ИСП РАН

Разработка доверенного ПО
Вызовы и перспективы

ПАДАРЯН В.А.
Ведущий научный сотрудник ИСП РАН
vartan@ispras.ru

11 февраля 2021

Уязвимости – причина компьютерных атак

Принципиальное наличие **уязвимостей*** в ПО и аппаратуре

- функциональные
- архитектурные
- программного кода/микрокода

***Границы между ошибками программиста, закладками, НДВ размыты**

Утечка информации о уязвимостях

Хакеры «для интереса»

- Взломы ради известности
- Ограничены ресурсы
- Только известные эксплойты

Хакеры «для ущерба»

- Вандализм
- Ограничены ресурсы

Преступность

- Взлом ради прибыли
- Существенные ресурсы
- Синдикаты
- Специально разработанные программы для кражи данных

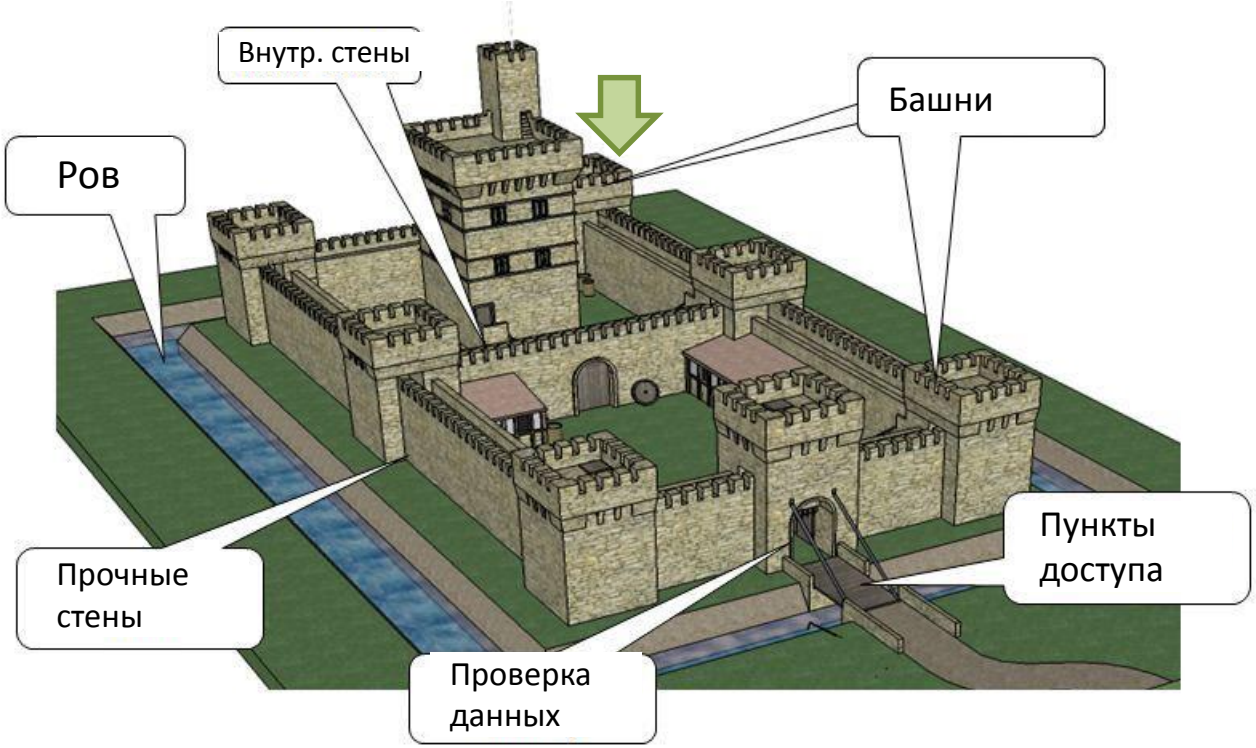
Спецслужбы

- Атаки на критические информационные инфраструктуры, промышленный шпионаж
- Практически неограниченные ресурсы
- Сложное ВПО и программы для взлома
- Постоянные угрозы

Рост ресурсов и сложности атак

Классические методы защиты не являются основными

- Защита по периметру
- Организационные мероприятия (проверка доступа)
- Антивирусы и др.



Вызовы

- Эскалация размеров. ПО – большие данные (размер Astra Linux – 150 млн. строк кода, размер Debian – более миллиарда строк)
- Сложность среды разработки и сборки (*например, компилятор может добавить уязвимость, которой нет в исходном коде*)
- Облачно-мобильные платформы, ИИ, «Интернет вещей» (отсутствие локально изолированных систем)…

Зарубежный опыт: разработка безопасного ПО

- Госстандарты: Common Criteria (развитие с 1999), документы МО США (Program Protection Plan для всех этапов разработки, с 2000х)
- Технологии, инструменты анализа выбираются сертификационными лабораториями → разработчики ПО вынуждены пользоваться инструментами требуемого уровня
- Заложено непрерывное обновление стандартов по мере развития технологий (текущая версия CC – 3.1)
- Принятие нормативных документов привело к взрывному росту рынка технологий анализа
  https://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html

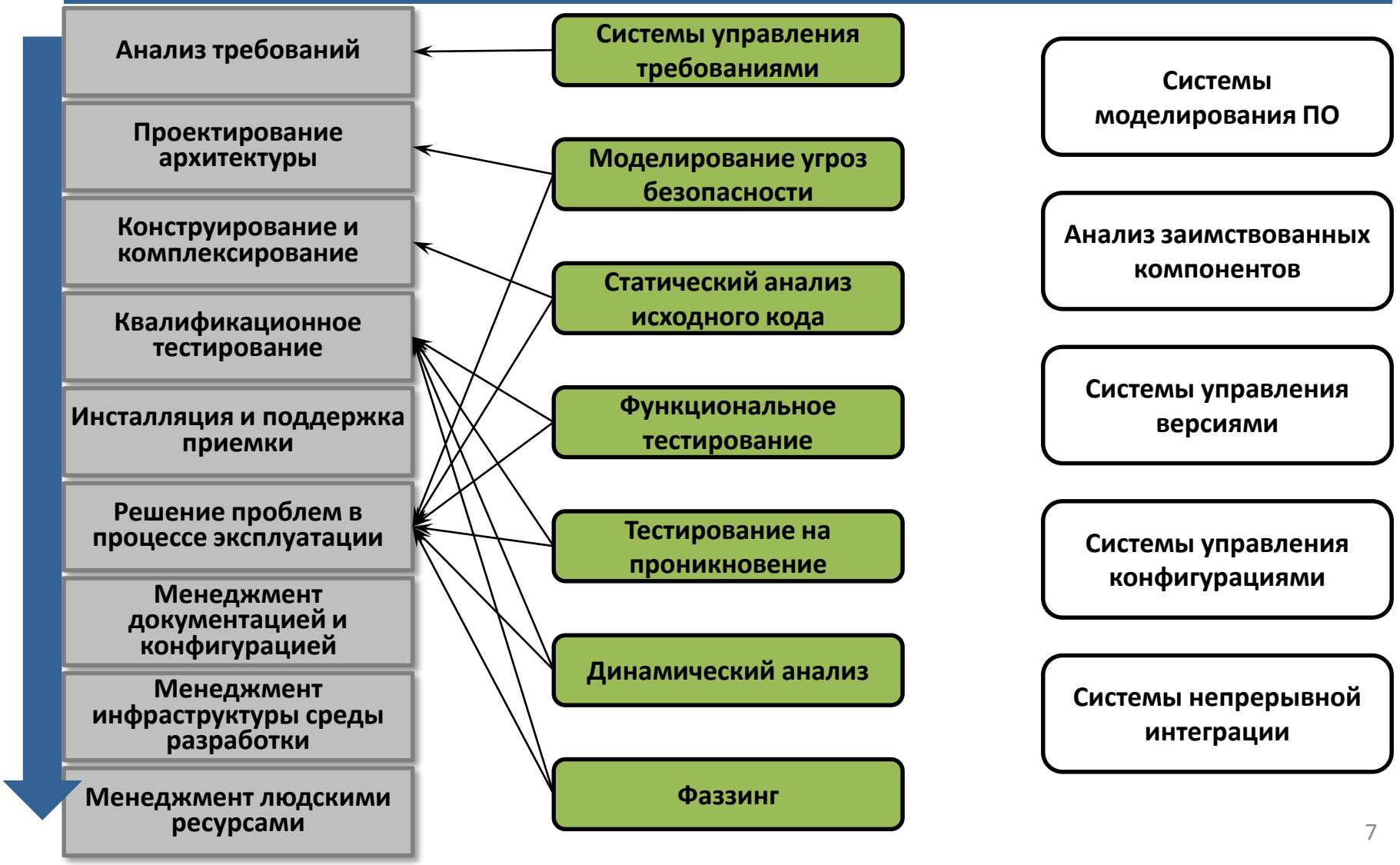
Документы ФСТЭК России

- **ГОСТ Р 56939-2016** «Защита информации. Разработка безопасного программного обеспечения. Общие требования» Утверждён и введен в действие **01.06.2016**

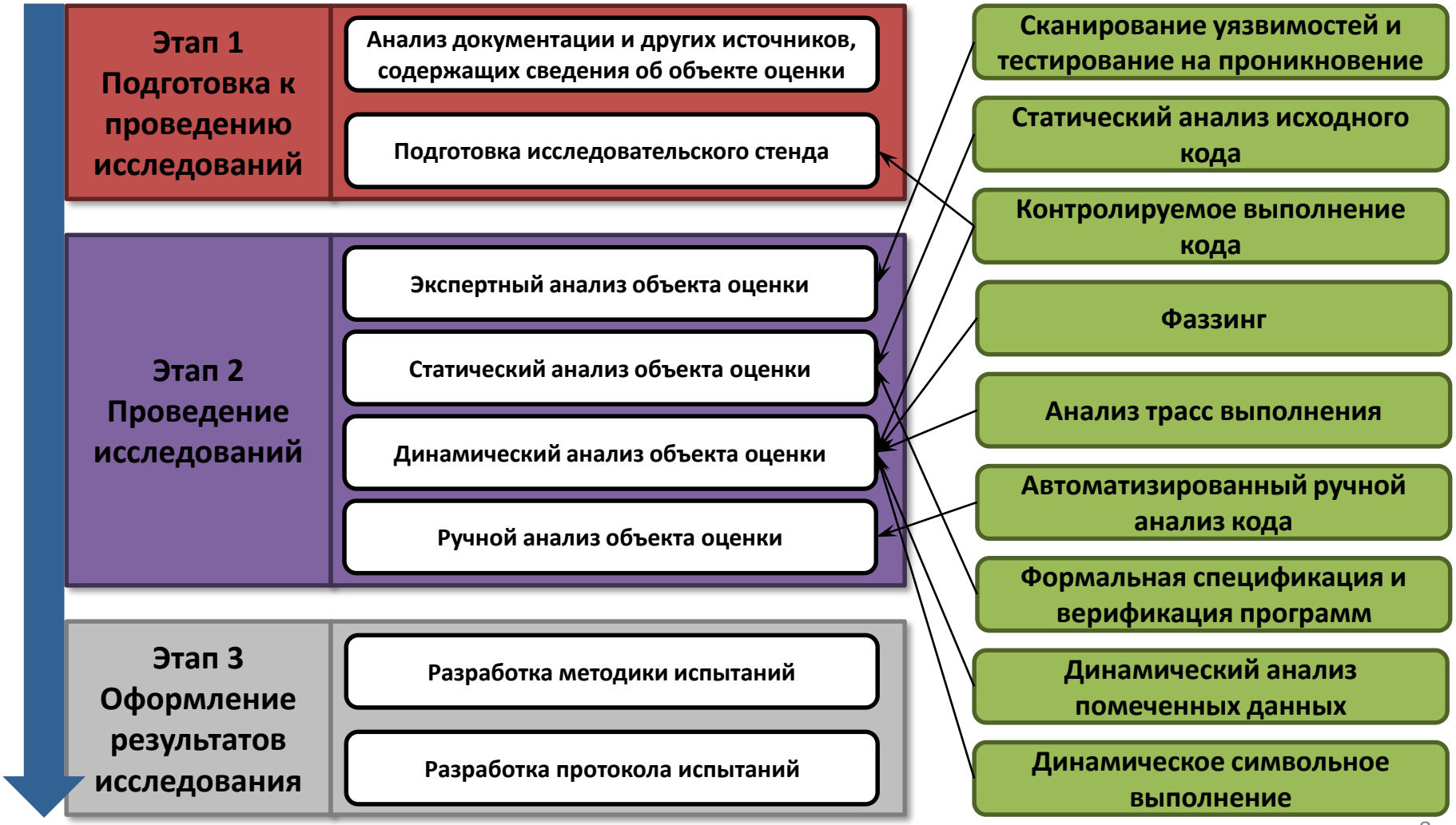
- «**Методика выявления уязвимостей и недекларированных возможностей в программном обеспечении**» утверждена ФСТЭК России 11 февраля 2019 г.
  - Применяется при проведении сертификационных испытаний **с 1 июня 2019 г.**

- ***Задано направление на применение в системе сертификации автоматизированных и автоматических технологий анализа безопасности программного кода***

ГОСТ Р 56939-2016



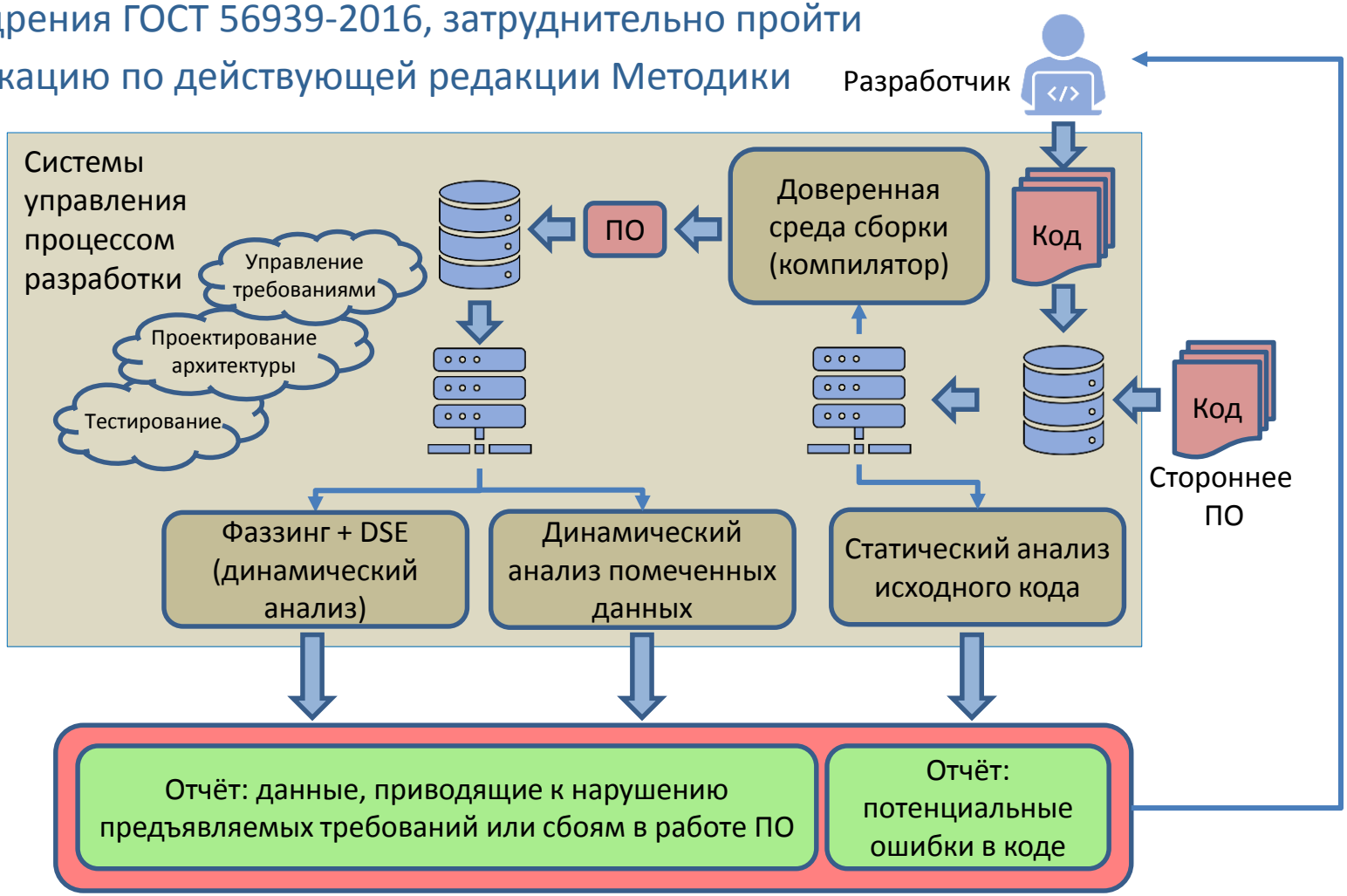
Методика выявления уязвимостей и НДВ★



★Первая редакция Методики, 2019 г.

Технологии анализа безопасности программного кода в процессах разработки

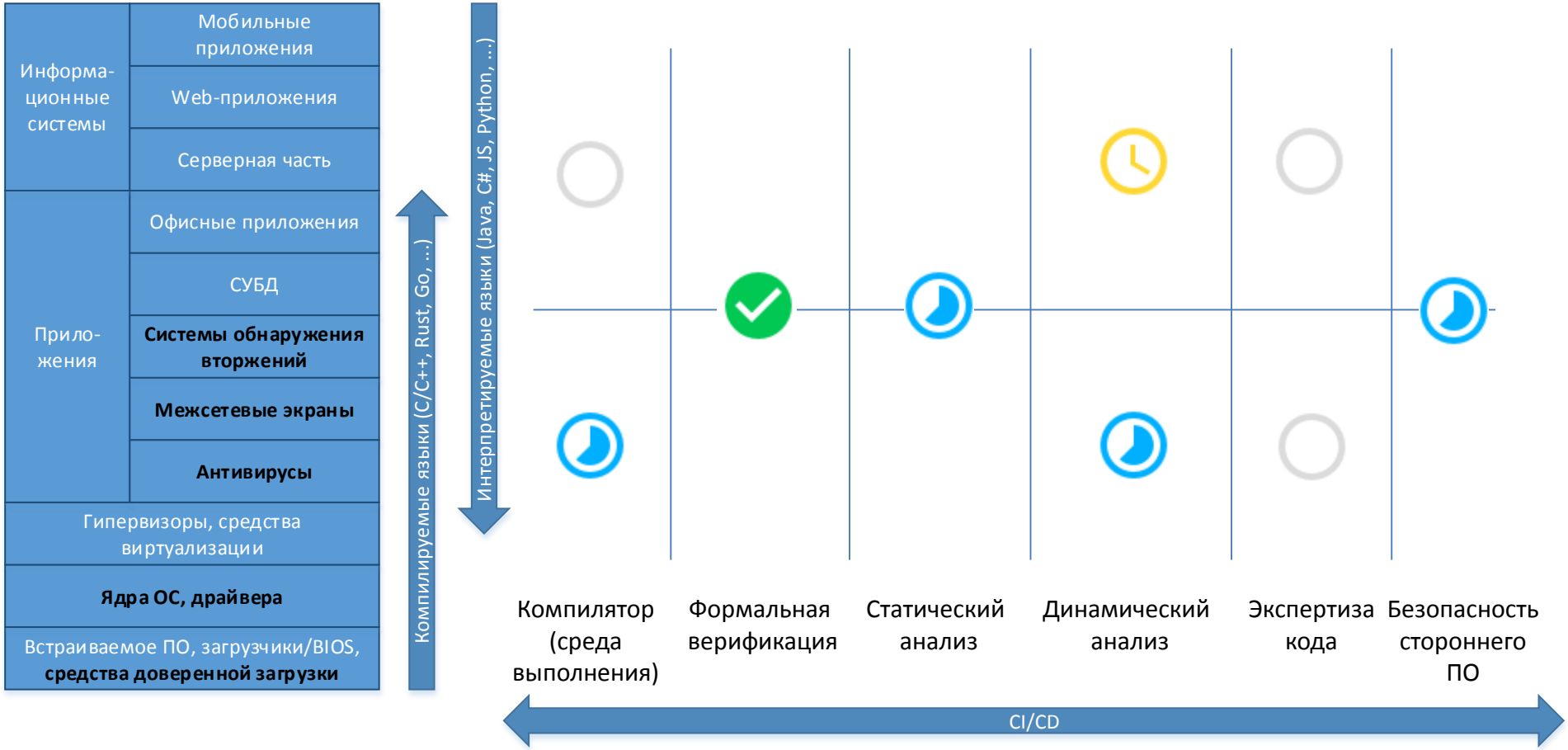
Без внедрения ГОСТ 56939-2016, затруднительно пройти сертификацию по действующей редакции Методики



Развитие нормативной базы в 2021 г.

- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Руководство по оценке безопасности разработки программного обеспечения**»
- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Руководство по проведению статического анализа. Общие требования**»
- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Руководство по проведению динамического анализа. Общие требования**»
- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Доверенный компилятор языков Си/Си++. Общие требования**»
- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Управление безопасностью программного обеспечения при использовании заимствованных и привлекаемых компонентов**»
- Пересмотр (**обновление**) **ГОСТ Р 56939-2016** «Защита информации. Разработка безопасного программного обеспечения. **Общие требования**»
- Пересмотр (**обновление**) **Методики выявления уязвимостей и недекларированных возможностей в программном обеспечении**

«Карта» стандартизации



Доверенная компиляция

- Уязвимости в программе могут появляться не только из-за ошибок в ее коде, но и в результате оптимизаций, выполняемых компилятором
- В набор средств для обеспечения безопасности ПО должны быть также включены средства разработки (компилятор)

Доверенный компилятор

- Требования доверия
  - Безопасные оптимизации: нельзя опираться на предположения об априорной «корректности» программы
    - Например, значения указателей различающихся типов могут совпадать, могут происходить целочисленные переполнения, …
  - Принудительная инициализация всех переменных
  - Предупреждение (ошибка компиляции) при обнаружении в коде неопределенных ситуаций
  - Встраивание механизмов защиты, санитайзеров и т.п.
- Необходимо сохранять приемлемую производительность
- Пилотный компилятор – на базе gcc

Статический анализ исходного кода

Технологии:

1. Доверенный компилятор
2. Инструменты статического анализа программ

- Доступные на рынке инструменты:
- Svace (ИСП РАН, Россия)
- Klocwork (RogueWave, США)
- Coverity (Synopsys, США)
- Java FindBugs (Maryland University, США)
- Pyhton: PyLint, PyType
- JavaScript: JSHint, JSLint, Google Closure Linter
- …

Фаззинг и динамический анализ

Технологии:

1. Фаззинг + Sanitizers
2. Динамическое символьное исполнение
3. Комбинированные инструменты

- Доступные на рынке инструменты:
- Anxiety (ИСП РАН, Россия)
- ИСП Фаззер (ИСП РАН, Россия)
- Peach Fuzzer (США, Peach Tech)
- Synopsys Defensics (Synopsys, США)
- angr (Open source)
- Americal Fuzzy Lop (Open source)
- Driller (Open source)
- …

Динамический анализ помеченных данных

Технологии:

1. Контролируемое выполнение в виртуальной машине
2. Автоматизированная обратная инженерия бинарного кода
3. Анализ потоков данных и управления на уровне бинарного кода

Доступные в РФ инструменты:

- Среда анализа бинарного кода ТРАЛ (ИСП РАН, Россия)

Известные инструменты:

- MAYHEM (ForAllSecure, США)
- APAC (2013-2015), VET (2013), CGC (2016), CHESS (2019) (Проекты DARPA)

Технологии ИСП РАН жизненного цикла разработки безопасного ПО

Разра-ботка

- **Svace и Binside: статический анализ исходного и бинарного кода**
- **Поиск клонов кода: ошибки, нарушения лицензии**
- **Дедуктивный анализ моделей безопасности**

Тестиро-вание

- **Контролируемое выполнение на базе эмулятора QEMU**
- **Crusher (ISP Fuzzer и SyDr): фаззинг и динамическое символьное исполнение, расширенное тестирование**

Выпуск

- **Доверенный компилятор**
  - Диверсификация кода
  - ИСП–Обфускатор: защита кода от анализа
  - Доработка системных защитных механизмов (ASLR …)

Реагиро вание

- **Непрерывный поиск дефектов в выпущенном ПО**
- **Анализ аварийных завершений для оценки критичности программных дефектов**

Bug Bounty Program: жизнь после SDL

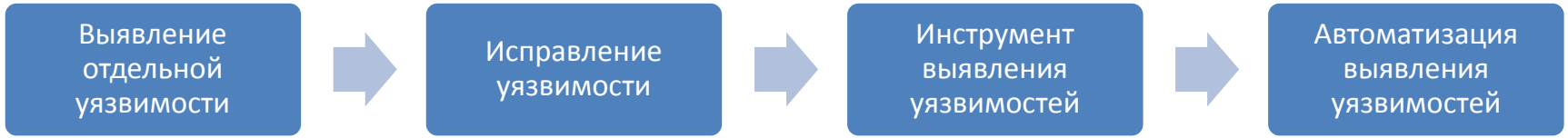
- Bug Bounty Program (BBP) – программа поощрения сторонних специалистов за найденные ошибки в ПО, в первую очередь – эксплуатируемых уязвимостей

Тип	Примеры	Особенности
Системообразующие BBP	Microsoft, Facebook, Google, US DoD	Проводятся непосредственно крупнейшими разработчиками ПО, которые устанавливают порядок выкупа информации об ошибках
Платформы BBP	BugCrowd, HackerOne, TippingPoint	Легальная общая площадка для BBP различных организаций Размер вознаграждения определяется самой площадкой
Частные посреднические BBP	Zerodium, Sonosoft	Выкупает уязвимости с целью перепродажи Размер вознаграждения больше, чем в легальных программах

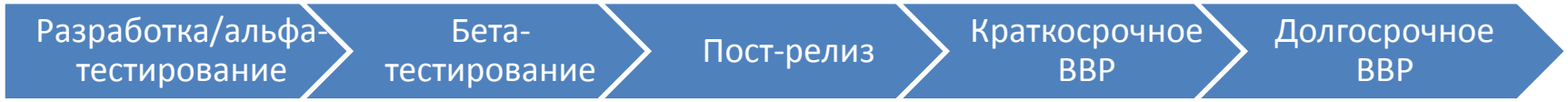
- Размер вознаграждения сильно варьируется
  - Скидочный купон на 12,50$ от Yahoo!
  - До 2 500 000$ за полную цепочку реализации уязвимости в Android смартфоне (с сохранением присутствия) от Zerodium
- BBP – mainstream. Почему?
  - Кратный ROI при правильной организации BBP, снижение репутационных издержек

Bug Bounty Program
Легко сказать, сложно сделать

- Как определить область действия BBP?
  - Отдельный программный продукт / текущая версия продукта
  - Вся линейка версий программного продукта
  - Сторонние библиотеки, ПО поглощенных компаний
- Как организовать долгосрочное использование результатов BBP, т.е. «обогатить» SDL новыми технологиями и практиками?



- На каком этапе жизненного цикла подключать продукт к BBP?

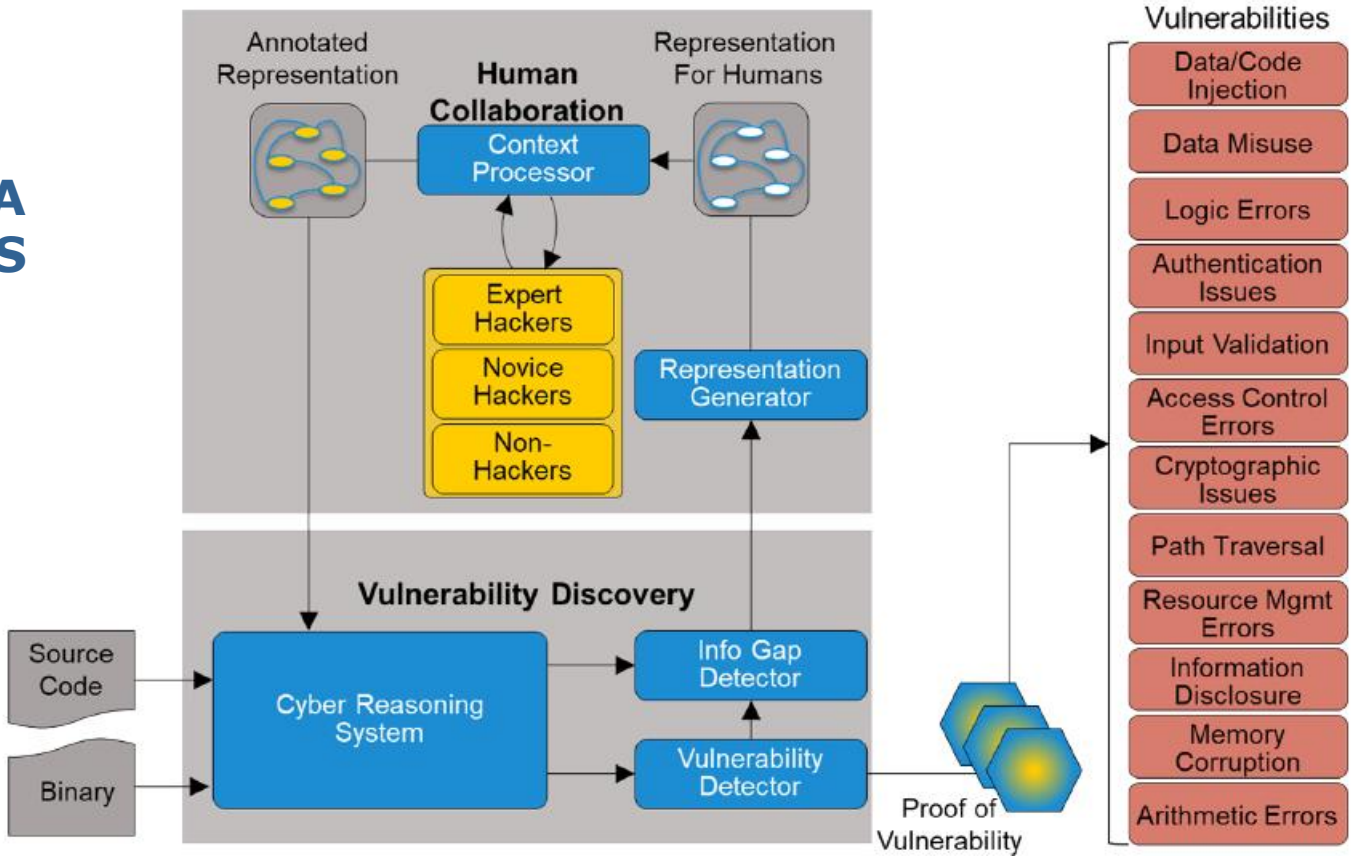


Какую информацию о продукте (документация, система сборки, исходные коды, система тестирования, тесты, результаты тестирования, …) допустимо раскрыть сторонним (или приглашенным) специалистам?
Больше информации $\Rightarrow$ лучше результат + больше рисков утечки информации

- Анализ потока обращений: отсеивание False Positive, соотнесение сбоя с кодом, нахождение уникальных ошибок, оценка критичности ошибки, …

Зарубежный опыт: автоматизированный поиск критических дефектов

DARPA CHESS

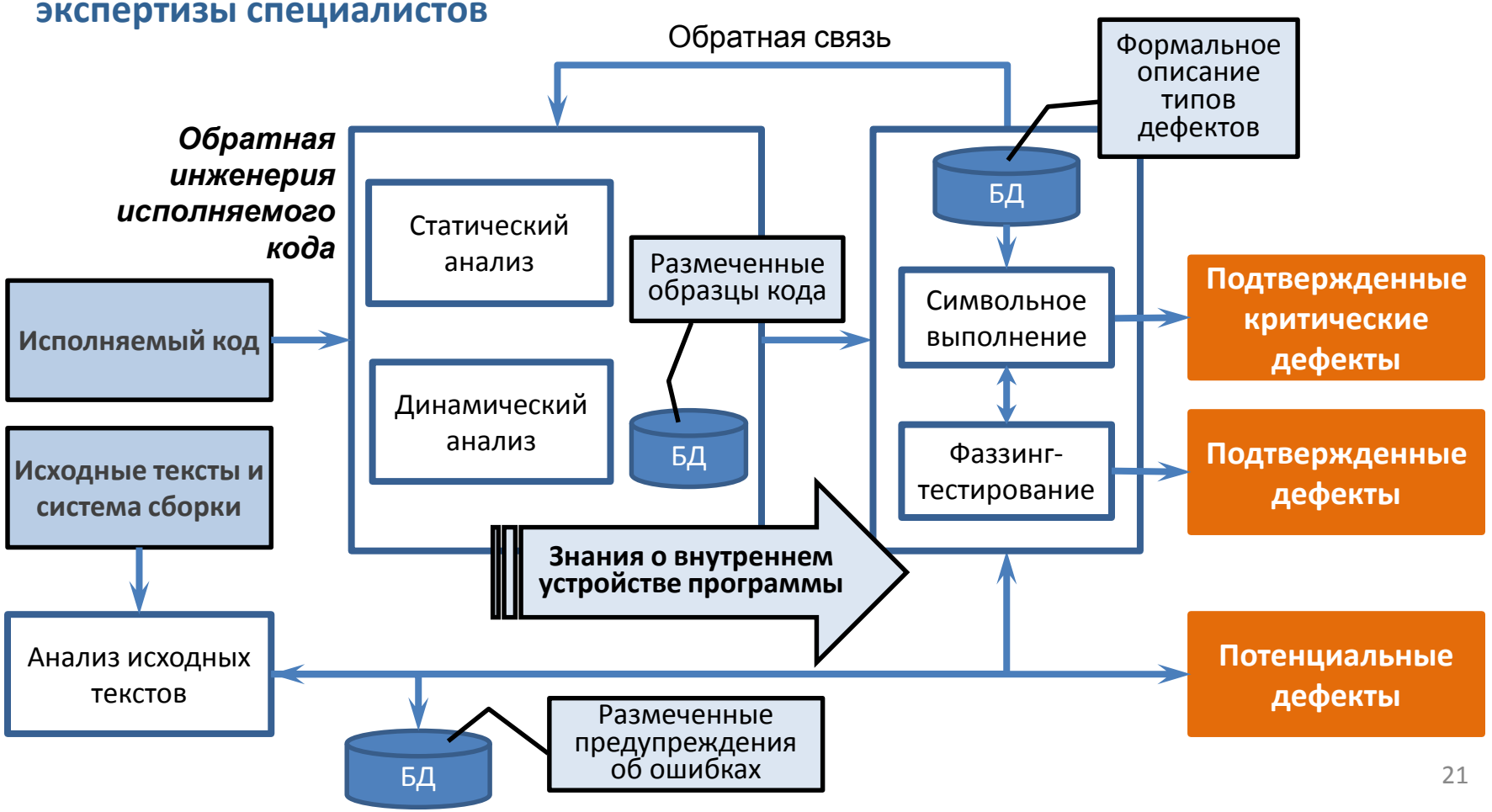


- **DARPA Cyber Grand Challenge** (2016)
  Соревнование по автономному поиску и исправлению критических дефектов в бинарном коде, победитель: Mayhem – Carnegie-Mellon-University

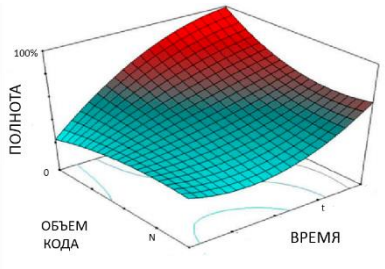
- Проект **DARPA CHESS** (2018, объявлен конкурс)
  Масштабируемый поиск критических дефектов за счет синергии инструментов анализа и экспертных знаний

Необходимый этап SDL: непрерывный поиск дефектов в ПО после сдачи в эксплуатацию

- **Нет ограничения по времени**
- **Требует больших ресурсов: вычислительных мощностей и экспертизы специалистов**



Подтверждение доверия ПО



- Непрерывный анализ безопасности ПО, на выделенных вычислительных мощностях
- Совместное участие разработчиков, регуляторов, науки
- Возможные пилотные проекты: ядро Linux, СУБД, runtime и VM Java, …

Деятельность ИСП РАН

- **Исследования**
  - Ведущая в РФ научная школа в области системного программирования
- **Разработка**
  - НИОКР в области кибербезопасности, в том числе в рамках ГОЗ
  - Инструменты для разработки безопасного ПО
  - Большие данные и анализ социальных сетей
  - …
- **Обучение**
  - Кафедры в ведущих университетах
  - Обучение для сотрудников Испытательных Лабораторий (ФСТЭК России)
- **Сотрудничество**
  - Соглашение о сотрудничестве между ФСТЭК России и ИСП РАН
  - ИСП РАН возглавляет Подкомитет 4 «Разработка безопасного программного обеспечения» в ТК 362 «Защита информации»

ИСП РАН

Спасибо за внимание
Вопросы

Разработка доверенного ПО
Вызовы и перспективы

ПАДАРЯН В.А.
vartan@ispras.ru