

## **Часть 1. Универсальные требования и правила создания дашбордов**

Ниже десять фундаментальных правил. Для каждого я добавил развёрнутую мотивацию, практические советы и типичные ловушки.

### **Определите единственную цель панели.**

Пока не сформулирована главная бизнес-задача («оператор поддержки ловит инциденты быстрее 5 мин»), визуализация превращается в склад случайных метрик. Запишите user-story, согласуйте с заказчиком и держите перед глазами в ходе всей разработки.

Ловушка: попытка «угодить всем» и запихнуть продажи, маркетинг и DevOps на один экран.

### **Знайте свою аудиторию.**

СТО интересуют р99-задержки, а менеджера продукта — конверсии.

Поговорите с 3-5 представителями роли, выясните их словарь и решение, которое они принимают по данным.

Ловушка: показывать бизнес-людям термины вроде OOMKilled и goroutines без расшифровки.

### **Постройте визуальную иерархию.**

Человеческий взгляд сканирует по «F-шаблону»: левый-верх самый заметный. Размещайте KPI именно там, тренд рядом, детализацию — ниже. Шрифт для KPI  $\geq 24$  px, детали 12-14 px.

Ловушка: равный вес всех виджетов, из-за чего пользователь не понимает, на что смотреть первым.

### **Сведите шум к минимуму.**

Исследования Nielsen Norman показывают, что лишние визуальные украшения крадут до 30 % внимания. Убирайте тени, 3-D, «красивые» сетки и клип-арт. Работайте с 2-3 цветами из бренд-палитры.

Ловушка: 3-D-пироги, фон-градиенты и логотип-водяной знак, заслоняющий график.

### **Используйте интерактивность вместо перегрузки.**

Фильтр по датам, чек-бокс «Debug», drill-down по клику позволяют держать на экране только главное.

Ловушка: выносить каждый месяц или сегмент на отдельную страницу, заставляя пользователя прыгать по вкладкам.

### **Сравнивайте показатели рядом.**

Когда «факт» и «план» на одном графике, мозгу не приходится удерживать два диапазона в памяти. Бар-чарты с группировкой, спарклайны «сегодня vs вчера» экономят десятки секунд анализа.

Ловушка: разносить сравниваемые метрики по разным страницам или строить оси с разным масштабом без явной подписи.

### **Выбирайте адекватные типы графиков.**

Линия — для тренда, столбцы — для ранжирования, scatter — для корреляций, heatmap — для времени суток. Поддерживайте семантику цвета: красный — риск, зелёный — рост.

Ловушка: пироги для временных рядов и stacked-area с нечётко различимыми оттенками.

### **Позаботьтесь о производительности и масштабировании.**

Медленный дашборд перестаёт быть полезным. Включайте кеш SQL-запросов, materialized views, rate() в PromQL, шардируйте ClickHouse, выставляйте графану за CDN.

Ловушка: «один большой запрос» на год данных при каждом обновлении и отсутствие индексов.

### **Тестируйте на боевых данных.**

Заглушки обычно идеальны: нет пропусков, отрицательных значений и «скачков». Возьмите анонимизированный снимок production-БД или сгенерируйте нагрузку на staging, чтобы заранее увидеть артефакты.

Ловушка: строить финальную панель на demo-файле sample-superstore.csv и удивляться «битым» графикам в проде.

### **Автоматизируйте жизненный цикл дашборда как кода.**

Храните JSON-конфигурации Grafana/Superset в Git, деплойте их через CI + Terraform/Ansible, пишите unit-тесты на SQL и alert «нет данных > 5 мин».

Ловушка: ручное «клик-клик» в UI; потеря настроек при переустановке контейнера и отсутствие алертов на 0-значения.

## **Часть 2. Обзор инструментов (доступных в РФ, с бесплатным режимом)**

Ниже не таблица, а короткие «портреты» сервисов — что они умеют, как быстро стартовать и где пригодятся бэкендеру.

**Google Looker Studio.** Сервис Google, до сих пор частично доступен из РФ (VPN иногда нужен). Позволяет собирать отчёты буквально «перетаскиванием» метрик из Sheets/BigQuery. Для бэкенда ценен REST-экспорт в JSON и возможность писать собственные визуализации на JS. Идеален для прототипов и демонстраций, но зависит от экосистемы Google.

**Microsoft Power BI Desktop.** Бесплатное десктоп-приложение с богатым DAX-языком и поддержкой прямых SQL-/REST-соединений. Готовые отчёты можно публиковать в Power BI Service (платно) или Power BI Report Server (on-prem). Для back-end-инженера Power BI полезен, когда нужно быстро связать разнородные источники (PostgreSQL, Excel, Web-API) и показать руководству красивую, «корпоративную» панель.

**Tableau Public.** Бесплатно, но все рабочие книги публикуются публично. Хорошо подходит для портфолио или обмена шаблонами. Через `tableau-hyper-api` можно загружать данные из Python. Если нужна конфиденциальность, придётся покупать Tableau Server.

**Grafana OSS.** Де-факто стандарт мониторинга. Разворачивается Helm-чартом в Kubernetes или `docker-compose` за 5 мин. Datasource-плагины к Prometheus, Loki (логи), Tempo (traces), ClickHouse. Terraform-провайдер и API позволяют полностью управлять панелями из Git. Лучший выбор, когда ключевые метрики лежат в Prometheus или ClickHouse.

**Apache Superset.** Open-source BI на Python/Flask. Ставится одной командой `docker-compose up`. Дружит со всеми базами, что понимает SQLAlchemy (Postgres, MySQL, Snowflake, ClickHouse). Имеет SQL Lab, ролевою модель и подключение к OAuth. Оптимален, когда хочется self-host BI с гибкими правами и без vendor-lock.

**Plotly Dash.** Фреймворк, где дашборд — это полноценное Flask-/FastAPI-приложение на Python, а компоненты — React-виджеты. Контроль стопроцентный: от маршрутов до кастомной логики. Разворачивается в Gunicorn + Nginx или как контейнер. Подходит, когда нужно приложить интерактивную панель прямо к существующему бэкенду.

**Kibana (лицензия Basic).** Входит в Elastic-стек, бесплатна для core-функций. Позволяет развернуть визуализацию логов nginx за 5 мин (`docker-run elastic/kibana`). Canvas и Lens дают no-code графики, есть drill-down до исходного документа. Отлична там, где уже есть ELK для логирования.

**Metabase OSS.** Запускается в Docker, поддерживает «запросы без кода», но даёт и чистый SQL-редактор. Умеет отправлять email-/Slack-отчёты, ставить alert при отклонении от порога. Хороший выбор, если нужно быстро повернуть BI для команды и не хочется учить DAX или LookML.

**Redash OSS.** Единый SQL-редактор + визуализации. Более 30 коннекторов (включая ClickHouse). Сниппеты запросов и REST-API позволяют автоматизировать многое. Используется, когда разработчикам нужен единый «запросник» и лёгкая публикация результатов.

**Яндекс DataLens.** SaaS-BI внутри Yandex Cloud; даёт 10 ГБ бесплатного in-memory-движка. Есть Web SQL-IDE, авто-намёки на join-ы и дешёвый S3-storage за спиной. Полностью в рублёвой зоне, так что не страдает от санкций. Хорошо ложится в стек, если сервисы уже живут в YC.

**Форсайт.Аналитика.** Российский Enterprise-продукт (on-prem RPM), сертификация ФСТЭК, ETL-конструктор на Python. Бесплатна только демо-версия; используется крупными госкорпорациями, где важен регуляторный комплаенс.

**DataFan.** Сервис российской разработки, специализируется на соц-медиа-аналитике. Freemium-модель: базовые отчёты бесплатны, за расширенную детализацию — подписка. Позволяет бэкендеру быстро прикрутить мониторинг упоминаний бренда.

**ClickHouse + Grafana.** «Сам себе BI» — OLAP ClickHouse для хранения, Grafana с нативным datasource для визуализации. Стек полностью open-source и устанавливается за полчаса. Удобен, если нужно обрабатывать большие объёмы событий (логов/телеметрии) и сразу строить панели.

### **Часть 3. Примеры дашбордов для back-end-разработчика**

**Kubernetes Cluster Monitoring (Grafana ID 315)** — классическая панель состояния кластера: CPU/Memory/FS per node, состояние pod-ов, системные сервисы. <https://grafana.com/grafana/dashboards/315-kubernetes-cluster-monitoring-via-prometheus/>

**Node Exporter Full (Grafana ID 1860)** — подробный дашборд ресурсов Linux-хоста (CPU, RAM, disk I/O, сеть) на метриках Prometheus Node Exporter. <https://grafana.com/grafana/dashboards/1860-node-exporter-full/>

**Web Server Logs Dashboard (Kibana Sample Data)** — готовый набор визуализаций запросов, ошибок и географии клиентов, поставляется в Kibana вместе с «Sample Logs». <https://www.elastic.co/docs/explore-analyze/dashboards/create-dashboard-of-panels-with-web-server-data>