

State of DevOps 2022

Содержание

- Trunk-based development
- Continuous Delivery
- SLSA Framework
- Documentation

Trunk-based development

Trunk-based development — это практика непрерывного объединения кода в «trunk» и избегания долгоживущих функциональных ветвей. Эта практика считается дополнением к CI и, как было доказано на протяжении многих лет, ускоряет скорость доставки программного обеспечения.

Trunk-based development

Влияние коммерческого опыта на успех практики

▼ **Decreased** overall software delivery performance

▲ **Increased** amounts of unplanned work

▲ **Increased** error-proneness

▲ **Increased** change failure rate

< 16 лет коммерческого опыта

▲ **Increased** overall software delivery performance

▼ **Decreased** amounts of unplanned work

▼ **Decreased** error-proneness

▼ **Decreased** change failure rate

16+ лет коммерческого опыта

Это из-за дополнительных практик, необходимых для успешного внедрения Trunk-based dev. Однако данный подход показывает оказывает положительное влияние на общую эффективность.

Я бы использовал данный подход на практике, поскольку от числа задач/участников команд растет число веток, что создает сложности в организации работы.

Continuous Delivery

- Команды которые сочетают version control и CI в 2.5 раза чаще достигают высокой производительности доставки программного обеспечения, чем команды, которые сосредоточены только на чем-то одном.

На практике я бы в своих подходах применял больше CI подходов для того чтобы избавиться от типовых ошибок

SLSA framework

SLSA framework - это чеклист требований, которые должны быть учтены в безопасной цепочке поставок (Supply Chain).

Интересно внедрить в практику команд написание безопасного кода в процессе разработки ПО

SLSA Practice	Survey Definition
Centralized CI/CD	Production releases are built by using a centralized CI/CD system, never on a developer’s workstation
History Preserved	Revisions and their change history are preserved indefinitely
Build Script	Builds are fully defined through the build script and nothing else
Isolated	Builds are isolated; they cannot interfere with concurrent or subsequent builds
Build Text Files	Build definitions and configurations are defined in text files stored in a version control system
Parameters Metadata	Build metadata (e.g. dependencies, build process, build environment) about an artifact includes all build parameters
Dependencies Meta-data	Build metadata (e.g. dependencies, build process, build environment) about an artifact documents all dependencies
Metadata Generated	Build metadata (e.g. dependencies, build process, build environment) is either generated by the build service, or by a build-metadata generator that reads the build service
Prevent Inputs	When running builds, build steps are prevented from loading any build inputs dynamically (i.e. all required sources and dependencies are fetched upfront)
Users No Edit	Build metadata (e.g. dependencies, build process, build environment) about an artifact cannot be edited by build services users
Metadata Available	Build metadata (e.g. dependencies, build process, build environment) is available to the people who need it (e.g. via a central database), and is delivered in a format that they accept
Two Person Review	Every change in a revision’s history must be individually reviewed and approved by two trusted persons prior to submission
Metadata Signed	The build metadata (e.g. dependencies, build process, build environment) about how an artifact was produced is signed by my build service

Table 1. SLSA-Related Survey Questions

Documentation

Документация становится все более автоматизированной практикой, особенно среди высокопроизводительных команд.

Подход «хороший ход - документирует сам себя» не всегда работает и часто необходимая задача. Поэтому на практике будет плюсом применять данную концепцию

Documentation practices negatively impacted software delivery performance.

Источник: State of DevOps 2022