SOLAR Rostelecom

Ростелеком

Как сделать разработку безопасной, сохранив силы и нервы

МОСКВА Январь, 2019

Теория vs Практика

- Microsoft выпустила методологию Secure SDLC еще в 2004м году, с достаточно детальной регламентацией
- До первого проекта нам казалось что мы все четко знаем и понимаем, пока не начали внедрять процессы
- Самое сложное – не техническая, а организационная часть
- На практике всплывает много интересных моментов, которые могут обнулить эффект от внедренных процессов

№1. Согласовать все со всеми

- Большое количество участников: разработчики, отделы управления изменениями и релизами, менеджеры проектов, безопасники, методологи, владельцы систем.

- Обычно у всех свое мнение и свое видение как должно быть. Учесть все невозможно

- Важно:
- 1. Уметь объяснять всем участникам, как оптимальная версия процесса будет либо помогать конкретному участнику, либо хотя бы не мешать
- 2. Иметь заранее согласованный с вышестоящим руководством план принятия решений, в случае, если не удалось договориться с кем-то из участников процесса

№2. Найти ресурсы внутри компании

- Необходимо вовлечение сотрудников в процесс, но большинству участников он просто мешает
- Сразу возникает вопрос отсутствия ресурсов – в первую очередь времени и сотрудников

- Важно:
- Изначально иметь в запасе бюджет на дополнительные ставки
- Иметь ввиду, что найм новых участников процесса, например, менеджера по безопасной разработке может сильно затянуться

№3. Ожидания скорости работы процесса

- Разработчики привыкли, что компиляция идет быстро, и ждут что статический анализатор тоже должен работать быстро и хотят, чтобы он запускался на каждый чих
- Но быстро анализ не пройдет, проанализировать только код разработчика нельзя - ведь статический анализатор строит трассы, а они могут проходить и вне кода этого разработчика

- Важно:
- либо находить компромисс, все это объясняя, либо выключать разработчиков из процесса сканирований, передавая им уже обработанные результаты. Иметь ввиду, что найм новых участников процесса, например, менеджера по безопасной разработке может сильно затянуться

№4. Учесть изменения в инфраструктуре компании

- Во всех крупных компаниях периодически проходит улучшение и рефакторинг инфраструктуры - CI/CD, репозиториев, процессов, с этим связанных и т.п.
- Проект по внедрению Secure SDLC может идти довольно долго, за это время может меняться инфраструктура - и тут приходится подстраиваться под эти изменения
- Важно:
- Учитывать запланированные изменения инфраструктуры компании до начала проекта по внедрению процесса

№5. Угадать требования по железу

- Как правило клиент просит расчет по железу заранее, передавая разные по адекватности оценки объема кодовых баз
- Технологии глубокого статического анализа приводят к тому, что оценку по железу можно дать, только просканировав код
- Также на оценку влияет частота проверок, объемы по увеличению кода

Важно:

- Заложить возможность перерасчета требований инфраструктуры

№6. Учесть все существующие регламенты и те, которые еще не разработаны

- Процесс Secure SDLC должен быть формализован в виде регламента
- в компаниях обычно уже есть регламенты: разработки, управления уязвимостями, управления релизами
- Во все эти регламенты надо встроиться, их надо учитывать.
- А еще может быть, что регламент надо написать - а других регламентов еще нет, они в разработке.

Важно:

- При оценке работ на внедрение зафиксировать скоуп имеющихся и планируемых регламентов

№7. Учесть появление новых систем

- Важный момент, который можно упустить при регламентировании процесса проверки кода на уязвимости - это добавление новых систем в процесс.
- Что делать, если в компании начал разрабатываться новый продукт, или компании принимает у подрядчика в поддержку новую кодовую базу?

Важно:

- При проектировании процесса должна быть предусмотрена ситуация, когда новые системы автоматически попадают в процесс

№8. Ожидания по срокам

- Процесс в среднем занимает 6-9 месяцев
- «С места в карьер» или «Новая жизнь с понедельника» не получается.

Важно:

- Внедрять процесс этапами, постепенно наращивая скоуп.

№9. Пропустить релиз в production или отправить на устранение уязвимостей

- На начальных этапах работы процесса будет выявлено большое количество уязвимостей
- Если накладывать вето на релизы, то очень скоро бизнес распорядится отказаться от процесса

Важно:

- Согласовать критерии допуска релиза в production
- Отказаться от наложения вето на релиз безопасниками в первые месяцы

№10. Что еще может пойти не так...

- Железо под проект не приехало вовремя
- Уволился кто-то из ключевых участников проекта в ходе компании, срыв сроков
- Пожелания участников процесса невозможно реализовать текущим функционалом инфраструктурных компонентов
- На поздних этапах внедрения всплывают системы, которые не отражены при инвентаризации
- Никто не хочет брать в поддержку внедренную систему и элементы интеграции

Важно:

- На берегу оговорить возможные риски и план действий.

SOLAR
Rostelecom

Ростелеком

Ростелеком-Solar – надежный партнер государства, бизнеса и населения в обеспечении кибербезопасности

info@rt-solar.ru
+7 (499) 755-07-70