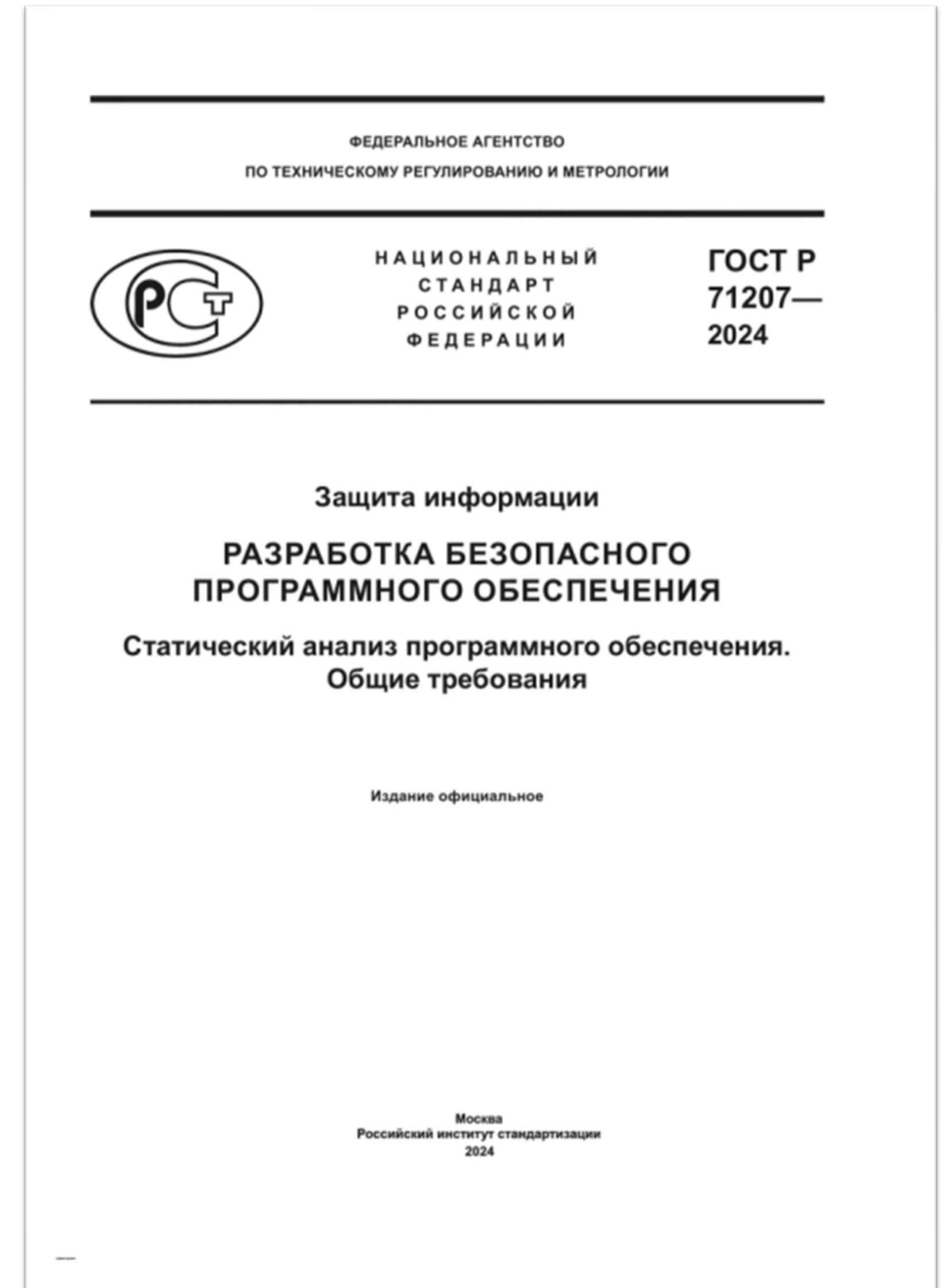


Регулярный статистический анализ

ГОСТ Р 71207-2024

Устанавливает общие требования к внедрению и выполнению статического анализа ПО, а также исходные данные, необходимые для его выполнения.



Проблематика

Накопление непроанализированных изменений приводит к:



Проблематика

Накопление непроанализированных изменений приводит к:

- ухудшению качества проводимой экспертизы результатов анализа;



Проблематика

Накопление непроанализированных изменений приводит к:

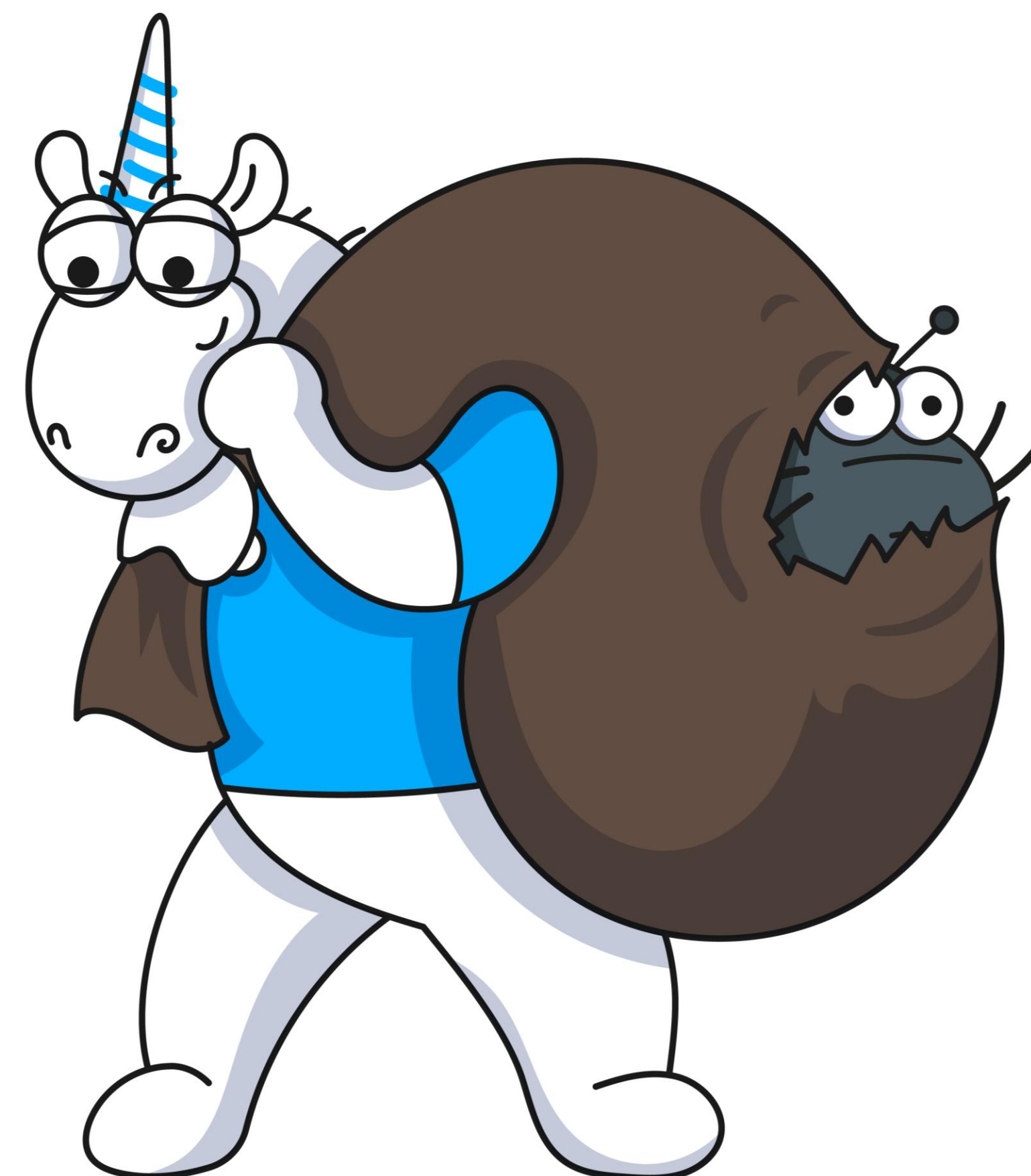
- ухудшению качества проводимой экспертизы результатов анализа;
- увеличению длительности проводимой экспертизы;



Проблематика

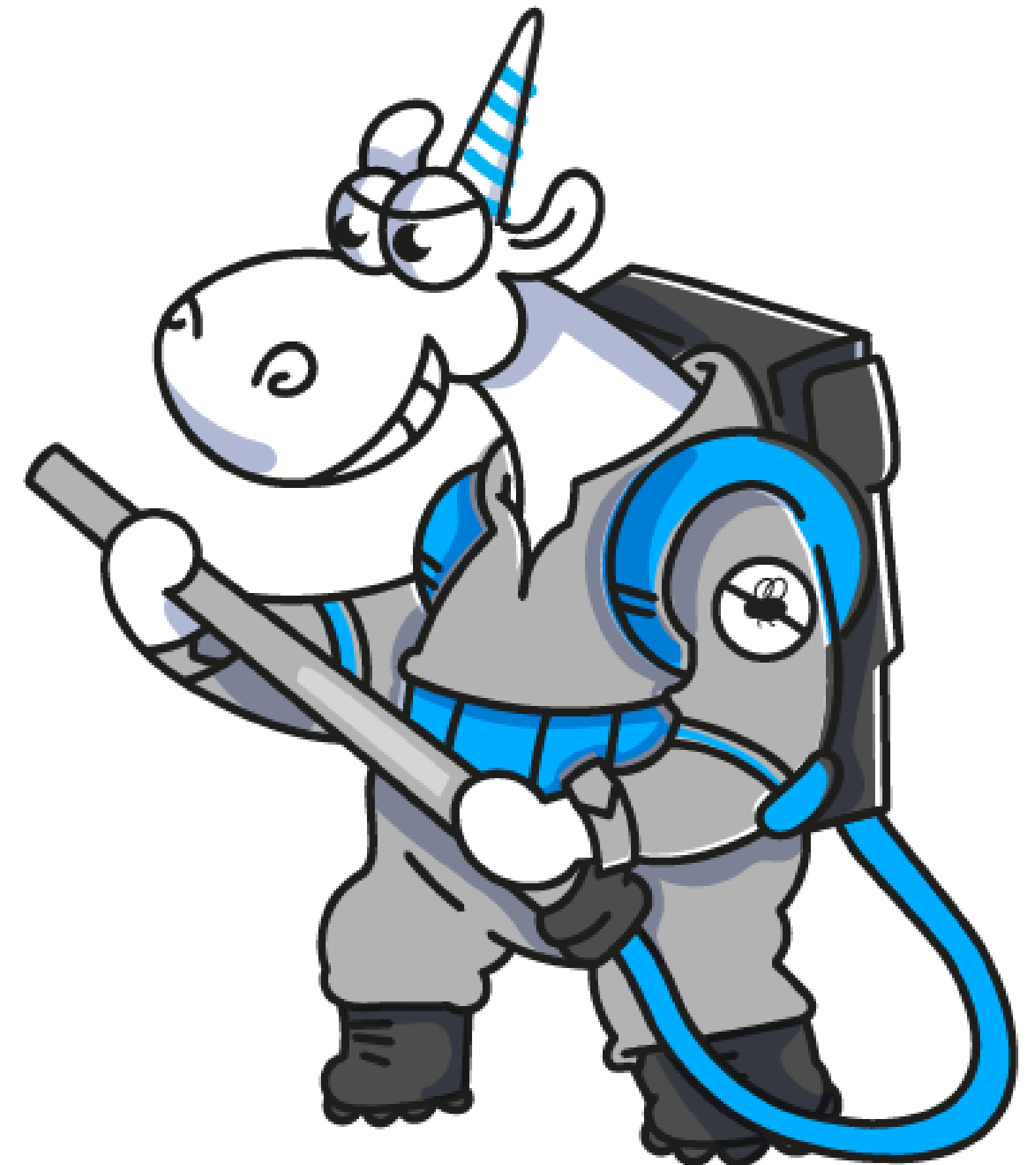
Накопление непроанализированных изменений приводит к:

- ухудшению качества проводимой экспертизы результатов анализа;
- увеличению длительности проводимой экспертизы;
- усложнению исправления ошибок.



ГОСТ Р 71207-2024, п. 5.6

«Для своевременного выявления и исправления ошибок статический анализ **должен** регулярно применяться к разрабатываемому ПО»



ГОСТ Р 71207-2024, пп. 5.5 и 5.8

Анализ:

- Для всего кода проводить не реже, чем раз в 10 дней после внесения изменений.
- Для нового кода проводить сразу после внесения изменений.



ГОСТ Р 71207-2024, пп. 5.5 и 5.8

Анализ:

- Для всего кода проводить не реже, чем раз в 10 дней после внесения изменений.
- Для нового кода проводить сразу после внесения изменений.

Просмотр и разметка предупреждений:

- Для нового кода не позднее, чем через 3 дня после анализа
- Для всего кода не позднее, чем через 10 дней после анализа



Best Practices

Примерно тех же требований к регулярности придерживаются большие компании.

«Lessons from Building Static Analysis Tools at Google»

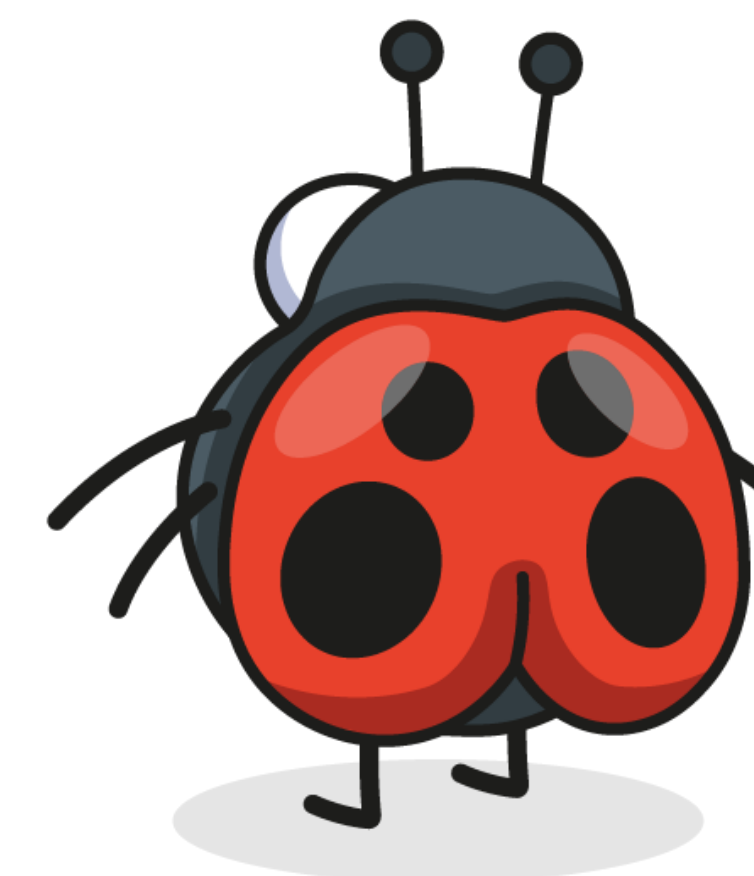


Критические ошибки

Выбираем правила

Для компилируемых языков:

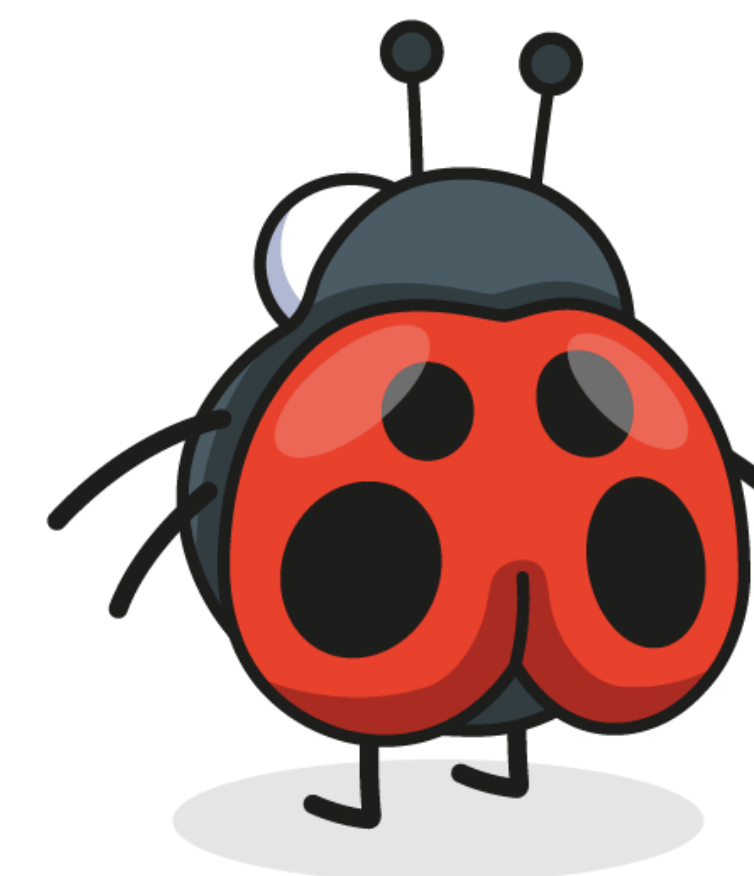
- ошибки непроверенного использования чувствительных данных;



Выбираем правила

Для компилируемых языков:

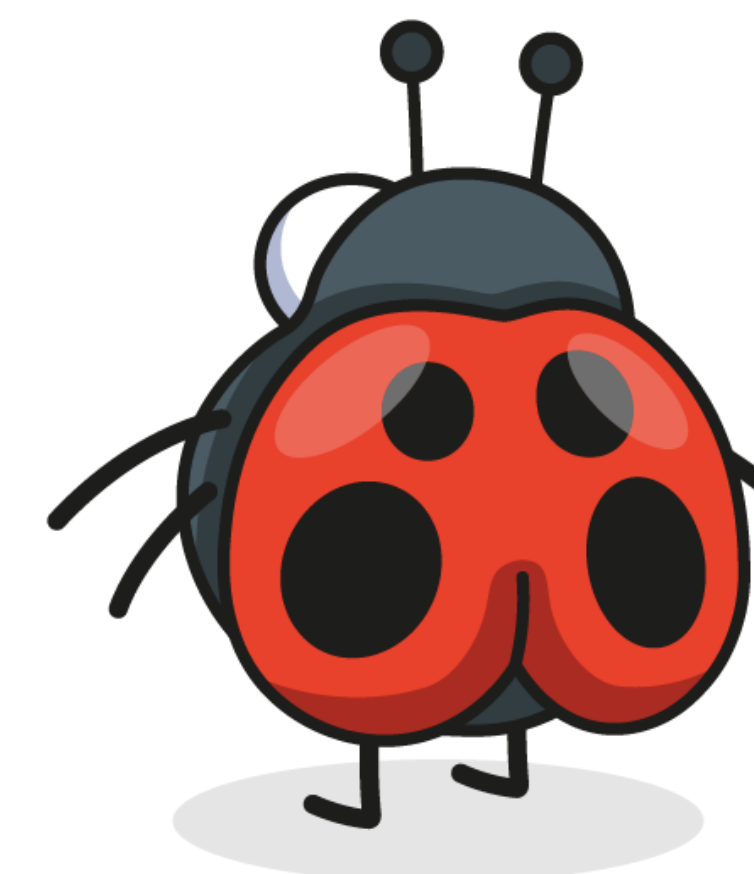
- ошибки непроверенного использования чувствительных данных;
- ошибки целочисленного переполнения и некорректного совместного использования знаковых и беззнаковых чисел;



Выбираем правила

Для компилируемых языков:

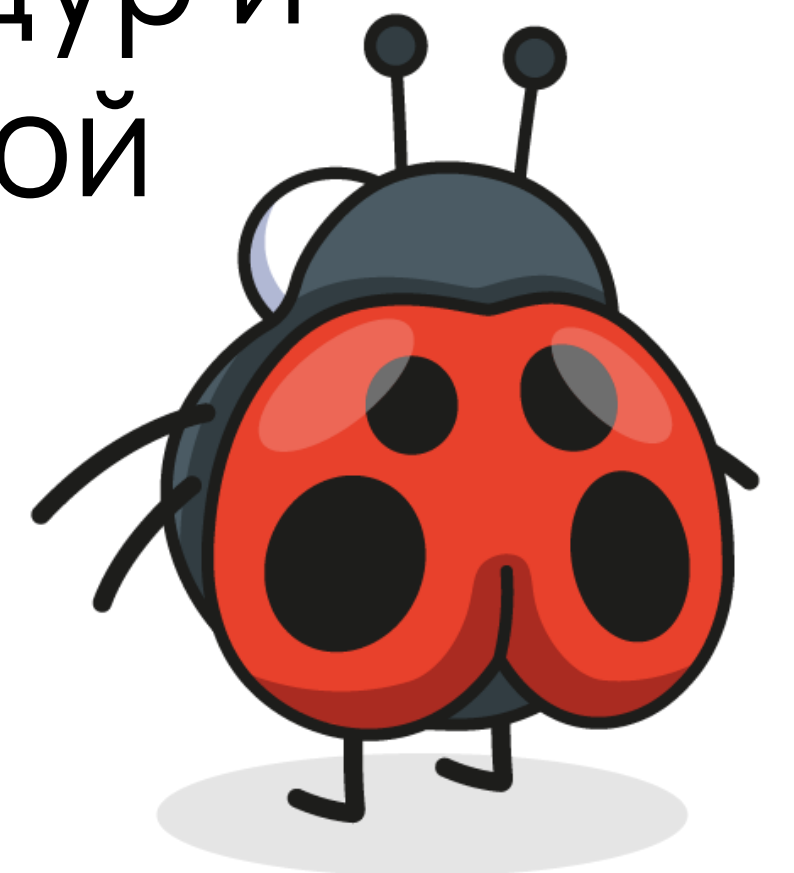
- ошибки непроверенного использования чувствительных данных;
- ошибки целочисленного переполнения и некорректного совместного использования знаковых и беззнаковых чисел;
- ошибки переполнения буфера;



Выбираем правила

Для компилируемых языков:

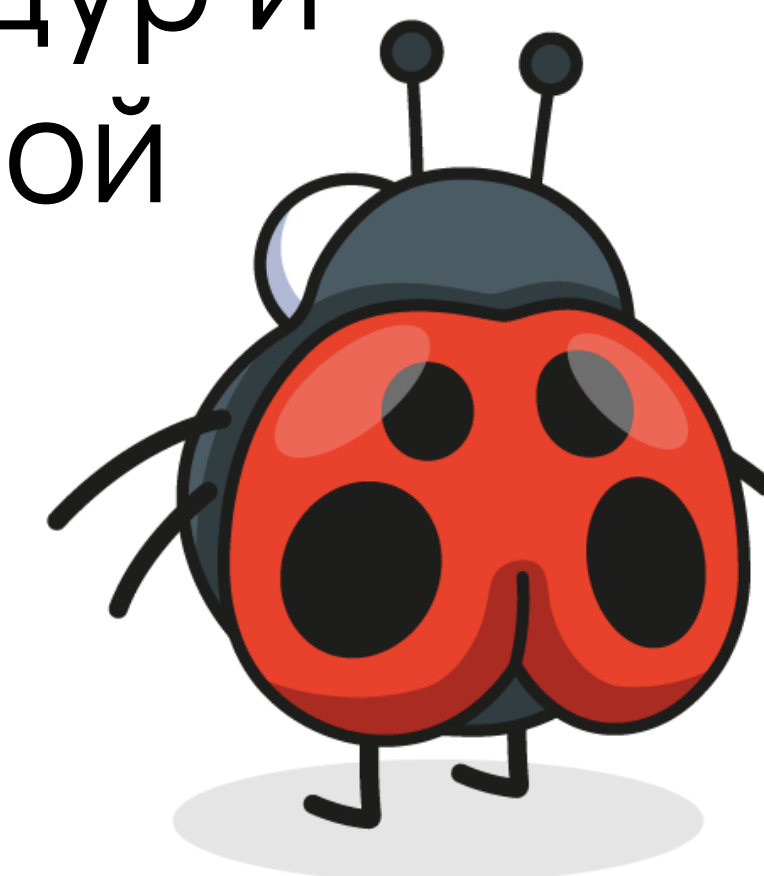
- ошибки непроверенного использования чувствительных данных;
- ошибки целочисленного переполнения и некорректного совместного использования знаковых и беззнаковых чисел;
- ошибки переполнения буфера;
- ошибки некорректного использования системных процедур и интерфейсов, связанных с обеспечением информационной безопасности;



Выбираем правила

Для компилируемых языков:

- ошибки непроверенного использования чувствительных данных;
- ошибки целочисленного переполнения и некорректного совместного использования знаковых и беззнаковых чисел;
- ошибки переполнения буфера;
- ошибки некорректного использования системных процедур и интерфейсов, связанных с обеспечением информационной безопасности;
- ошибки при работе с многопоточными примитивами.



Security Related Issues

- ☒ Append SAST identifiers to security related issues in the analyzer output

This would apply after the next analyzer run after this setting is set to true.

SAST ▼

SEC-TAINT

SEC-SYNCH
RONIZATIO
N

SEC-SYNCH
RONIZATIO
N

SEC-SYNCH
RONIZATIO
N

SEC-SYNCH
RONIZATIO
N

SEC-SYNCH
RONIZATIO
N



У меня локально работает!

Первый этап регулярного анализа – запуск на машинах разработчиков.



У меня локально работает!

Первый этап регулярного анализа – запуск на машинах разработчиков.

Как быстро проанализировать внесённые разработчиком изменения?



Инкрементально!

- Инкрементальный анализ позволяет анализировать только внесённые изменения



Инкрементально!

- Инкрементальный анализ позволяет анализировать только внесённые изменения
- Анализатор при сборке проекта получает информацию о том, какие файлы менялись, и анализирует их



Инкрементально!

- Инкрементальный анализ позволяет анализировать только внесённые изменения
- Анализатор при сборке проекта получает информацию о том, какие файлы менялись, и анализирует их

Таким образом можно проверять изменения локально, не тратя на это много времени.



Автоматизируем

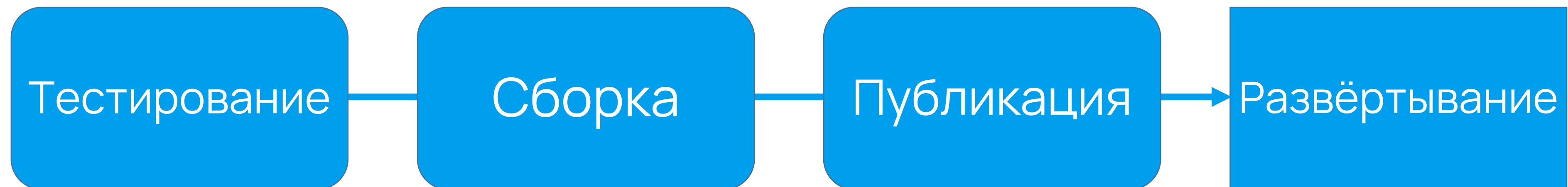
ГОСТ Р 71207-2024:

«Регулярность статического анализа ПО обеспечивается автоматизацией процедуры проведения анализа согласно требованиям 5.6—5.8, например с помощью системы непрерывной интеграции»



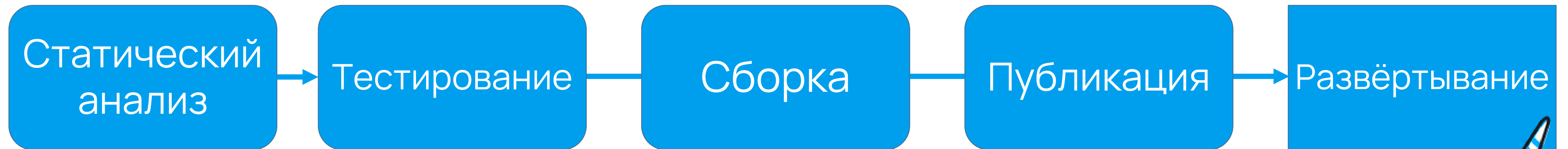
CI/CD + статический анализ

Не хватает одной маленькой детали...



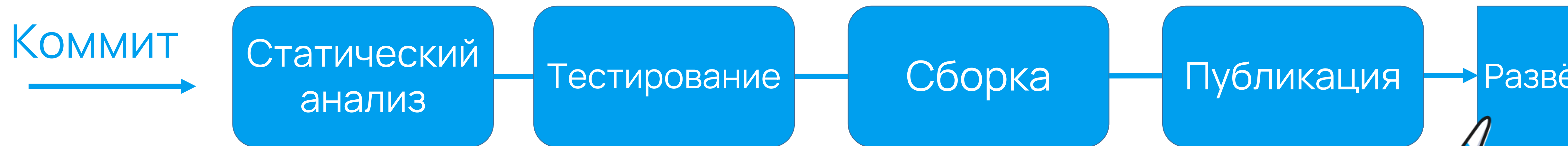
CI/CD + статический анализ

Уже лучше...

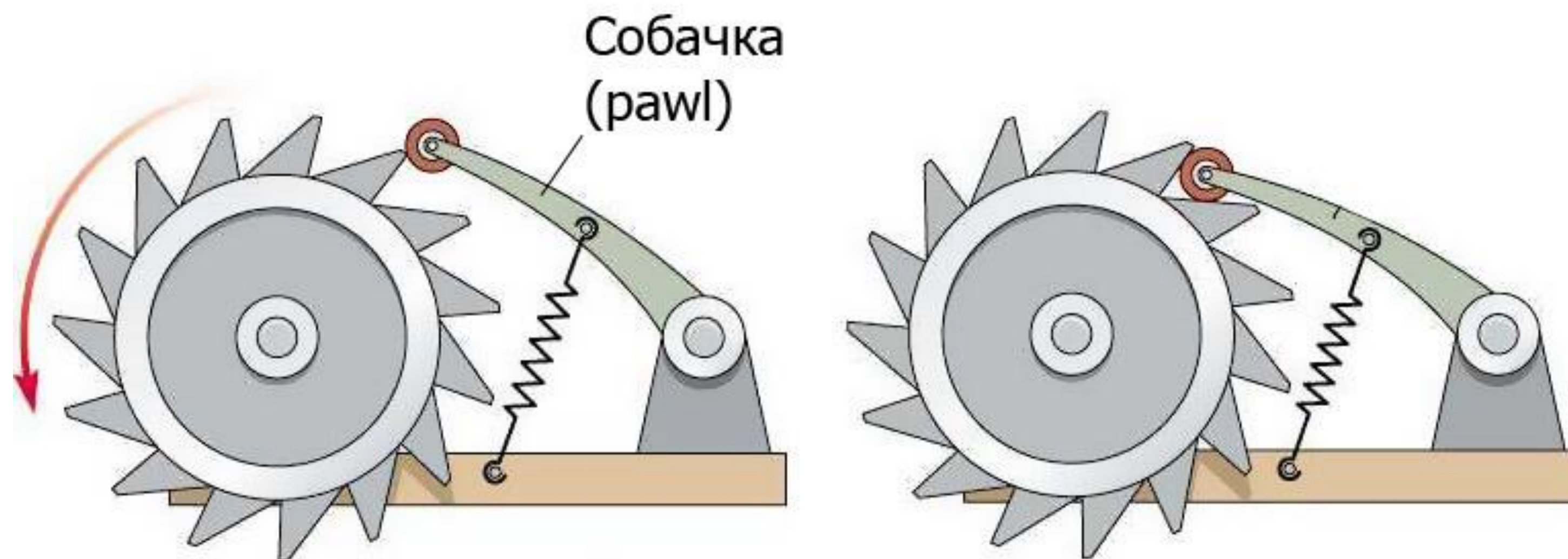


CI/CD + статический анализ

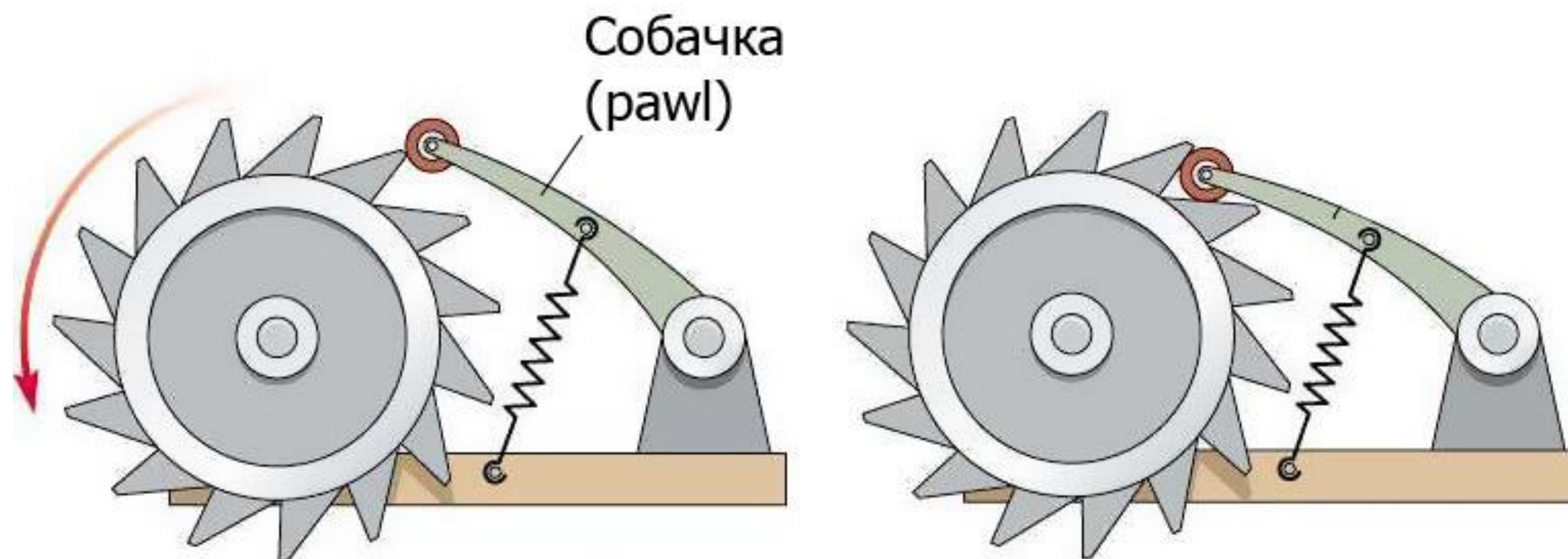
Вот так вот!



Храповик: качество на максимум



Храповик: качество на максимум



Интегрируем новый код только если в нём нет новых критических ошибок.

Работаем с результатами

- Используем различные инструменты для удобной работы с результатами анализа.



Работаем с результатами

- Используем различные инструменты для удобной работы с результатами анализа.
- Таким образом можно и распределить работу по разметке и исправлению срабатываний в команде



БЛАГОДАРИМ ЗА ВНИМАНИЕ!



InSeq.RBPO Легкий путь внедрения
практик безопасной разработки



PVS-Studio 30 дней бесплатно по
промокоду RBPO_2025