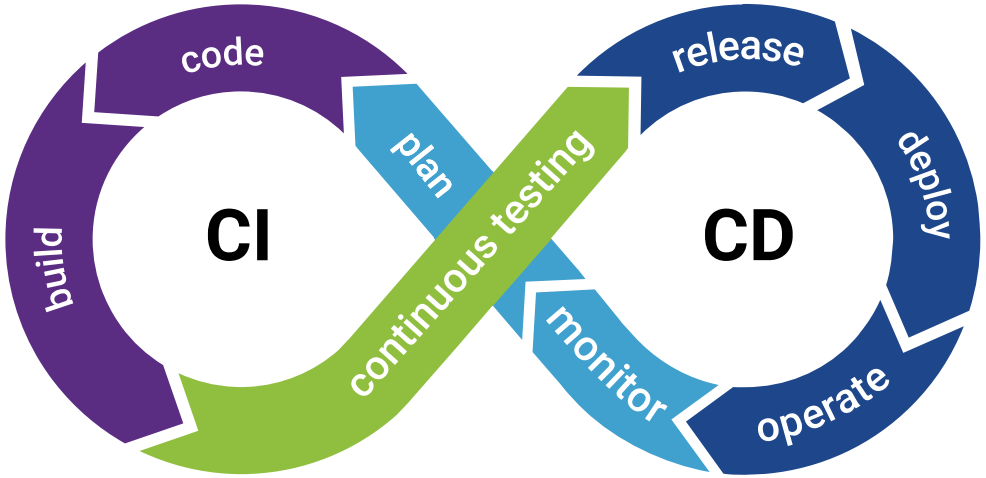
**Самым активным – брендированные подарки!**

# А что такое конвейер?

В современных конвейерах программное обеспечение не перемещается вручную между различными средами.

Вместо этого можно следовать автоматизированному процессу компиляции, сборки, интеграции и развертывания новых функций программного обеспечения.

Этот процесс называется CI/CD.



# А что такое конвейер?

**Мы можем создать так называемый конвейер CI/CD.**
**Такие конвейеры обычно состоят из следующих отдельных элементов:**

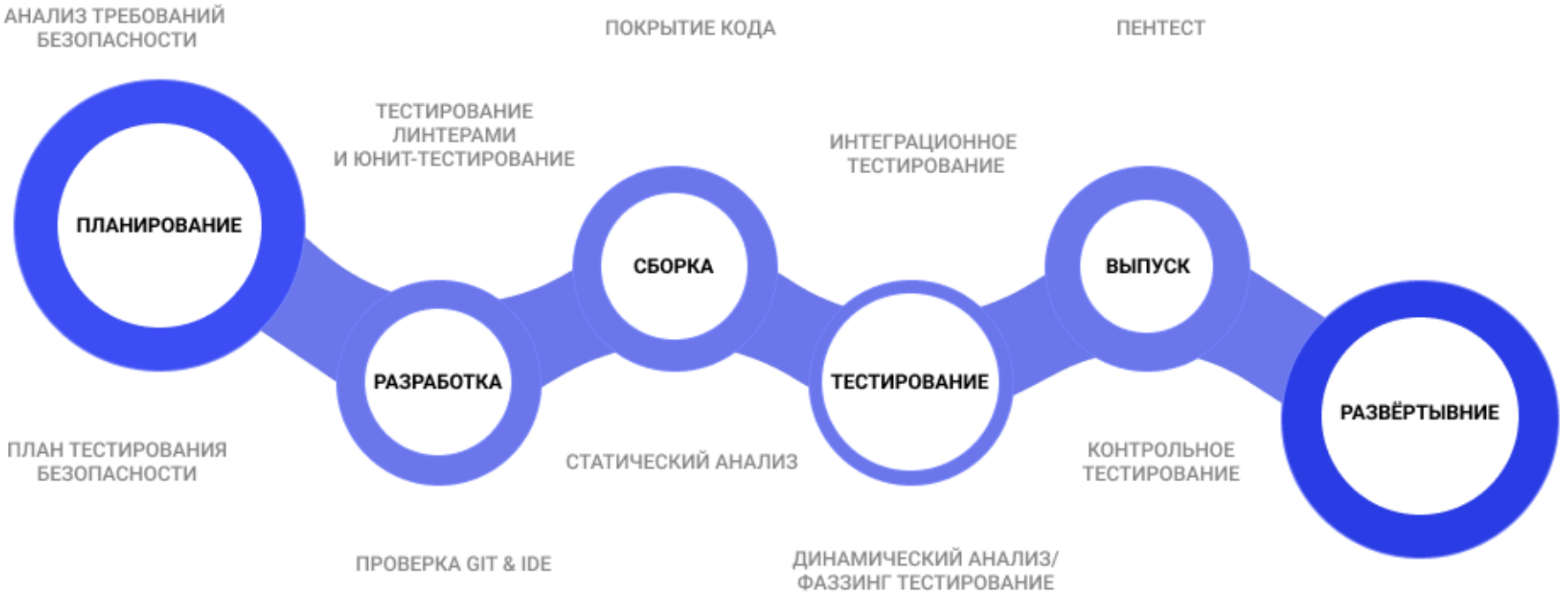
| | |
|---|---|
| **Первоначальное действие** | Запускает процесс конвейера. Например, делается push-запрос на определенную ветку |
| **Действия по созданию** | Они предпринимаются для создания проекта и новой функции |
| **Этап тестирования** | Действия, когда тестируют проект, чтобы убедиться, что новая функция не вмешивается в текущие функции приложения |
| **Этап развёртывания** | Если конвейер работает успешно, действия по развертыванию подробно описывают, что должно произойти со сборкой. Например, она должна быть перенесена в среду тестирования |
| **Этап выпуска в продакшен** | По мере развития процессов CI/CD основное внимание теперь уделяется не только самому развертыванию, но и всем аспектам доставки решения. Сюда входят такие действия, как мониторинг развернутого решения |

Для выполнения действий этих элементов CI/CD конвейерам требуется инфраструктура сборки.
Мы называем эту инфраструктуру для оркестрации и агентами сборки. Оркестратор сборки направляет различные агенты для выполнения действий CI/CD-конвейеров по мере необходимости.

# Понимание конвейера DevSecOps

| | |
|---|---|
| **Конвейер DevSecOps** | Это автоматизированный подход, который позволяет предприятиям создавать безопасное программное обеспечение на всех этапах разработки, тестирования и развертывания. Интегрируя безопасность, предприятия могут минимизировать поверхность атаки своего программного обеспечения, чтобы снизить риск эксплуатации со стороны киберпреступников и хакеров. |
| **Цель внедрения конвейера DevSecOps** | Обеспечить обнаружение и устранение лазеек в системе безопасности до развертывания программного обеспечения и свести к минимуму возможность нанесения ущерба инфраструктуре, данным или пользователям. |

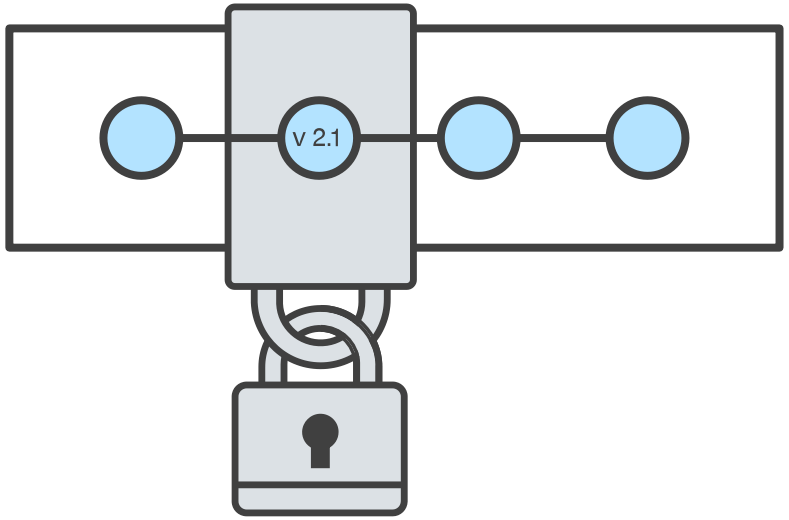
# Понимание конвейера DevSecOps




Типичный конвейер DevSecOps состоит из нескольких этапов, как и стандартный процесс SDLC, который включает такие этапы, как планирование, кодирование, создание, тестирование, выпуск и развертывание. Каждый этап процесса DevSecOps имеет свой собственный набор проверок безопасности.

# Контроль версии и почему он важен на этапе разработки ПО?

**Мы часто интегрируем новые функции в наше программное обеспечение**. Современные подходы к разработке, такие как Agile, означают, что мы постоянно обновляем наш код. Чтобы держать все эти обновления под контролем, нам нужен контроль версий. Над кодом работает вся команда разработчиков, а не только один разработчик. Чтобы обеспечить интеграцию изменений от нескольких разработчиков, необходим контроль версий.




Контроль версий позволяет нам хранить несколько версий кода. Это может быть конкретная версия, над которой работает каждый разработчик, но это могут быть и совершенно разные версии нашего приложения, включая второстепенные и главные версии.

# Контроль версии и почему он важен на этапе разработки ПО?

**Две наиболее распространенные системы хранения исходного кода и контроля версий — это Git и SubVersion** (SVN).

**Git** – это распределенный инструмент контроля исходного кода, что означает, что у каждого участника будет своя копия исходного кода.

С другой стороны, **SVN** – это централизованный инструмент контроля исходного кода, то есть управление репозиторием осуществляется централизованно.

***Часто случается так, что разработчик случайно фиксирует в репозитории Git секреты, такие как учетные данные или строки подключения к базе данных. Осознав свою ошибку, они удаляют секреты и создают новый коммит. Однако в репозитории теперь будут оба коммита. Если злоумышленник получит доступ к репозиторию, он может воспользоваться таким инструментом, как GittyLeaks***, который просканирует коммиты в поисках конфиденциальной информации. Даже если эта информация больше не существует в текущей версии, эти инструменты могут просканировать все предыдущие версии и раскрыть эти секреты.

# Как сократить риски, которые находятся в коде?

1. **Внутренние зависимости**
2. **Внешние зависимости**

**Соображения безопасности**

Основная проблема безопасности заключается в том, что зависимости — это код вне нашего контроля. Особенно в наше время, когда используется так много различных зависимостей, невероятно сложно отслеживать зависимости. Если в этих зависимостях есть уязвимости, это может привести к уязвимостям в нашем приложении.

# Как сократить риски, которые находятся в коде?

**Пример из реальной жизни**

Уязвимость 0day была обнаружена в зависимости Log4j в 2021 году под названием Log4Shell.

Log4j — это утилита протоколирования на базе Java. Она является частью Apache Logging Services, проекта Apache Software Foundation.

Уязвимость может позволить не аутентифицированному злоумышленнику получить удаленное выполнение кода на системе, использующей эту зависимость.

# Как тестировать приложение на наличие уязвимостей.

- **Юнит-тестирование**
- **Интеграционное тестирование**
- **SAST**
- **DAST**
- **RASP**
- **IAST**
- **Пентест**

**Инструменты, которые можно использовать на этом этапе:**

Существует несколько общих инструментов, которые можно использовать для автоматизированного тестирования. **GitHub** и **Gitlab** имеют встроенные инструменты SAST. Такие инструменты, как **Snyk** и **Semgrep**, также популярны для SAST и DAST.

# Безопасность на этапе развёртывания приложения.

# ГОТОВЫ ОТВЕТИТЬ НА ВАШИ ВОПРОСЫ!

**Музалевский Фёдор**

Директор технического департамента

**Семенов Артём**

Консультант по Информационной безопасности


+7 (495) 197-64-95


info@rtmtech.ru


https://rtmtech.ru


rtm.group

# Будьте в курсе последних новостей ИТ, ИБ и права!

Подписывайтесь на наш VK, канал YouTube и заходите на сайт.
**Узнавайте о важнейших событиях мира ИБ и ИТ первыми!**

**Rtm.group**

**Rtmtech.ru**

**RTM Group**