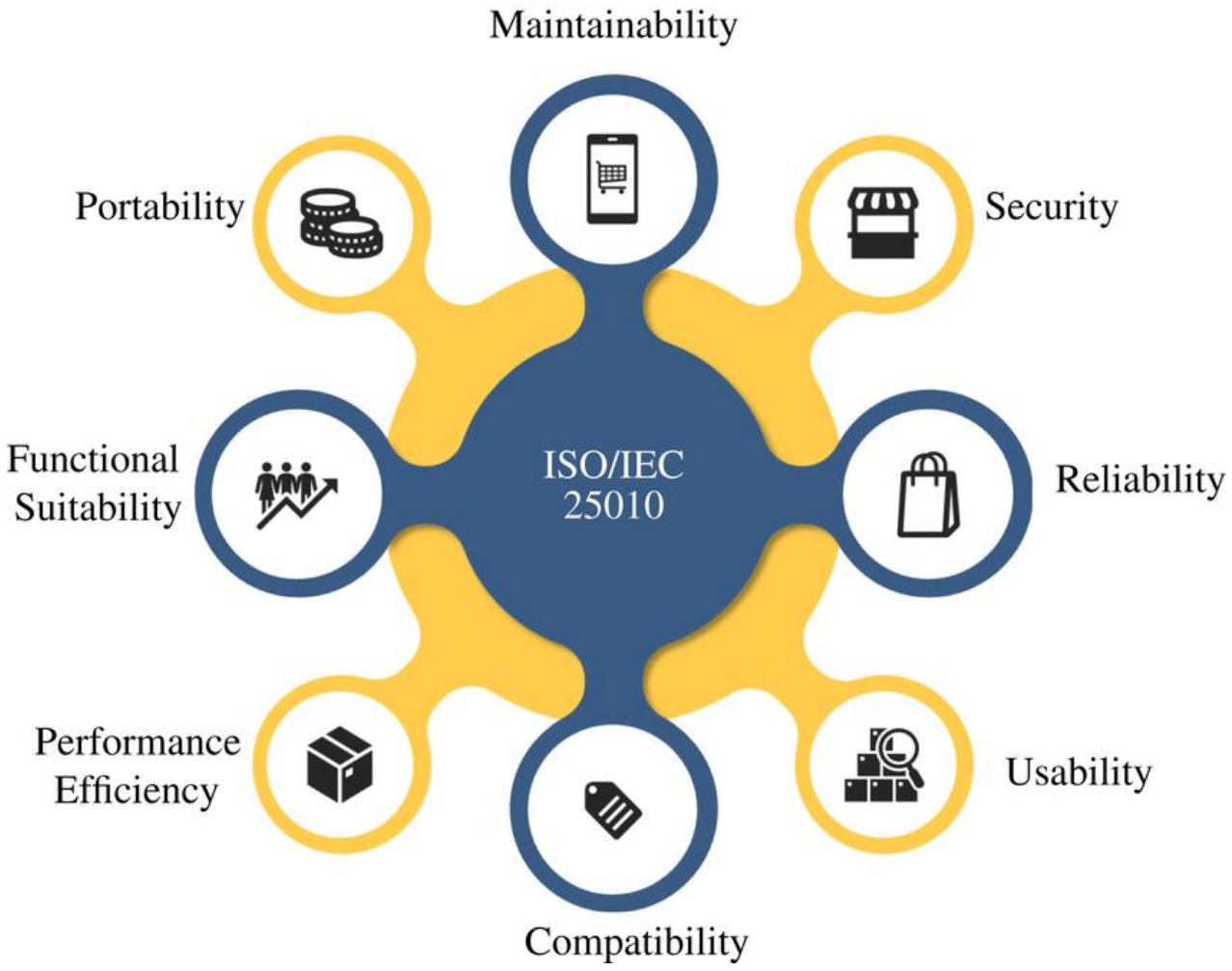
RUDN university

НАДЕЖНОСТЬ **ПО**

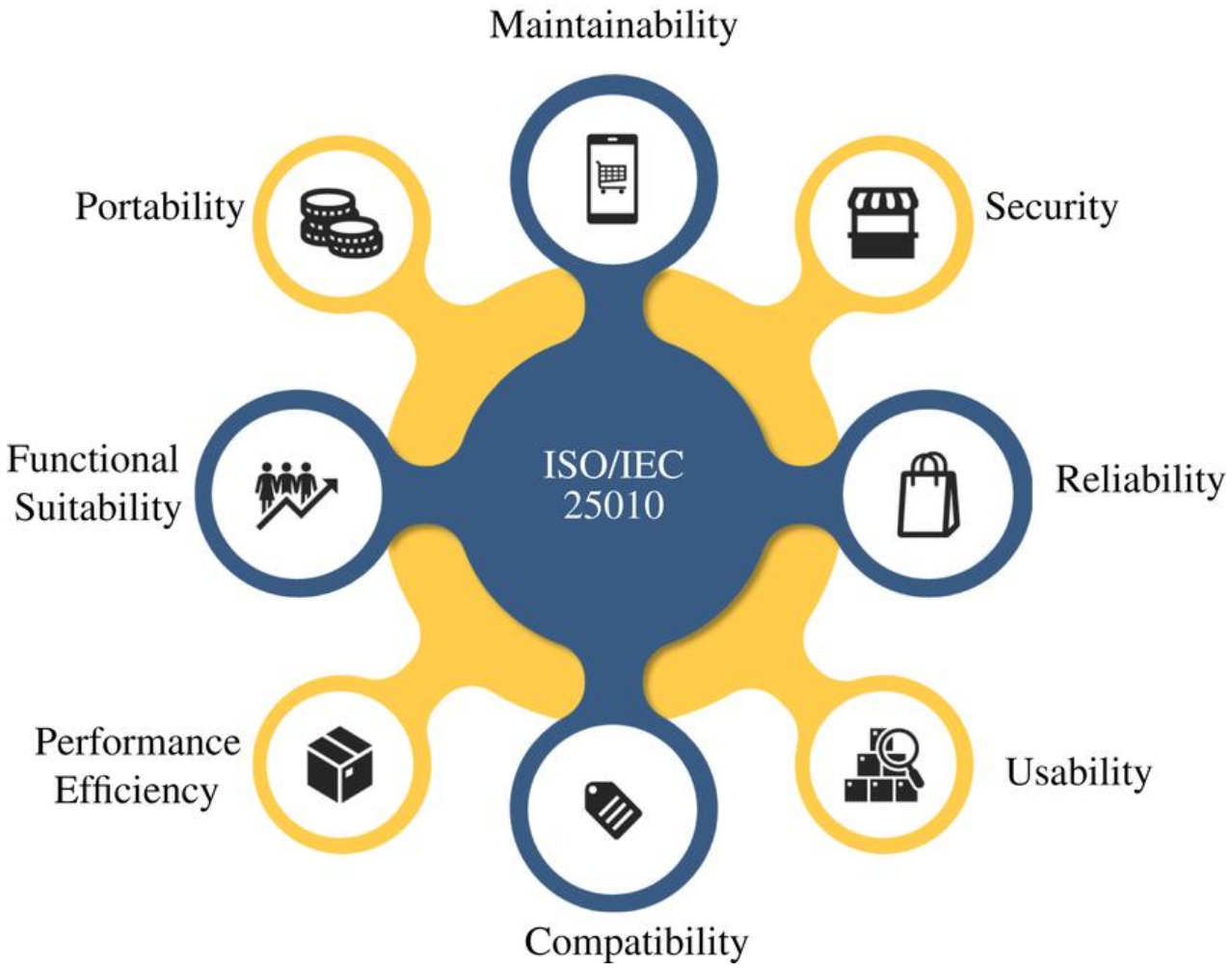
ЛОГИРОВАНИЕ И МОНИТОРИНГ

ISO/IEC 25010



ISO/IEC 25010 — это международный стандарт, предоставляющий всеобъемлющую и структурированную модель для определения и оценки качества программного обеспечения и систем.

ISO/IEC 25010



1. Функциональность (Functional suitability)

Способность программного обеспечения предоставлять функции, которые удовлетворяют явные и неявные потребности пользователя при использовании программного обеспечения в заданных условиях.

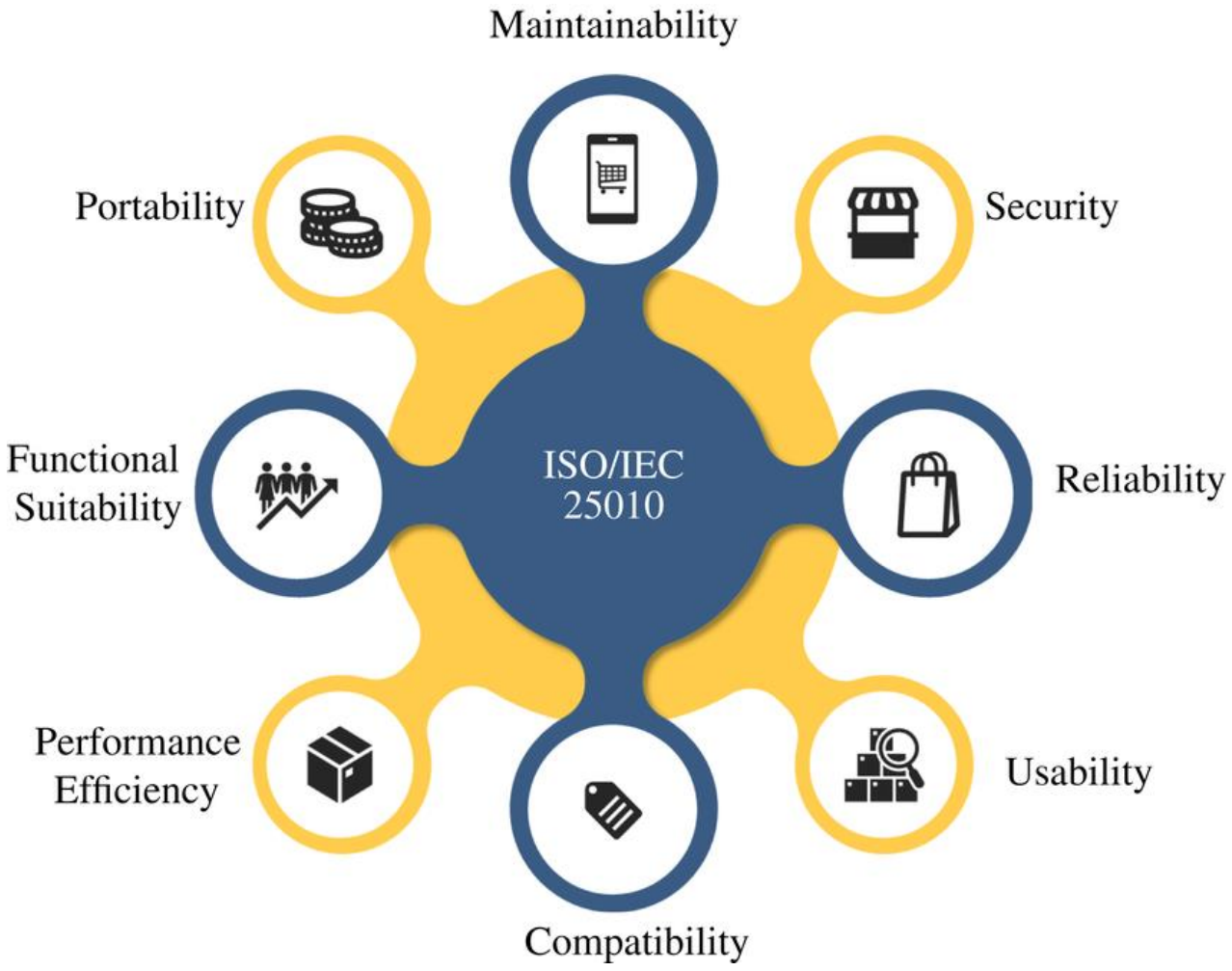
- Функциональная полнота
- Функциональная корректность
- Функциональная уместность

2. Надежность (Reliability)

Способность системы или компонента выполнять требуемые функции при заданных условиях в течение заданного периода времени.

- Зрелость
- Доступность
- Отказоустойчивость
- Восстанавливаемость

ISO/IEC 25010



3. Производительность (Performance efficiency)

Способность продукта предоставлять соответствующую производительность относительно количества используемых ресурсов при заданных условиях.

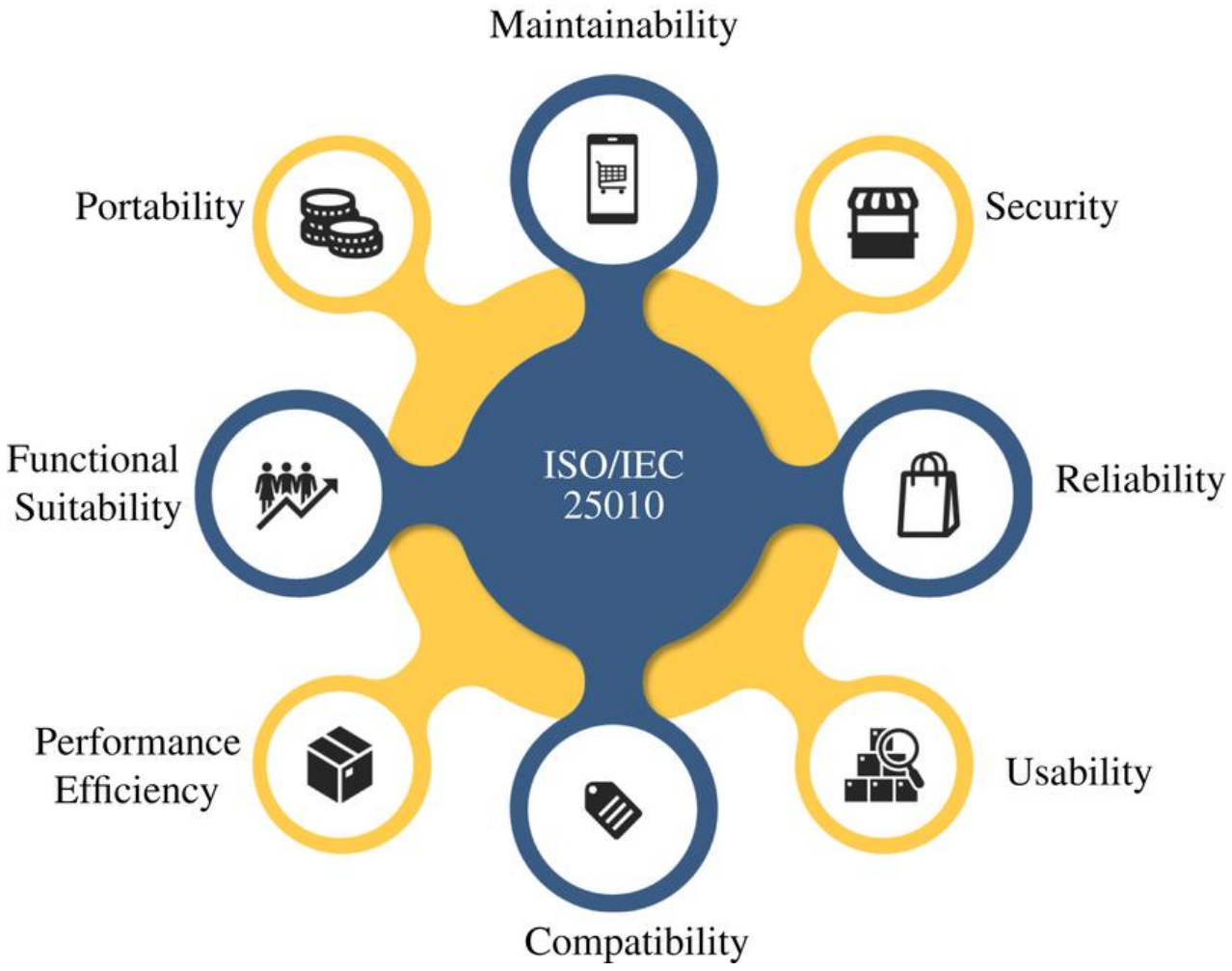
- Временная эффективность
- Использование ресурсов
- Емкость

4. Совместимость (Compatibility)

Способность двух или более систем или компонентов обмениваться информацией и/или выполнять требуемые функции, будучи использованными вместе.

- Сосуществование
- Взаимозаменяемость

ISO/IEC 25010



5. Удобство использования (Usability)

Степень, с которой продукт может быть использован определенными пользователями в определенном контексте использования для достижения поставленных целей с эффективностью, продуктивностью и удовлетворенностью.

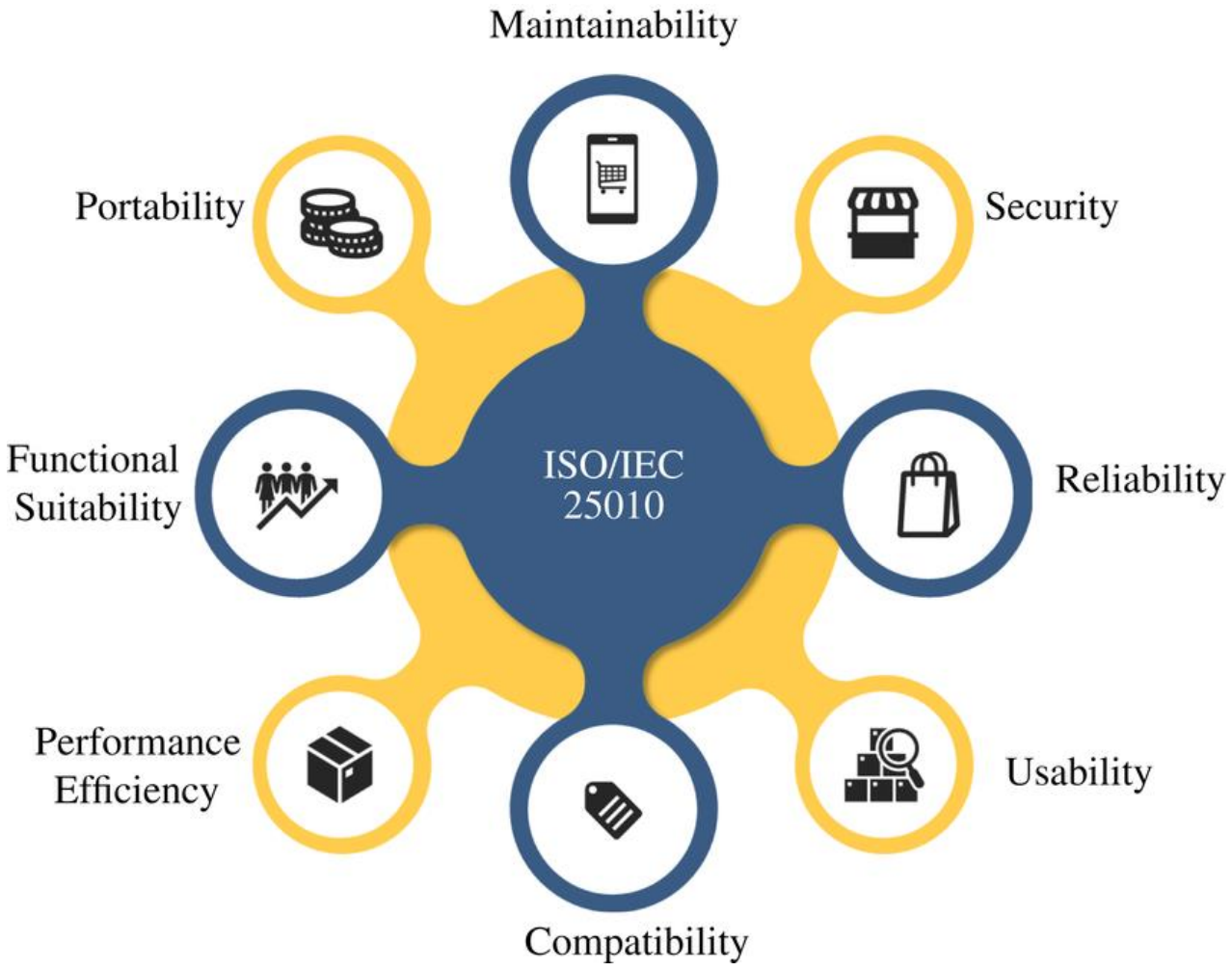
- Пригодность для узнавания
- Обучаемость
- Защита от ошибок пользователя
- Доступность

6. Безопасность (Security)

Способность продукта защищать информацию и данные так, чтобы обеспечить необходимый уровень доступа для authorized пользователей, и предотвратить доступ неавторизованных лиц и систем.

- Конфиденциальность
- Целостность
- Подотчетность
- Аутентичность

ISO/IEC 25010



7. Сопровождаемость (Maintainability)

Эффективность и результативность, с которыми продукт может быть модифицирован для улучшения, исправления или адаптации к изменениям в окружении и требованиях.

- Модульность
- Анализируемость
- Тестируемость

8. Переносимость (Portability)

Степень эффективности и результативности, с которой система, продукт или компонент может быть перенесен из одного аппаратного, программного или операционного окружения в другое.

- Адаптируемость
- Устанавливаемость
- Заменяемость

SRE

Site Reliability Engineering (SRE) — это практика использования программных инструментов для автоматизации задач ИТ-инфраструктуры, таких как управление системой и мониторинг приложений.

Ключевые принципы:

- Мониторинг приложений
- Постепенное внедрение изменений
- Автоматизация для повышения надежности

МЕТРИКИ **SLI, SLO, SLA**

SLI

SLI (Service Level Indicator) — это числовой показатель, который отражает, насколько хорошо система работает с точки зрения пользователя.

Например: доля успешных запросов, среднее время ответа API, процент времени доступности сервиса.

способ замерить производительность

SLO

SLO (Service Level Objective) — это целевое значение для SLI, которого команда стремится достичь.

Например: «99.9% успешных запросов за последние 30 дней».

ориентир, цель, к которой стремится система

SLA

SLA (Service Level Agreement) — это юридическое или формализованное соглашение между заказчиком и поставщиком услуги.

Например: «Если аптайм будет ниже 99.5% — клиент получает скидку».

юридическая ответственность за достижение целей

МЕТРИКИ **SLI, SLO, SLA**: ПРИМЕНЕНИЕ В КЕЙСАХ

1. Облачная платформа предоставляет API для финтех-приложений

SLI: из 10 миллионов запросов в сутки, 9.996 миллиона успешны — это 99.96% доступности.

SLO: целевое значение — 99.9%. Порог не превышен.

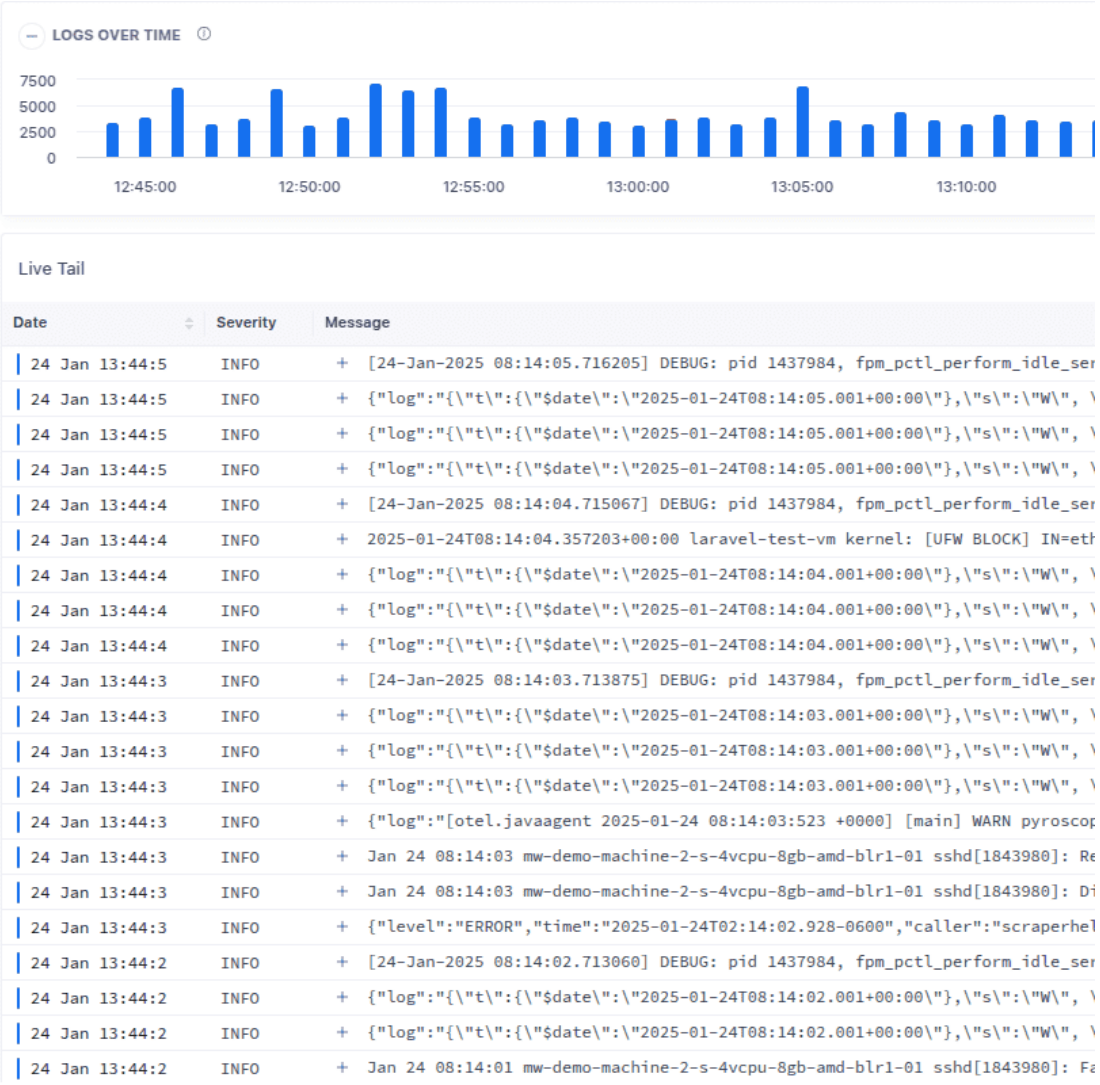
SLA: если доступность ниже 99.5%, клиенту предоставляется компенсация (прописано в контракте).

В этом кейсе всё укладывается в рамки. Команда может спокойно выкатывать обновления. Но если бы доступность опустилась до 99.3% — начался бы инцидент, штраф, возможно — мораторий на релизы.

2. Видеостриминговый сервис

Их **SLI** — доля пользователей, у которых видео воспроизводится без буферизации. При снижении этого показателя под **SLO**-порог (например, 98%) система активирует аварийный механизм: автоматически понижает качество видео, чтобы уменьшить нагрузку. Применение SLO помогает автоматизировать реакцию системы на падение качества, снижая ущерб пользователю ещё до того, как он пожалуется.

ЛОГИРОВАНИЕ



Логирование (журналирование) — это стратегический инструмент для диагностики, мониторинга и обеспечения безопасности приложений.

Достаточность логирования

Эффективное логирование должно балансировать между информативностью и лаконичностью.

УРОВНИ ЛОГИРОВАНИЯ

FATAL — дальнейшее исполнение программы невозможно.

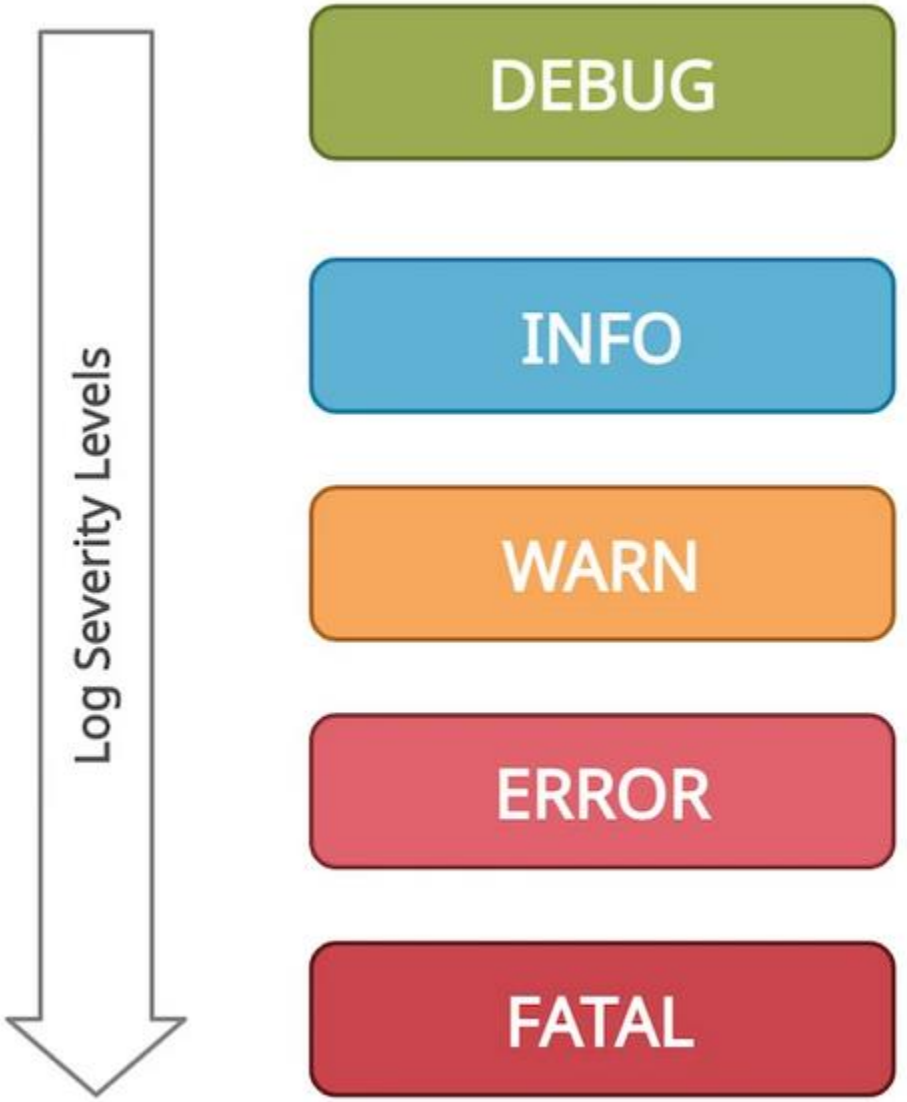
ERROR — отказы внешних систем, при которых программа может выполнять свою работу дальше.

WARN — поведение программы, которое требует внимания.

INFO — шаги бизнес-логики, включая ожидаемые ошибки.

DEBUG — более подробная, расширенная информация о шагах бизнес-логики, подробные шаги доступа во внешние системы.

TRACE — дополнительная информация к DEBUG.



> *Уровень логирования в production обычно устанавливается на WARN или INFO, в то время как на стендах разработки может использоваться DEBUG.*

КОНТЕКСТ ЛОГИРОВАНИЯ

```
2025-02-03 13:00:01 ERROR Отказ в репозитории
2025-02-03 13:00:01 INFO  Сохранение прошло успешно
```

Проблема

Нехватка информации для полного понимания работы приложения

VS

```
2025-06-15T08:15:30.123456789Z INFO  |b566a678| ProductRepository :
Регистрация продукта id=36
2025-06-15T08:15:30.987654321Z ERROR |d346f325| UserRepository :
Редактирование пользователя id=34. Ошибка AUTH-25
```

Решение

1. Когда случилось событие (2025-06-15T08:15:30.123456789Z).

2. Сквозной ID операции. Это особенно важно, если операция составная (|b566a678|).

3. Место, где произошло событие: класс, компонент и т. п. (ProductRepository).

4. Что именно произошло и какой контекст события (Редактирование пользователя id=34. Ошибка AUTH-25).

МАСКИРОВАНИЕ ЛОГОВ

```
2025-06-15T08:15:30.123456789Z INFO Запрос на обработку платежа.
Клиент: "Иванов Иван Петрович". Дата рождения: "15.03.1985". Email:
"ivanov@mail.ru". Номер карты: "1234 5678 9012 3456". CVV код карты:
"123". Сумма платежа: 1500.00 руб.
```

Проблема

Запись в логи конфиденциальной, либо персональной информации

VS

```
2025-06-15T08:15:30.123456789Z INFO Запрос на обработку платежа.
Клиент: "Иванов И.П.". Дата рождения: "**.**.1985". Email: "i***@mail.
ru". Номер карты: "**** **** **** 3456". CVV код карты: "***". Сумма
платежа: 1500.00 руб.
```

Решение

Автоматическая маскировка чувствительных полей с помощью подходящих методов или библиотек логирования

ПЕРИОДИЧЕСКИЕ ЗАДАЧИ

```
2025-02-03 13:01:01 INFO Обработано 0 заданий
2025-02-03 13:02:01 INFO Обработано 0 заданий
2025-02-03 13:03:01 INFO Обработано 0 заданий
2025-02-03 13:04:01 INFO Обработано 0 заданий
```

Проблема

Часто исполняемая задача, по расписанию или по событию

VS

```
2025-06-15T10:15:30.123456789Z INFO JobSchedule : За период
с 2025-06-15T08:15:30 по 2025-06-15T10:15:30 обработано
успешно 135 заданий, неудачно 3
2025-06-15T12:15:30.123456789Z INFO JobSchedule : За период
с 2025-06-15T10:15:30 по 2025-06-15T12:15:30 обработано
успешно 56 заданий, неудачно 0
```

Решение

Увеличить период записи в лог и дополнить статистической информацией

КОДЫ ОШИБОК/ОПЕРАЦИЙ

```
2025-06-15T10:15:30.123456789Z INFO  |b566a678| AuthModule:
Ошибка ввода неверного логина или пароля пользователя,
пользователь ivanov, неверный пароль
2025-06-15T10:15:30.123456789Z INFO  |e6575098| AuthModule:
Ошибка - пользователь petrov заблокирован
2025-06-15T12:15:30.123456789Z INFO  |c987a465| AuthModule:
Ошибка ввода неверного логина или пароля пользователя,
пользователь semenov, неверный пароль
```

Проблема

При генерации большого объёма логов даже небольшие оптимизации длины записи способны ощутимо сократить затраты на хранение. В логах часто встречаются повторы.

VS

```
2025-06-15T10:15:30.123456789Z INFO  |b566a678| AuthModule:
AUTH-ERR-1, ID=ivanov
2025-06-15T10:15:30.123456789Z INFO  |e6575098| AuthModule:
AUTH-ERR-4, ID=petrov
2025-06-15T12:15:30.123456789Z INFO  |c987a465| AuthModule:
AUTH-ERR-1, ID=semenov
```

Решение

Ввести коды ошибок (описанные в документации), который позволят кратко описать ситуацию

ОДНОТИПНЫЕ СООБЩЕНИЯ

```
2025-06-15T08:15:30.123456789Z INFO |b566a678| Worker :
Обработана заявка id=56736
2025-06-15T08:15:30.987654321Z INFO |b566a678| Worker :
Обработана заявка id=32453
2025-06-15T08:15:31.457457457Z INFO |b566a678| Worker :
Обработана заявка id=58568
2025-06-15T08:15:31.567564354Z INFO |b566a678| Worker :
Обработана заявка id=23552
```

Проблема

Много мелких однотипных сообщений

VS

```
2025-06-15T08:15:30.123456789Z INFO |b566a678| Worker :
За период с 2025-06-15T08:15:30 по 2025-06-15T08:15:31
обработаны заявки: 56736, 32453...
2025-06-15T08:15:30.123456789Z INFO |b566a678| Worker :
За период с 2025-06-15T08:15:31 по 2025-06-15T08:15:32
обработаны заявки: 23552, 23523...
```

Решение

Ограничивать размер записи лога и формировать пачки ID заявок заранее определённого, конечного размера

ЛОГИРОВАНИЕ: КЛЮЧЕВЫЕ ПРИНЦИПЫ

1. СТРУКТУРИРОВАННОСТЬ

- Подходящий формат

- Стандартизированные поля (timestamp, level, message, context)

- Упрощение автоматической обработки и анализа

2. БАЛАНС ИНФОРМАТИВНОСТИ

- Контекст каждой ошибки должен быть достаточным для диагностики

- Избегание избыточности и «шума» в логах

3. ПРОИЗВОДИТЕЛЬНОСТЬ

- Запись в отдельном потоке

- Оптимизация формата

4. БЕЗОПАСНОСТЬ

- Маскирование чувствительных данных

- Отсутствие учетных данных и персональных данных в открытом виде

МОНИТОРИНГ

Мониторинг — это процесс непрерывного сбора, анализа и визуализации количественных данных о состоянии и производительности работающей системы с целью обнаружения инцидентов и принятия оперативных решений.

МОНИТОРИНГ: АРХИТЕКТУРА

Основные компоненты мониторинга

- Средства сбора и отправки данных
- Транспортировка и буферизация
- Хранилище данных
- Инструмент обработки и аналитики
- Инструмент визуализации
- Система оповещения

МОНИТОРИНГ: МОДЕЛИ СБОРА ДАННЫХ

PULL-модель — основной модуль системы мониторинга сам обращается к целевым узлам и забирает с них нужные метрики.

Инициатор – Сервер мониторинга

Действие – Активно запрашивает данные с целевых систем

Аналогия – Проверка нами почты (сами заходим и смотрим есть ли новые письма)

Примеры систем: Prometheus, Zabbix, Nagios

PUSH-модель — агенты собирают метрики с нужных компонентов системы и отправляют в ядро системы мониторинга.

Инициатор – Агенты на целевых системах

Действие – Активно отправляют данные на сервер мониторинга

Аналогия – Получение писем (почтальон сам приносит нам письма)

Примеры систем: StatsD, Graphite

МОНИТОРИНГ: ВИДЫ И КЛЮЧЕВЫЕ ПРИНЦИПЫ

Прикладные метрики

1. **Задержки**

Показывают, насколько медленными являются самые неудачные запросы

2. **Ошибки**

Проблемы в прикладном коде

3. **Нагрузка**

Нагрузка в запросах в секунду

4. **Утилизация**

Насколько система близка к пределу своих возможностей

Бизнес-метрики

Количество успешных заказов, регистраций, отправленных сообщений — эти метрики не только показывают здоровье системы, но и помогают оценить влияние сбоя на бизнес.