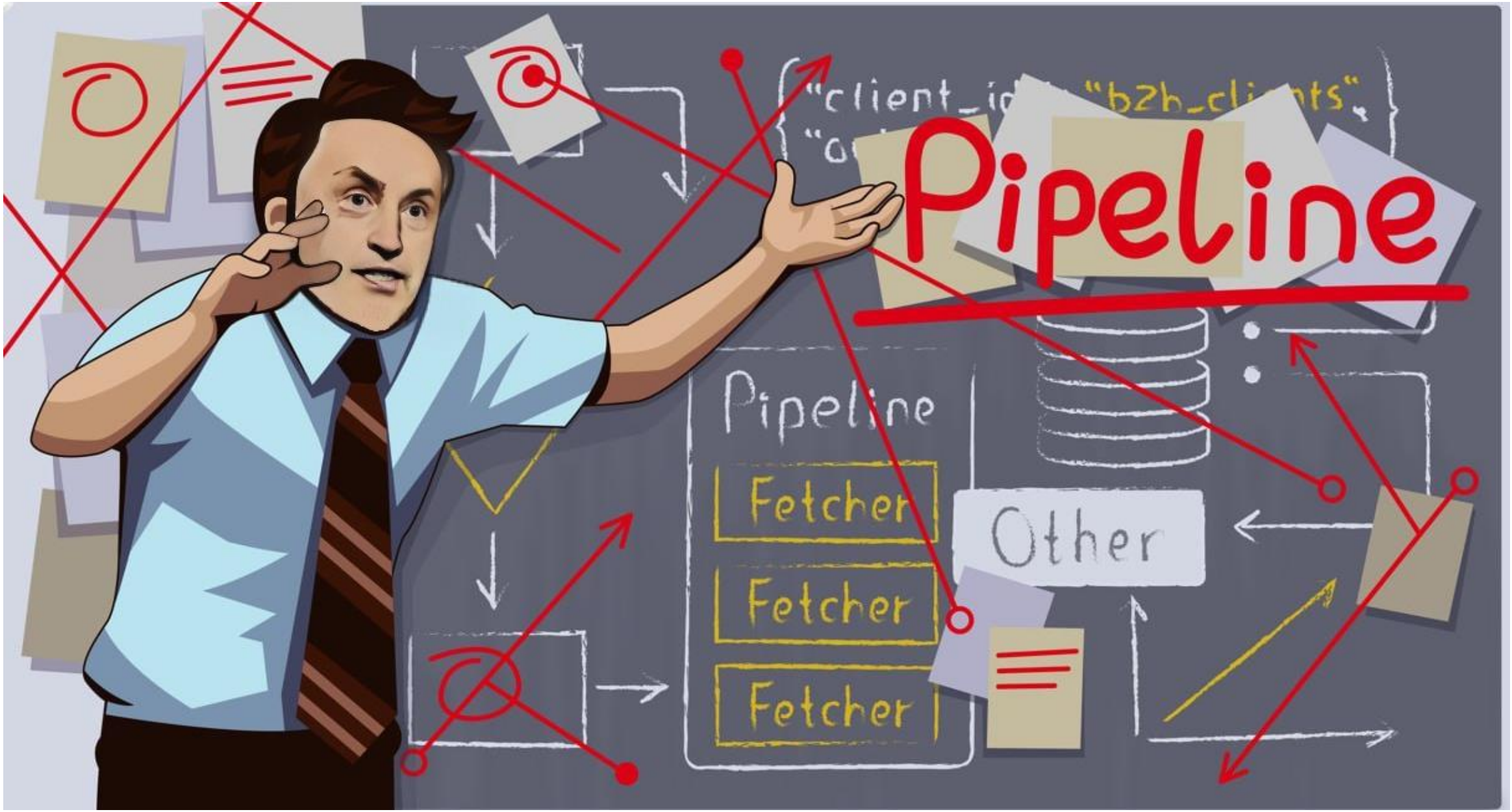
Испытательная лаборатория
ООО Научно-технический центр «Фобос-НТ»

Современная лаборатория, конвейеризация внутренних процессов

Испытательная лаборатория ООО НТЦ «Фобос-НТ» это:

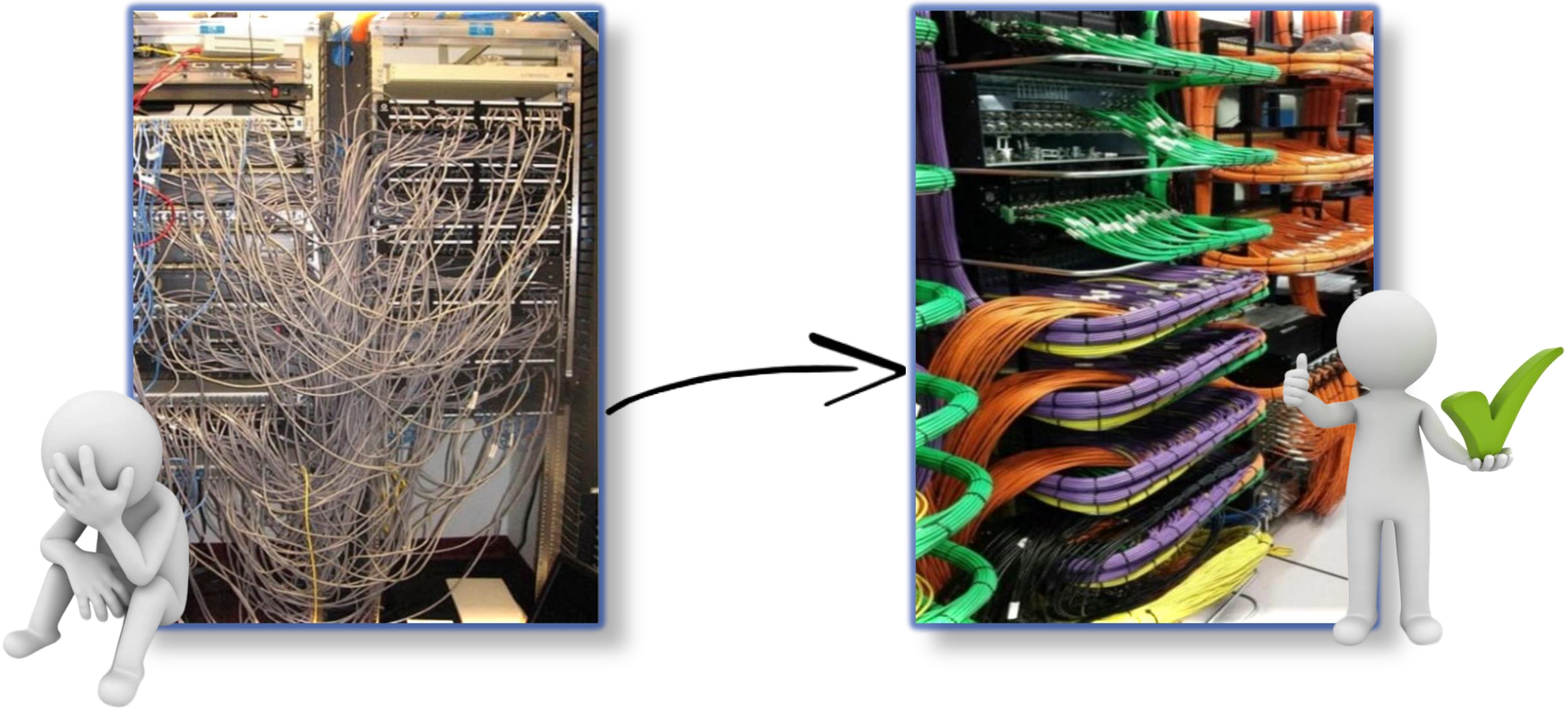
- Первая лаборатория, сотрудники которой прошли аттестацию экспертов;
- Более 20 завершенных сертификаций в 2024 году;
- Активная работа в Центре исследований безопасности системного ПО, совместно с коллегами из компаний Базальт СПО и Базис обработано более сработок статического анализатора 3000


PostgresPro
КОД безопасности
TCC
ареал консалтинг
ИВК
РФЯЦ-ВНИИЭФ РОСАТОМ
ФАСТЕК
FLANT Deckhouse
kaspersky
R-Vision
NGRSOFTLAB
АСКОН
LUNTRY
1830
Dr.WEB®
НАМ ДОВЕРЯЮТ
СОЛАР
Ростелеком
РТК-ЦОД
BASIS
ГАРДА
base alt
UserGate
MTC RED
CODE SCORING
АМИКОН
Аладдин рд
AXIOM JDK
НПЦ КСБ
RASU ROSATOM
ideco
ROSA
SECURITM
КЕЙСИСТЕМС

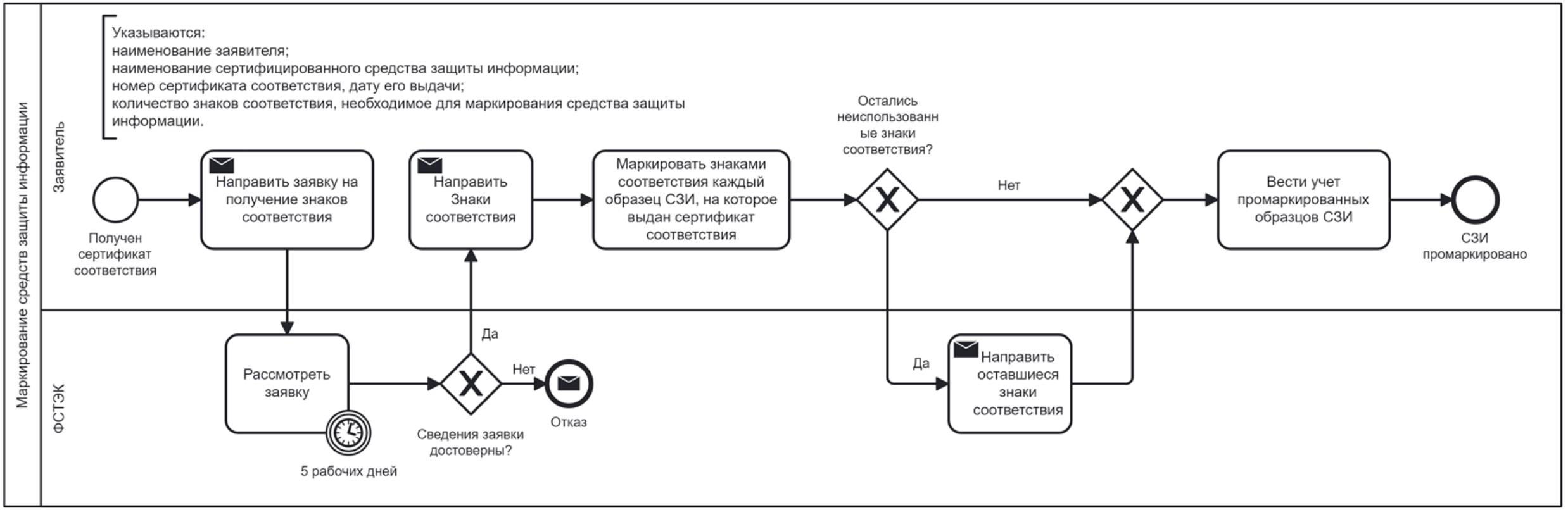

Pipeline
Pipeline
Fetcher
Fetcher
Fetcher
Other

Анализ документации и иных исходных данных

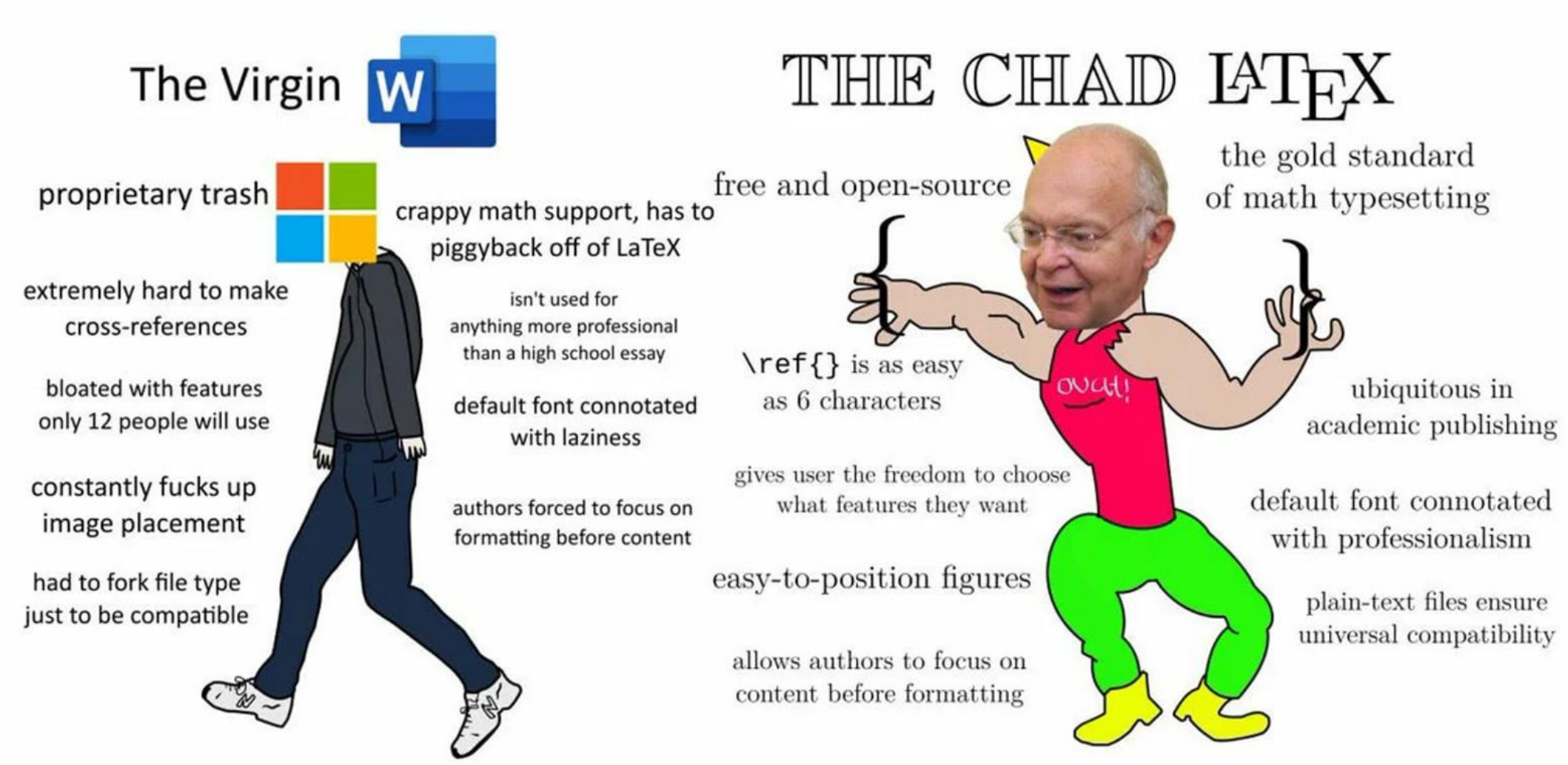
ИСТОРИЯ ШАБЛОНОВ



BPMN
НАЗАД В БУДУЩЕЕ



ДОКУМЕНТАЦИЯ ПО ИМЕНИ
docs as code



Анализ архитектуры, определение поверхности атаки и параметров безопасности

Проблематика

Анализ образов контейнеров (анализ статическими методами)

Поиск уязвимостей в программных пакетах и зависимостях в образе

Trivy (может все, но хотелось бы больше)

```
$ trivy image python:3.10.14-alpine3.20
2024-07-03T10:04:35-04:00       INFO    Vulnerability scanning is enabled
2024-07-03T10:04:35-04:00       INFO    Secret scanning is enabled
2024-07-03T10:04:35-04:00       INFO    If your scanning is slow, please try '--scanners vuln' to disa
2024-07-03T10:04:35-04:00       INFO    Please see also https://aquasecurity.github.io/trivy/v0.52/doc
2024-07-03T10:04:35-04:00       INFO    Detected OS     family="alpine" version="3.20.0"
2024-07-03T10:04:35-04:00       INFO    [alpine] Detecting vulnerabilities...   os_version="3.20" repo
2024-07-03T10:04:35-04:00       INFO    Number of language-specific files       num=1
2024-07-03T10:04:35-04:00       INFO    [python-pkg] Detecting vulnerabilities...

python:3.10.14-alpine3.20 (alpine 3.20.0)

Total: 8 (UNKNOWN: 0, LOW: 0, MEDIUM: 8, HIGH: 0, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	
busybox	CVE-2023-42364	MEDIUM	fixed	1.36.1-r28	1.36.1-r29	busybox: us

Grype + Syft (генератор ППК для образа)

Container Vulnerability Report

Name: python:3.10.14-alpine3.20
Type: image
Date: Monday, June 17, 2024 at 05:30 PM

grype

0	1	8	0	1
Critical	High	Medium	Low	Unknown

Search: Export to PDF

Name	Version	Type	Vulnerability	Severity	State	Fixed In
▸ busybox	1.36.1-r28	apk	CVE-2023-42365	Medium	fixed	1.36.1-r29
▸ busybox	1.36.1-r28	apk	CVE-2023-42364	Medium	fixed	1.36.1-r29

Анализ образа на соответствие лучшим практикам и выявление недостатков безопасности

Dockle

CIS Benchmark

1. Create a user for the container
2. Use trusted base images for containers
3. Do not install unnecessary packages in the container
4. Scan and rebuild the images to include security patches
5. Enable Content trust for Docker
6. Add HEALTHCHECK instruction to the container image
7. Do not use update instructions alone in the Dockerfile
8. Remove setuid and setgid permissions in the images
9. Use COPY instead of ADD in Dockerfile
10. Do not store secrets in Dockerfiles
11. Install verified packages only

Original Checkpoints

1. Avoid null password user (CVE-2019-5021)
2. Be unique UID/GROUPs
3. Avoid `sudo` command
4. Avoid sensitive directory mounting
5. Avoid `apt-get upgrade`, `apk upgrade`, `dist-upgrade`
6. Use `apk add` with `--no-cache`
7. Clear `apt-get` caches
8. Avoid `latest` tag

Безопасность k8s кластера (анализ статическими методами)

Особенности k8s
- YAML
- Self-documenting
- Декларативная природа
  - Infrastructure as Code
  - Security as Code
  - …

Многие проверки можно совершить **автоматически** с помощью инструментов статического анализа:
- **Kube-bench**
- **Checkmarx/KICS**
- **checkov**

```
# Пример «плохой» конфигурации пода
apiVersion: v1
kind: Pod
metadata:
   name: pod1
spec:
   containers:
   - name: example-container
     image: example/image
     command: ["/bin/sleep", "99999"]
```

checkov

Check: CKV_K8S_30: Отсутствует **securityContext.** ИЛ настоятельно рекомендует следовать принципу наименьших привилегий: **сбрасывать все привилегии и точечно добавлять необходимые.**

```
<...>
securityContext:
      capabilities:
            drop:["ALL"]
            add:["NET_BIND_SERVICE"]
<...>
```

Анализ динамическими методами

securityContext на уровне Pod

```
apiVersion: v1
kind: Pod
metadata:
   name: example-pod
spec:
   securityContext:
      runAsUser: 1000
      fsGroup: 2000
   containers:
<...>
```

securityContext на уровне контейнера

```
<...>
   containers:
   - name: example-container
      image: example/image
      securityContext:
         capabilities:
            add: ["CAP_SYS_ADMIN"]
```

Если «статика» не может,
«динамика» поможет!

Проверка Linux Capabilities
capsh в запущенном контейнере/процессе на испытательном стенде

```
$ grep Cap /proc/PID/status
$ capsh --decode=00000000XXXXXXXX
$ capsh --print
```

Проверка работы приложения в контейнере в Natch
Отслеживание процессов запущенных от root;
Отслеживание процессов внутри контейнеров.

OpenJDK: Сценарий с десериализацией данных в java `ObjectInputStream()`

java-функция

```
JavaCalls::call_helper(JavaValue*, methodHandle const&, JavaCallArguments*, JavaThread*) make/hotspot/src/hotspot/share/runtime/javaCalls.cpp:346 void /DataObjectDeserializer::main(java
  0x7f03180218b8 java/lang/Object java/io/ObjectInputStream::readObject()
    0x7f0318021b28 java/lang/Object java/io/ObjectInputStream::readObject(java/lang/Class)
      0x7f0318024050 java/lang/Object java/io/ObjectInputStream::readObject0(java/lang/Class, boolean)
        0x7f03180257c0 java/lang/Object java/io/ObjectInputStream::readOrdinaryObject(boolean)
          0x7f03180394a0 byte java/io/ObjectInputStream$BlockDataInputStream::readByte()
            0x7f0318024760 java/io/ObjectStreamClass java/io/ObjectInputStream::readClassDesc(boolean)
              0x7f0318024eb0 java/io/ObjectStreamClass java/io/ObjectInputStream::readNonProxyDesc(boolean)
                0x7f03180221a0 java/lang/Class java/io/ObjectInputStream::resolveClass(java/io/ObjectStreamClass)
                  0x7f0318026310 java/lang/ClassLoader java/io/ObjectInputStream::latestUserDefinedLoader()
                    0x4413e0 java/lang/Class java/lang/Class::forName(java/lang/String, boolean, java/lang/ClassLoader)
                      0x1e5a8 java/lang/SecurityManager java/lang/System::getSecurityManager()
                        0x4412c8 java/lang/Class java/lang/Class::forName0(java/lang/String, boolean, java/lang/ClassLoader, ja lang/Class)
                          Java_java_lang_Class_forName0 (libjava.so)
                            jni_GetStringUTFRegion make/hotspot/src/hotspot/share/prims/jni.cpp:2783 (libjvm.so)
                              UNICODE::as_utf8(signed char const*, int, char*, int) make/hotspot/src/hotspot/share/utilities/utf8.cpp:462 (
```

cpp-функция в JVM

1

2

```
4afbcaf  jdk17 / src / hotspot / share / utilities / utf8.cpp

Code  Blame  532 lines (495 loc) · 16.1 KB

461
462  char* UNICODE::as_utf8(const jbyte* base, int length, char* buf, int buflen) {
463    assert(buflen > 0, "zero length output buffer");
464    u_char* p = (u_char*)buf;
465    for (int index = 0; index < length; index++) {
466      jbyte c = base[index];
467      int sz = utf8_size(c);
468      buflen -= sz;
469      if (buflen <= 0) break; // string is truncated
470      if (sz == 1) {
471        // Copy ASCII characters (UTF-8 is ASCII compatible)
```

3

```
Commit f5ab7f6

tstuefe committed on Mar 2, 2021

8262472: Buffer overflow in UNICODE::as_utf8 for zero length output buffer

Reviewed-by: dholmes, iklam
```

4

```
__AFL_FUZZ_INIT();

/* Main entry point. */

/* To ensure checks are not optimized out it is recommended to disable
   code optimization for the fuzzer harness main() */
#pragma clang optimize off
#pragma GCC            optimize("O0")


int main(int argc, char **argv) {

  ssize_t       len;                    /* how much input did we read? */
  unsigned char *buf;                   /* test case buffer pointer    */

  __AFL_INIT();

  buf = __AFL_FUZZ_TESTCASE_BUF;  // this must be assigned before __AFL_LOOP!

  while (__AFL_LOOP(UINT_MAX)) {  // increase if you have good stability

    len = __AFL_FUZZ_TESTCASE_LEN;

    int utf8_len = utf8_length((jbyte*) buf, len);

    u_char* res[utf8_len + 1];

    as_utf8((jbyte*) buf, len, (char*) res, utf8_len + 1);
  }
```

5

```
AFL ++4.22a (default) (./utf8-as_utf8-fuzz-coverage) [explore]
process timing                                  overall results
        run time : 0 days, 1 hrs, 23 min, 0 sec     cycles done : 42.8k
   last new find : 0 days, 1 hrs, 13 min, 0 sec     corpus count : 54
last saved crash : none seen yet                  saved crashes : 0
 last saved hang : none seen yet                   saved hangs : 0
cycle progress                                  map coverage
  now processing : 15.259195 (27.8%)              map density : 21.35% / 37.08%
  runs timed out : 0 (0.00%)                    count coverage : 13.12 bits/tuple
stage progress                                  findings in depth
  now trying : splice 13                        favored items : 7 (12.96%)
 stage execs : 3/12 (25.00%)                     new edges on : 7 (12.96%)
total execs : 567M                             total crashes : 0 (0 saved)
  exec speed : 14.8k/sec                        total tmouts : 0 (0 saved)
```

Research, Research, Research...

ИСП РАН, Профископ, НТЦ Фобос-НТ и Базальт СПО в Великом Новгороде, 2025 г.

Natch

К следующей неделе готовится большая статья по результатам совместной работы...

А пока можно почитать следующее:

Как найти поверхность атаки незнакомых приложений с помощью Natch

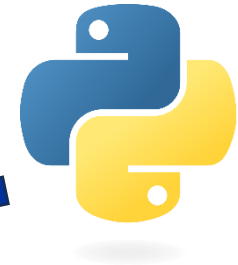
Статический анализ

Организация работы с командой статического анализа

Руками в Excel

Автоматом в

Svacer



Автоматизация

Принципы работы :

1. Централизованная выдача задач;
2. Централизованный сбор статистики;
3. Разметка в соответствии с регламентом Центра.

Плюсы:

1. Сбор размеченных сработок;
2. Автоматическое наполнение trophy case;
3. Прививание размечающим единого стиля разметки и написания комментариев.

Статистика разметок по пакетам

Контракт	Пакет	Количество разметок				Всего
		Confirmed	Won't fix	False Positive	Ревью	
Контракт №1	activemq 2.25.0	33	291	31	0	355
	apacheds 2.0.0	20	74	6	0	100
	qemu 8.0.5	0	35	16	0	51
	directory-ldap-api 2.1.6	16	148	6	0	170
Итого:						676

Пользовательские детекторы

Перенос детекторов из Semgrep в Svace

Детектор libxml2-expand-remote-dtd

```
36  int main(int argc, char **argv) {
37      // ruleid: libxml2-expand-remote-dtd
38      xmlDocPtr doc = xmlReadMemory(argv[1], sizeof(buf),
        "noname.xml", NULL, XML_PARSE_DTDLOAD |
        XML_PARSE_NOENT);
39  }
```

API

USER.XXE_LIBXML Potential XXE vulnerability : libxml2 function 'xmlReadMemory' contains options, that could lead to XXE attack

```
xmlDocPtr doc = xmlReadMemory(buf, sizeof(buf),
 "noname.xml", NULL, XML_PARSE_NOENT | XML_PARSE_DTDLOAD ;

xmlFreeDoc(doc);
```

Разработка детекторов

Пример использования tainted()

```
@Override
public void createChecker() {
/*1*/    SvaceLightPattern tainted_val = tainted();
/*2*/    String arg_name = "exec(?:(vpe)|(lp)|(le))";
/*3*/    SvaceLightPattern exec_call = funcCall(arg_name, 0, tainted_val);
/*4*/    registerChecker(warningOSCommand,
                "First argument in function is tainted.", exec_call);
}
```

API

USER.OS_EXEC_COMMANDS Incorrect call to execvpe with arguments (sid'89'argument, sid'90'argv, sid'91'env[0]). First...

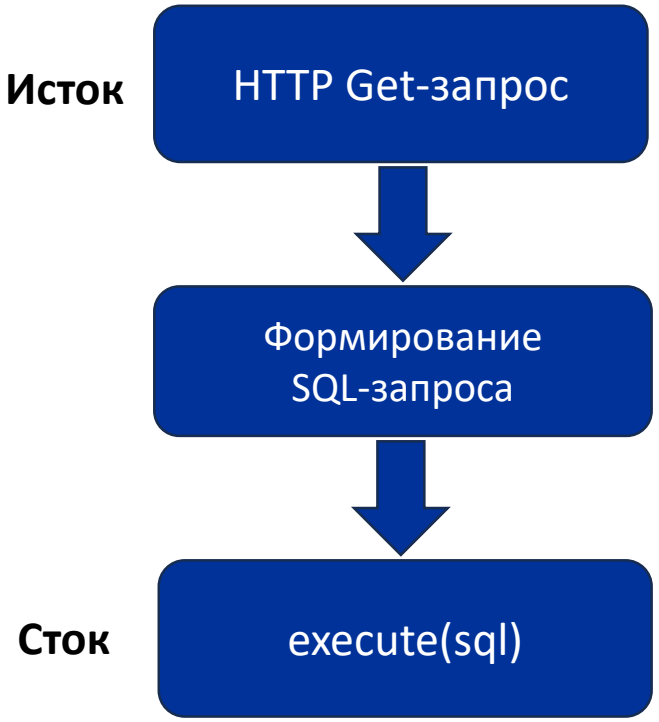
```
execvpe(argument, argv, env); //vuln
execvpe(argv[0], argv, env); // vuln
execle(argv[0], argv[0], "-al", NULL, env3); // vuln
```

Репозиторий ИСП РАН
пользовательских детекторов для SvaceAPI!!!

Анализ помеченных данных

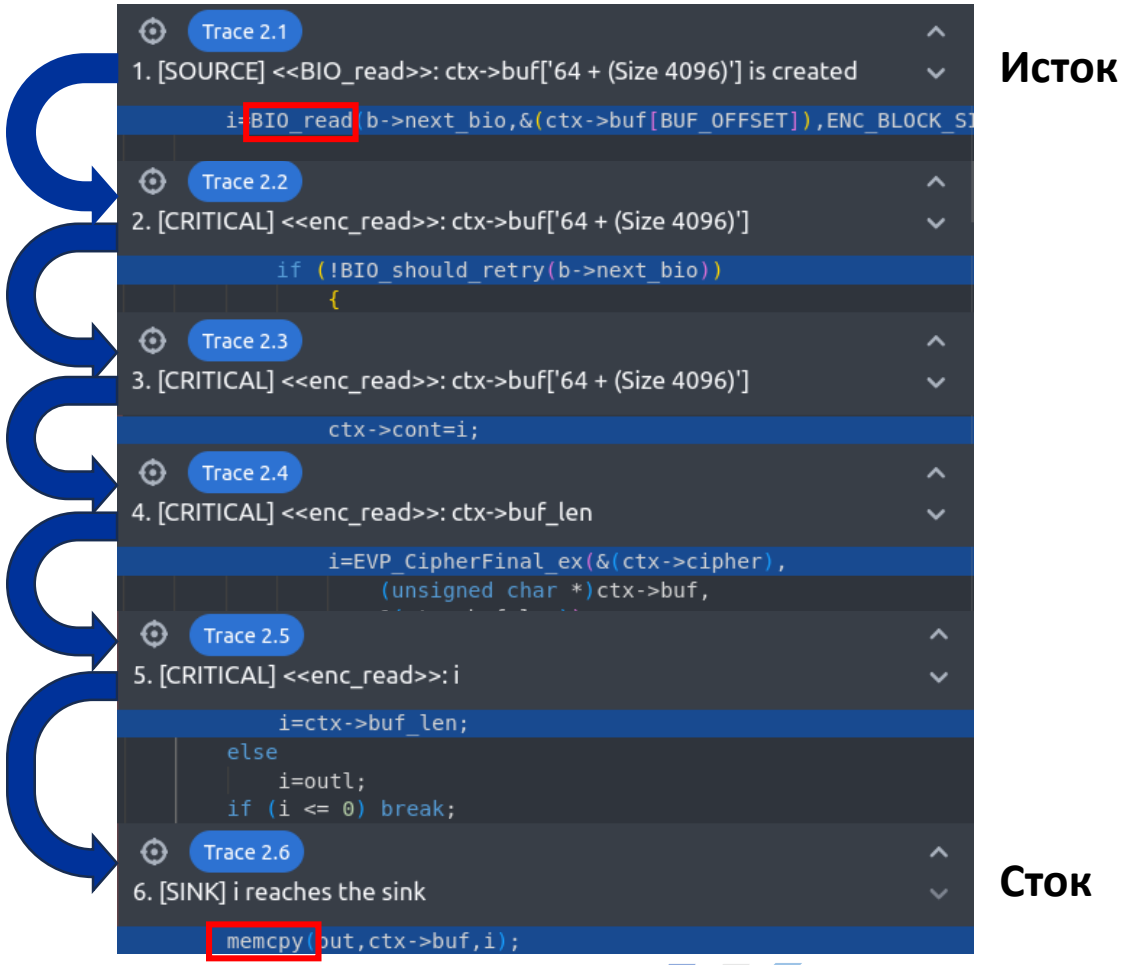
Taint-анализ - обнаружение потенциальных уязвимостей в коде программ путём исследования потоков данных между истоками и стоками помеченных данных.

Пример Tainted данных



Участвуем в апробации статического анализатора помеченных данных **Ирбис**.

Пример трассы маркера



Наши результаты за 2024 год

Обработано **более 3500** сработок статического анализатора Svace.

Более 40 рекомендаций по улучшению и исправлению дефектов кода принято в основную ветку разработки исследуемых компонентов для компонентов:

1. Apache Directory LDAP API;
2. nss-pam-ldapd;
3. dotnet;
4. activemq-artemis;
5. ApacheDS;
6. libsocket;
7. linux-pam;
8. Kubernetes;
9. Qemu;
10. Gnutls;
11. 389-ds-base;
12. Cpython.

Фобос-НТ
НАУЧНО-ТЕХНИЧЕСКИЙ ЦЕНТР

Home Статистика Распределение

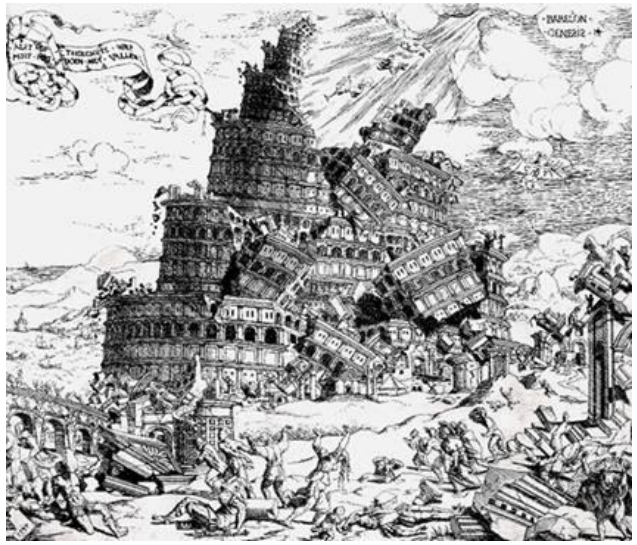
Show 10 entries Search:

№	Пакет	Ссылка	Дата фикса	Автор
1	crun	DEREF_AFTER_NULL.EX /src/libcrun/linux.c: 2134	2024-11-14	p.nekrasov
2	crun	INTEGER_OVERFLOW /src/create.c: 170	2024-11-14	p.nekrasov
3	389-ds-base 2.2.9	LIB.INSECURE_STRNCMP /389-ds-base/src/ldap/servers/slapd/charray.c: 358	2024-10-31	p.nekrasov
4	activemq 2.25.0	BAD_COPY_PASTE /artemis-protocols/artemis-stomp-protocol/src/main/java/org/apache/activemq/artemis/core/protocol/stomp/VersionedStompFrameHandler.java: 275	2024-03-19	a.slepykh
5	activemq 2.25.0	BAD_COPY_PASTE /artemis-server/src/main/java/org/apache/activemq/artemis/core/server/impl/jdbc/ActiveMQScheduledLeaseLock.java: 138	2024-03-19	a.slepykh
6	activemq 2.25.0	FB.DLS_DEAD_LOCAL_STORE /artemis-protocols/artemis-amqp-protocol/src/main/java/org/apache/activemq/artemis/protocol/amqp/broker/AMQPLargeMessage.java: 320	2024-03-19	a.slepykh
7	activemq 2.25.0	FB.UUF_UNUSED_FIELD /artemis-maven-plugin/src/main/java/org/apache/activemq/artemis/maven/ArtemisCLIPlugin.java: 34	2024-10-25	a.slepykh
8	activemq 2.25.0	FB.UUF_UNUSED_FIELD /artemis-maven-plugin/src/main/java/org/apache/activemq/artemis/maven/ArtemisCreatePlugin.java: 44	2024-10-25	a.slepykh

Фаззинг-тестирование

Зачем нам нужен фаззинг конвейер?

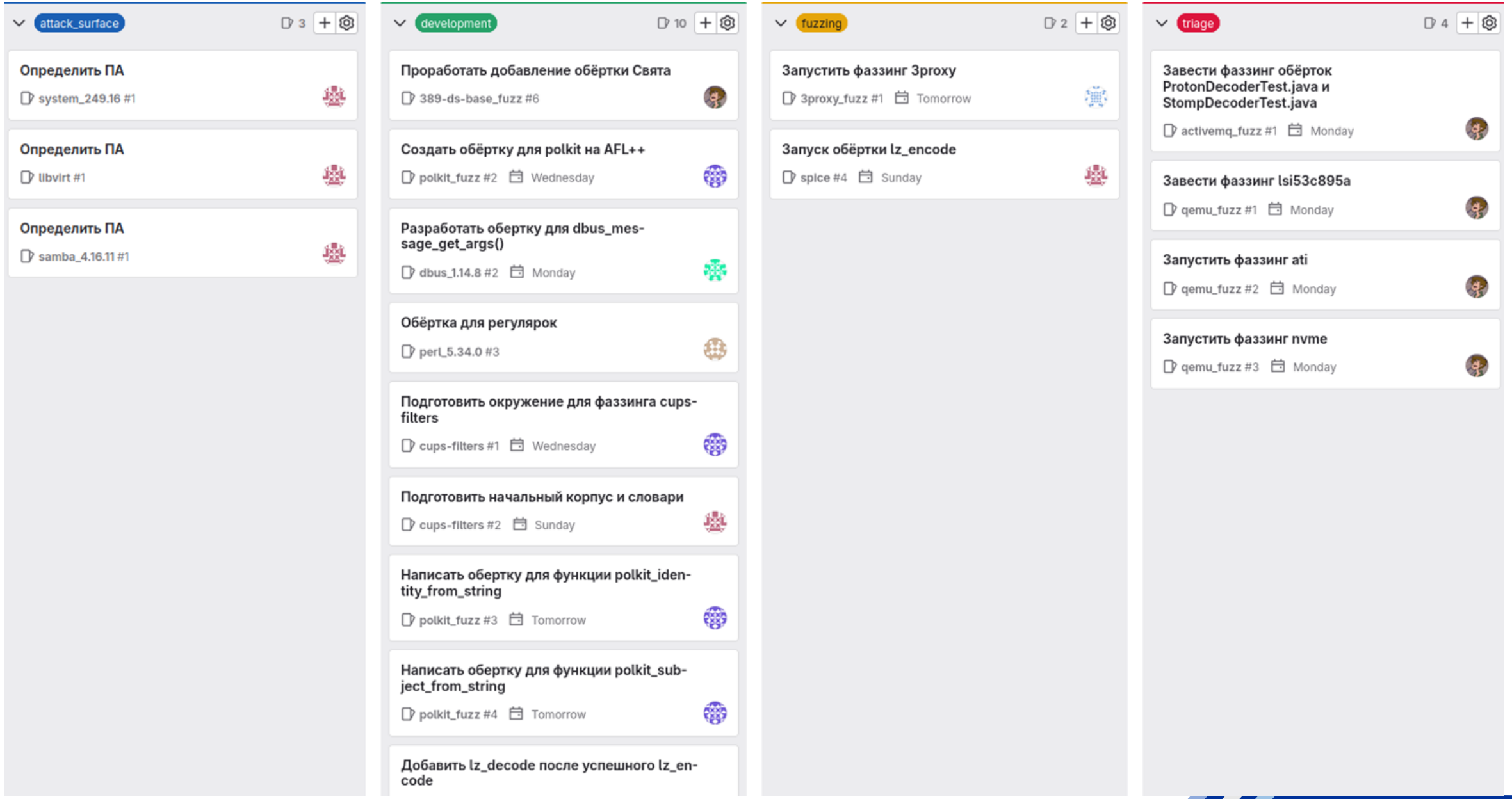
- совместная разработка фаззинг обёрток
- распределение ролей в команде
- автоматизация фаззинг процессов
- управление фаззинг процессами



Фаззинг без конвейера

Фаззинг с конвейером

Совместная разработка на gitlab

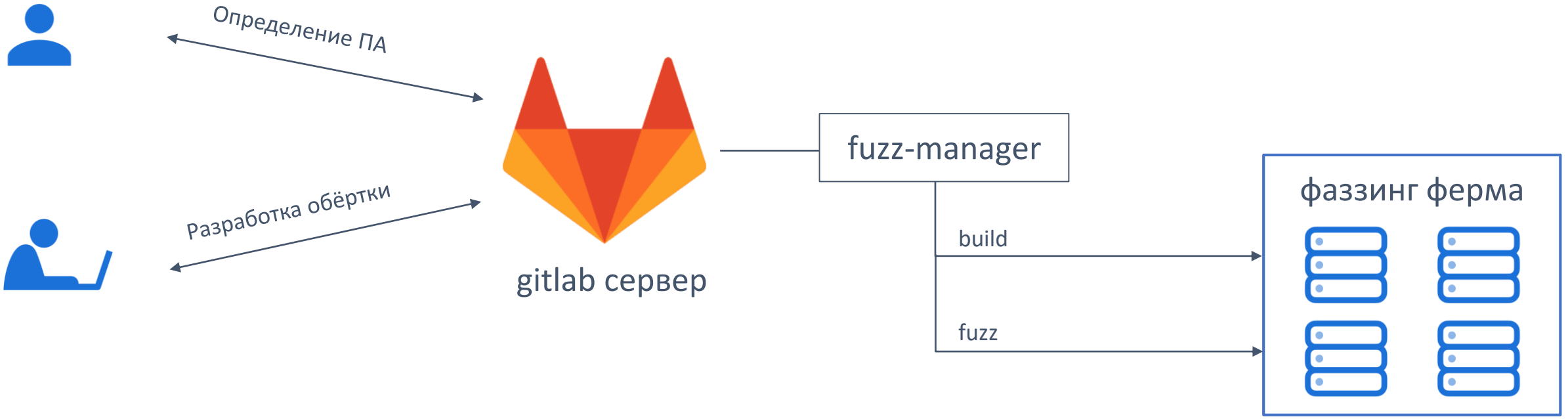


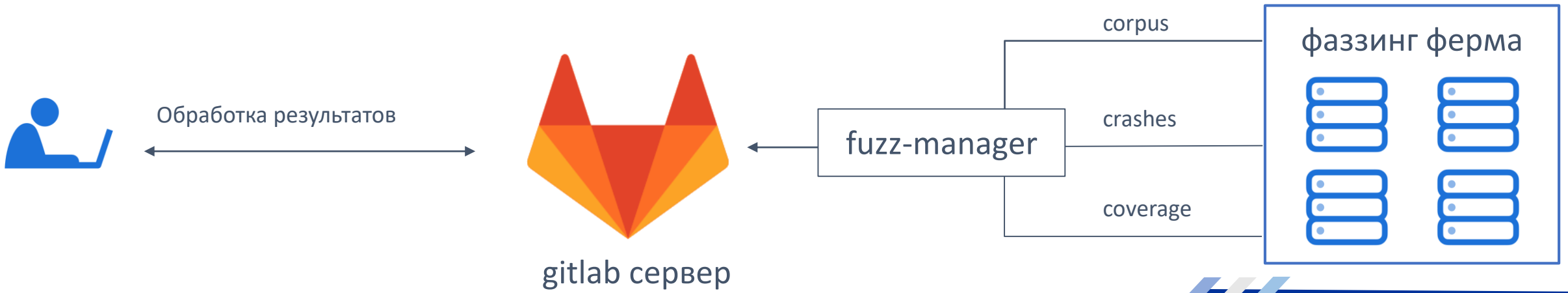
Зачем нам нужен фаззинг конвейер?

- совместная разработка фаззинг обёрток
- распределение ролей в команде
- автоматизация фаззинг процессов
- управление фаззинг процессами

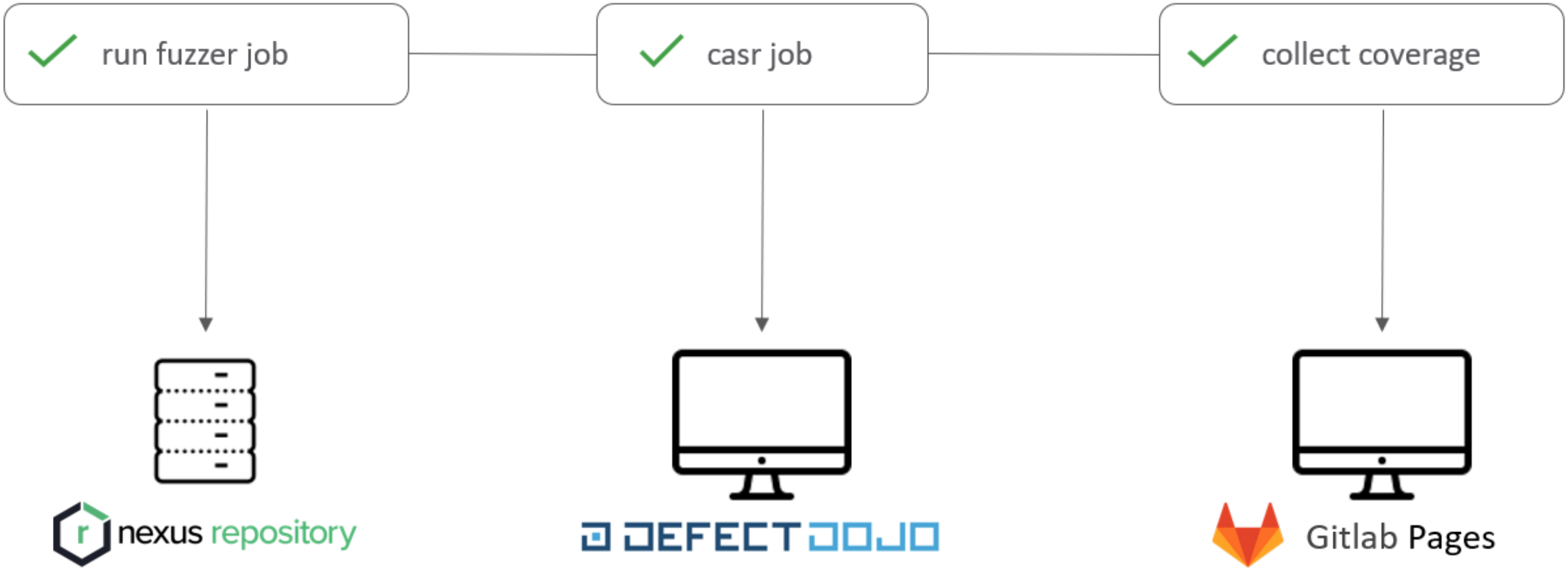
Разработка первого пайплайна
(фото в цвете MMXXIV г. н.э)

Зачем нам нужен фаззинг конвейер?



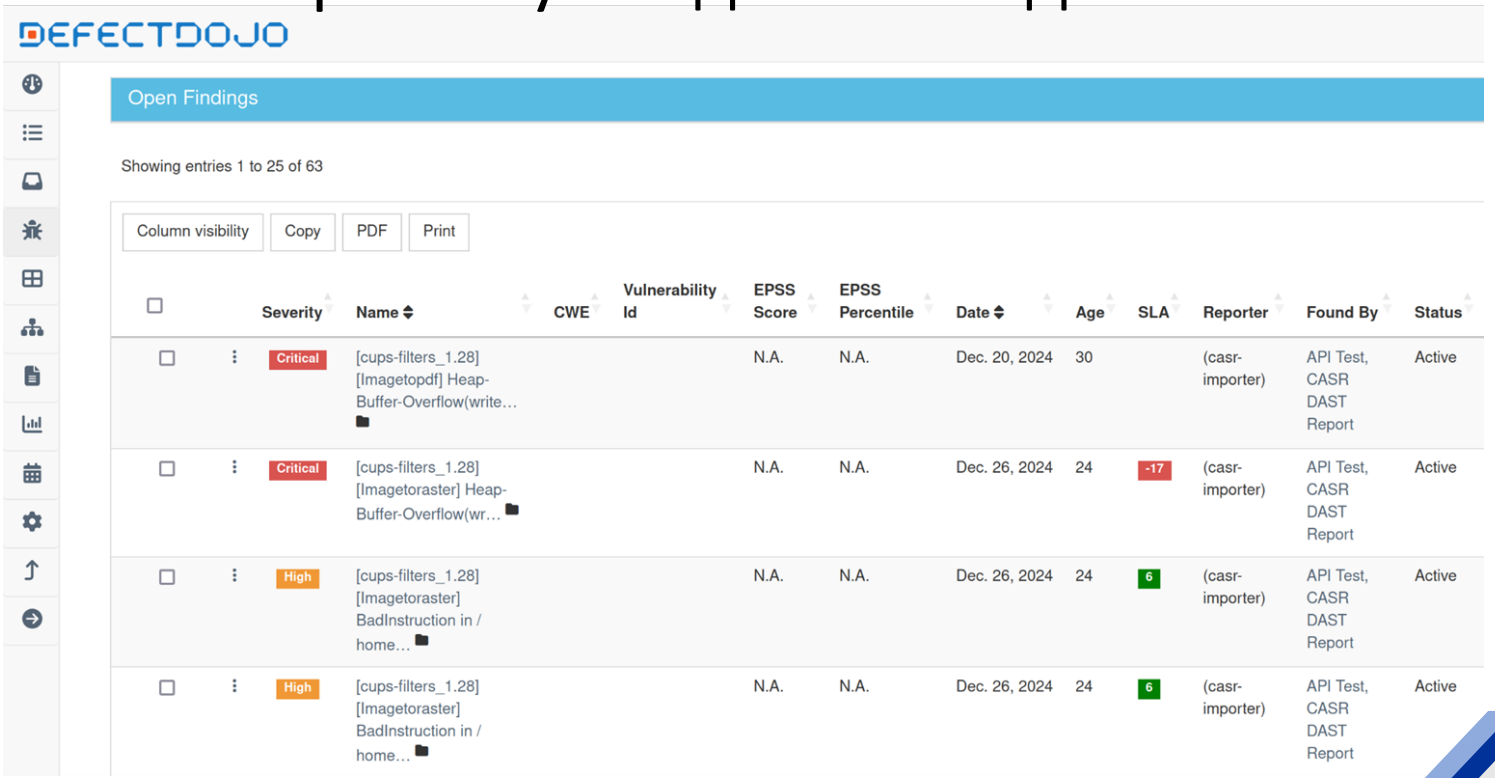


Обработка наработанных артефактов



Обработка наработанных артефактов

- организовать совместную работу команды из 8 человек
- автоматически задействовать доступные ресурсы для фаззинга
- уменьшить время после написания обёртки до запуска фаззинга
- шаблонизировать типичные стадии фаззинг тестирования
- централизовать обработку найденных падений



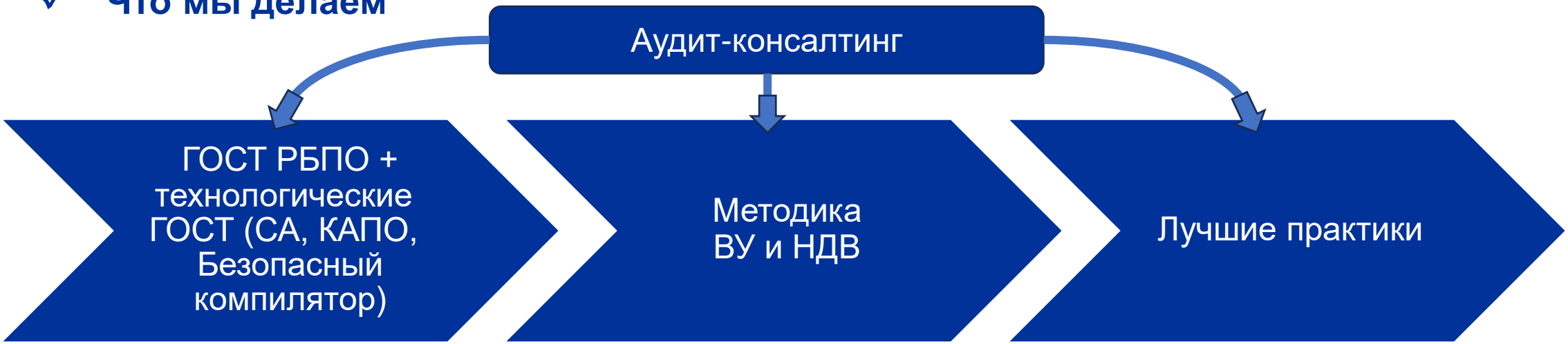
Аудит процессов разработки безопасного программного обеспечений

Аудит процессов РБПО

✓ **Что даёт**

- Возможность получить «взгляд со стороны» ➡
- ➡ Возможность выстроить безопасную и **качественную** разработку!

✓ **Что мы делаем**



Фобос-НТ входит в ТК362 и участвует в разработке стандартов семейства РБПО

Аудит процессов РБПО

✓ Как устроен процесс

- 3 месяца (7-8 встреч, очно/ВКС)

Аудит:

- изучение документации,
- изучение организации взаимодействия подразделений,
- изучение качества разработки и тестирования,
- изучение конвейера сборки.

✓ Что в итоге

- **не просто** оценка соответствия, **а** комплект рекомендаций по развитию процессов для конкретного Заказчика (базовый или расширенный комплект рекомендаций).

Безопасная сборка

ИСП РАН

- SAFEC (на основе GCC);
- Safelang (на основе Clang).

ЕДЕРАЛЬНОЕ АГЕНТСТВО
ОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 71206— 2024

Защита информации

РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. БЕЗОПАСНЫЙ КОМПИЛЯТОР ЯЗЫКОВ С/С++

Общие требования

Издание официальное

Москва
Российский институт стандартизации
2024

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56939– 202X
(проект, окончательная редакция)

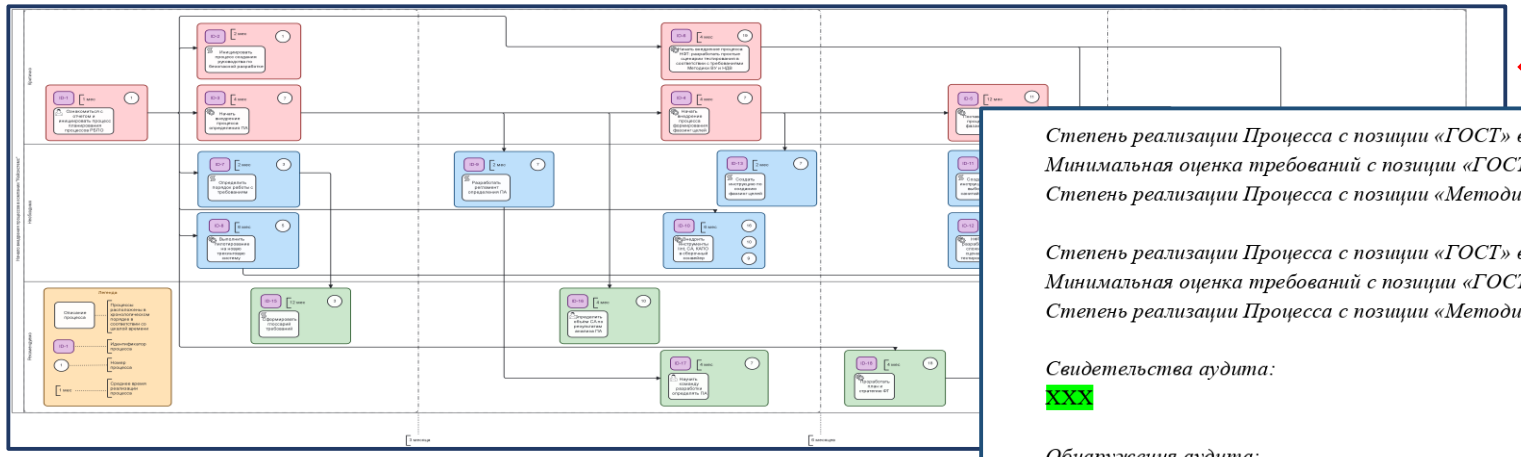
Защита информации

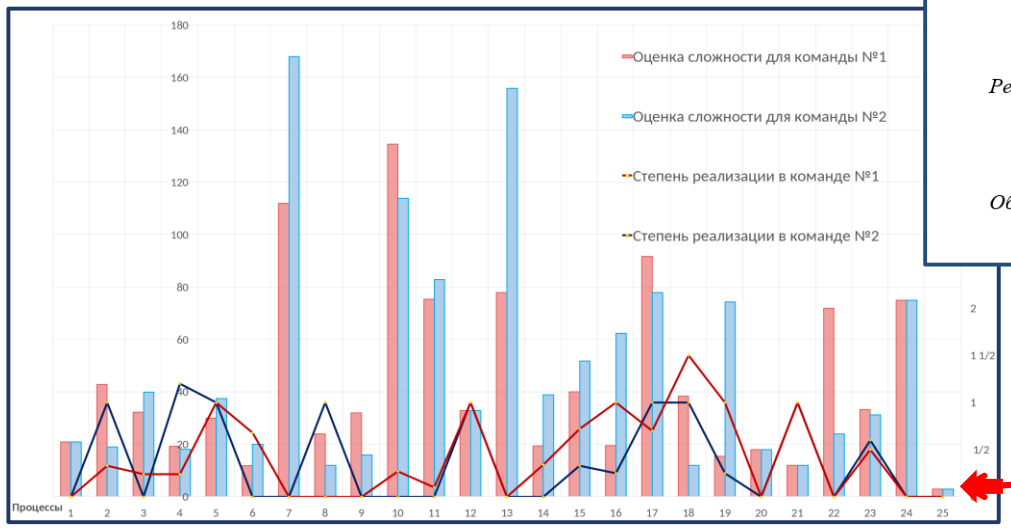
РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Общие требования

Москва
ФГБУ «Российский институт стандартизации»
202X

Аудит процессов РБПО





Степень реализации Процесса с позиции «ГОСТ» в команде №1: X
Минимальная оценка требований с позиции «ГОСТ» в команде №1: X
Степень реализации Процесса с позиции «Методика» в команде №1: -

Степень реализации Процесса с позиции «ГОСТ» в команде №2: X
Минимальная оценка требований с позиции «ГОСТ» в команде №2: X
Степень реализации Процесса с позиции «Методика» в команде №2: -

Свидетельства аудита:
XXX

Обнаружения аудита:
a. XXX
b.

Рекомендации по устранению несоответствий
a. XXX
b.

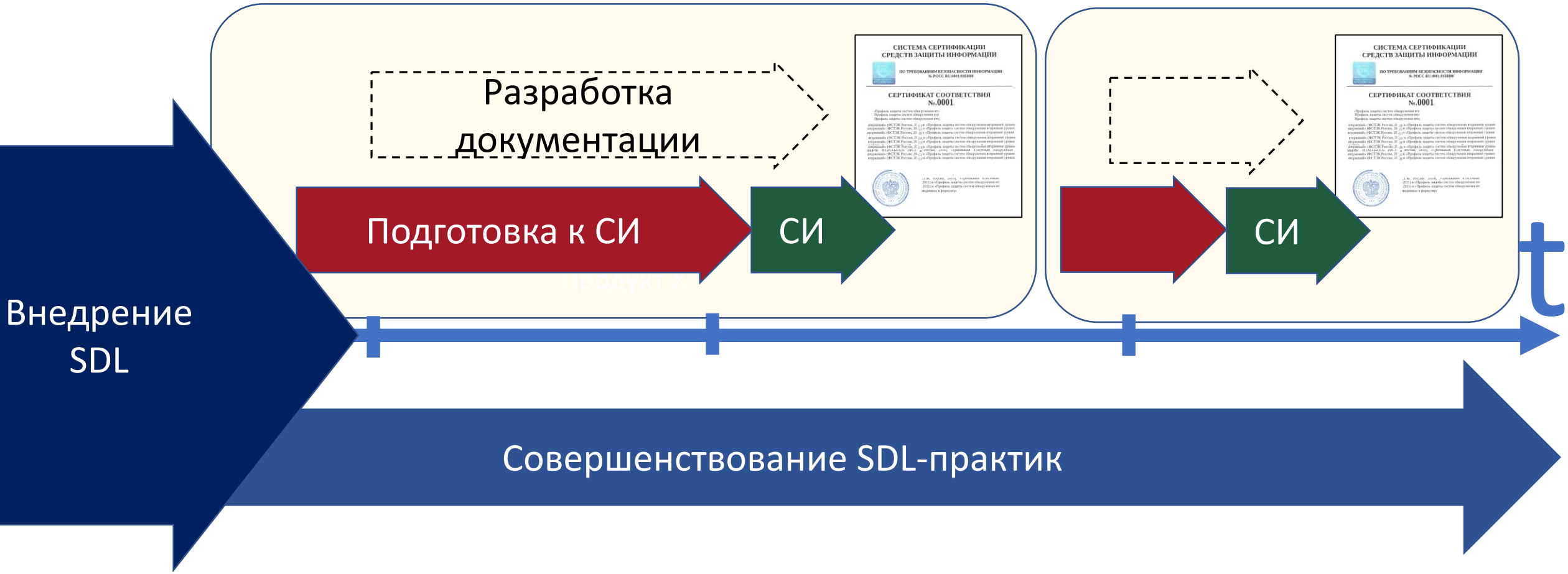
Общие рекомендации:
– XXX

Расширенный комплект

- Отчёт
- Отчёт — **база**
- Приложение №2
- Приложение №4
- Приложение №5

ИИ...

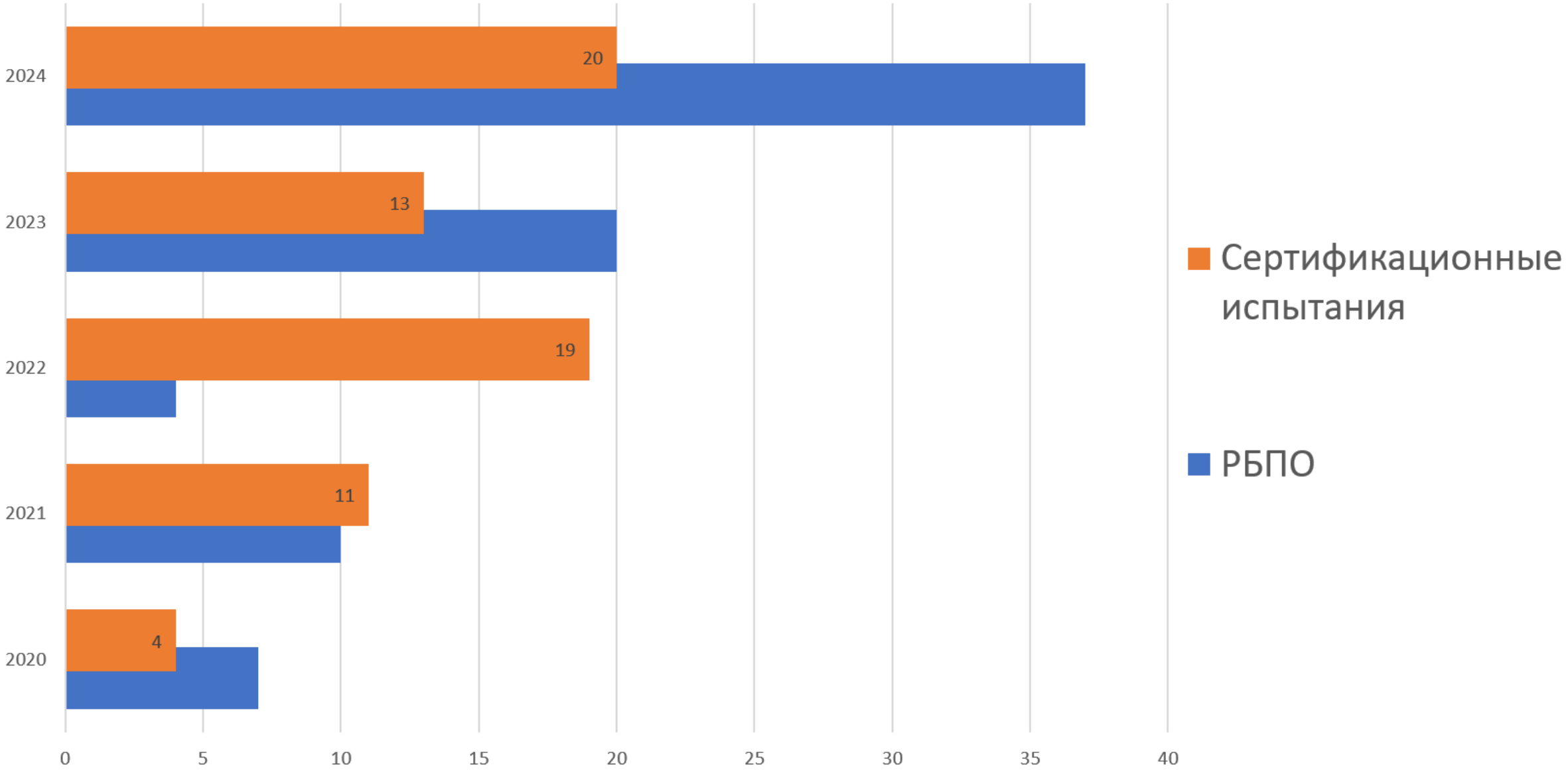
Сертификационные испытания



Что это значит для бизнеса:

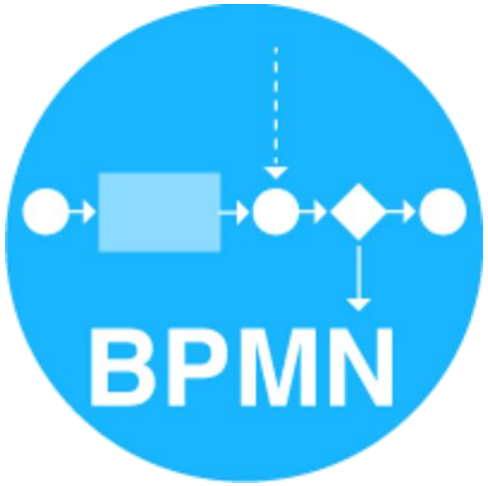
– **предсказуемые сроки** получения сертификата соответствия
– в первую очередь **вклад в свою команду**, а не в лабораторию за «*магические действия*»
– готовность к **сертификации процессов** РБПО

Немного статистики



Нужно бежать со всех ног, чтобы только оставаться на месте

Тестирование на проникновение



Моделирование бизнес-процессов

Автоматизация сборки и тестирования

Нотация моделирования бизнес-процессов

Композиционный анализ

Безопасность средств оркестрации

Спасибо за внимание!