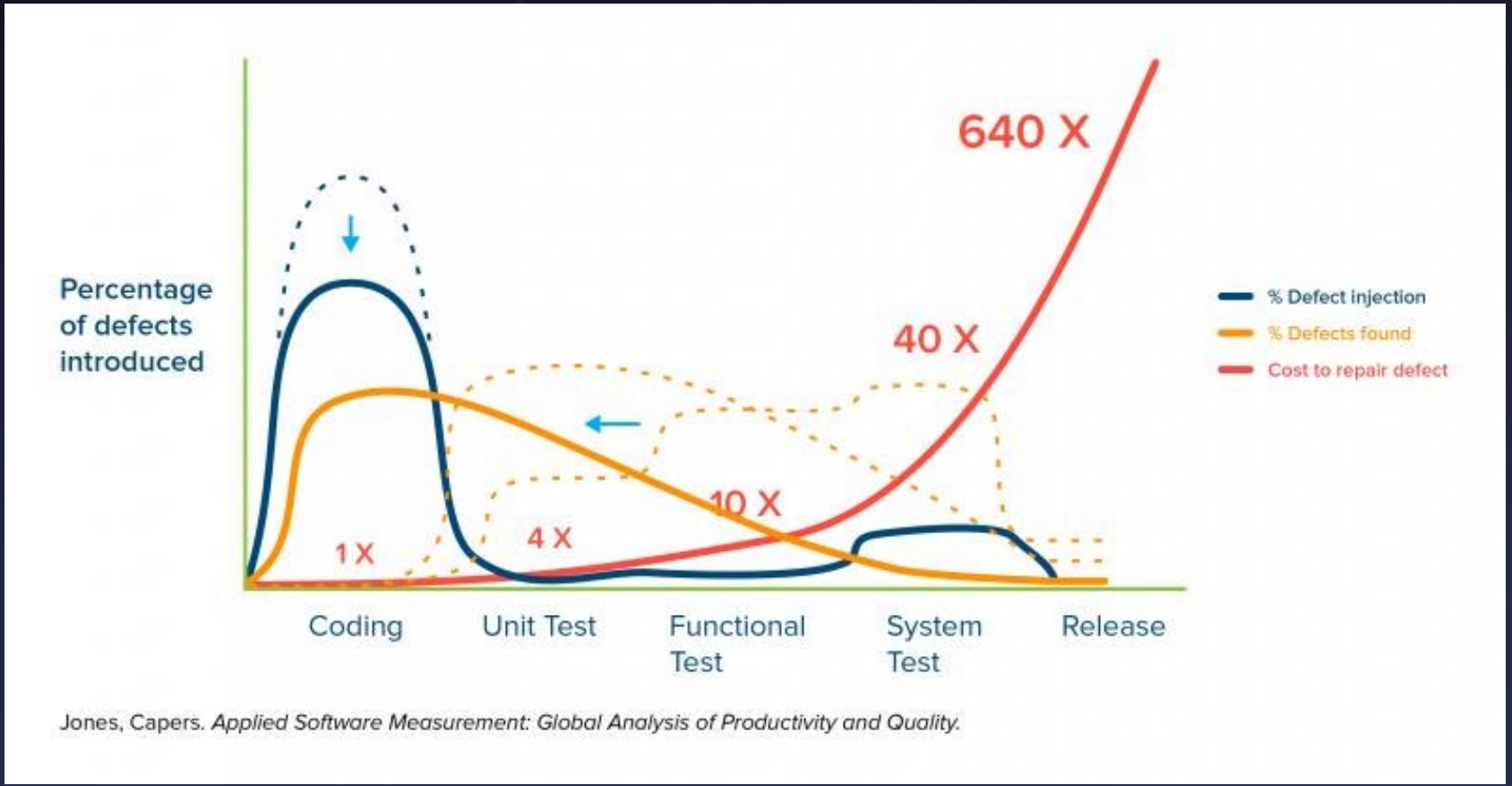
Shift ~~Left~~ Everywhere Security в каждый дом

Дмитрий Евдокимов
Founder, CTO

Обо мне

- Основатель и технический директор Luntry
- Опыт в ИБ более 10 лет
- Соорганизатор конференций ZeroNights, DEFCON Russia (#7812)
- Быв. автор статей и редактор рубрик в журнале “ХАКЕР”
- Автор Telegram-канала “k8s (in)security”
- Автор курса “Cloud Native безопасность в Kubernetes”
- Не верит, что систему можно сделать надежной и безопасной, не понимая ее.
- Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, PHDays, DevOpsConf, KuberConf, VK Kubernetes Conference, HighLoad++ и др.

Shift Left testing

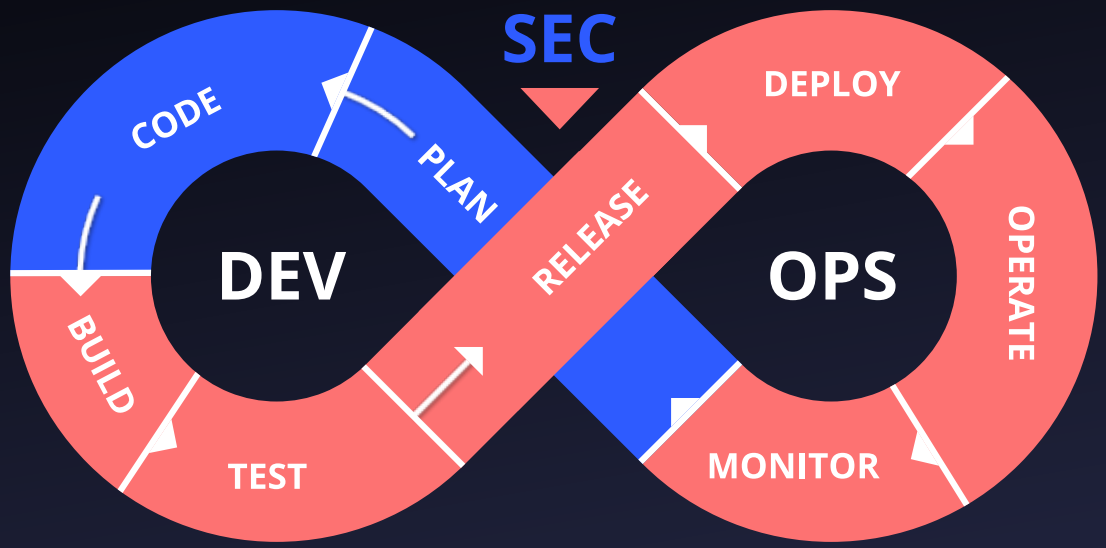


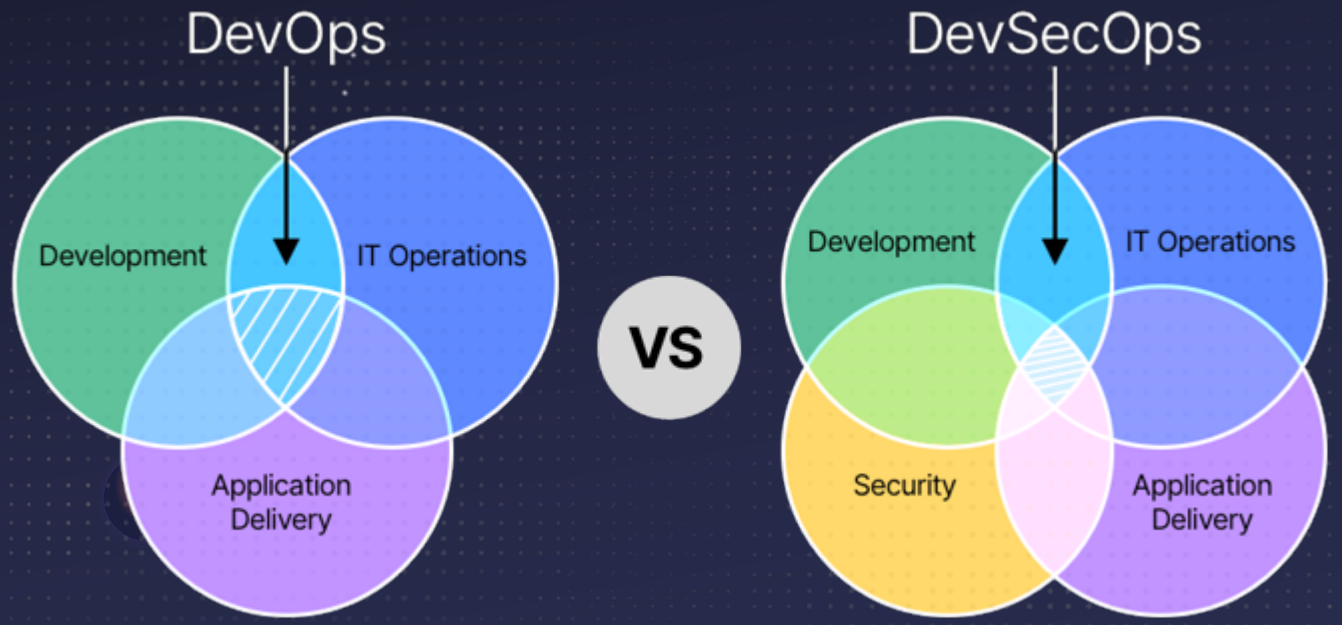
Источник: https://www.stickyminds.com/article/shift-left-approach-software-testing

SecDevSecOpsSec

Ссылка на выступление https://www.youtube.com/watch?v=pfOFwAE5Hwo&ab_channel=DevOpsChannel

Sec


SEC
CODE
PLAN
BUILD
TEST
DEV
RELEASE
DEPLOY
OPERATE
MONITOR
OPS


DevOps
Development
IT Operations
Application Delivery
VS
DevSecOps
Development
IT Operations
Security
Application Delivery

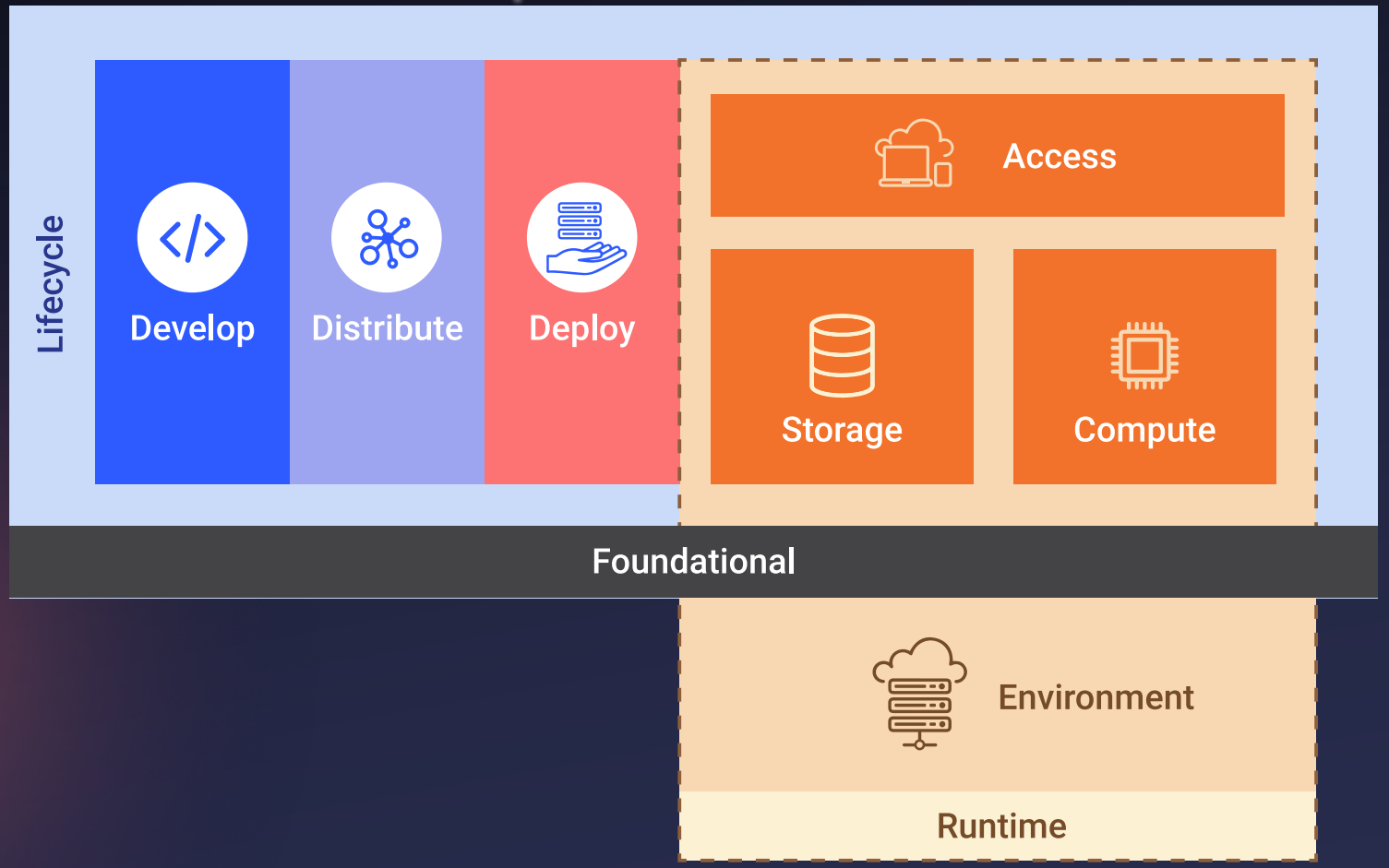
Shift Everywhere

“Shift Left” is becoming “Shift Everywhere.”

- Although shift left has been promoted as doing some security testing during development, that is a large simplification of what we meant. More accurately today, some secure software development lifecycles (SSDLs) seek to conduct an activity as quickly as possible with the highest fidelity as soon as the artifacts on which that activity depend are made available. Sometimes, that’s to the left of where you’re doing things today, but often times, it’s to the right. In addition, technology trends naturally require shifting right to produce rapid and accurate telemetry from modern languages, frameworks, and software orchestration.
- Established practices such as secure code review are leveraging enhanced source code management features to allow review during multiple phases. For example, shift left to initial code commits and shift right to augment metadata offered as part of pull requests sent to repository maintainers when code is finished and tested. These options reflect a desire to present results both where they can be achieved the soonest and where they will be most impactful.
- Some organizations evaluating defect discovery tools and services are showing a growing preference for continuous event-based security telemetry throughout a value stream rather than a single point-in-time analysis.
- Those organizations attempting to maintain accurate software inventory data are discovering the need to align efforts across source code content management, the build process, the deployment process, and the operations environment, where inventory granularity and content will likely be different with each view and will also change frequently. Such organizations are struggling to maintain the effectiveness of their existing inventory efforts while also adapting to new software lifecycles, software architecture changes and any underlying software, deployment, and cloud technologies changes happening in response to the engineering self-service trends and the digital transformation sea change.

Источник: Building Security In Maturity Model (BSIMM) 11

Безопасность Runtime



Модели нарушителей

- Внутренний
- Внешний
- ???

Вместо заключения

- Безопасность Runtime очень важная составляющая безопасности
  - В Runtime идет работа с реальными данными
  - От Runtime двигаются в левую сторону и реализуют Shift Everywhere Security
  - Нужно видеть и понимать что происходит в Runtime
- Помните о моделях нарушителя
  - Влияет на выбор контролей безопасности
  - Сканирования в pipeline это не панацея
  - Безопасность вещь комплексная

Спасибо за внимание!

luntry.ru

Email: de@luntry.ru

Twitter: @evdokimovds

Telegram: @Qu3b3c