

Попова Лена| Ведущий инженер

Партизанская война за качество

Или как обеспечить техническое качество в 100+ проектах



Лена Попова

Лид команды DX

Email e.y.popova@tbank.ru

Telegram @PopovaLenaYu



Кто устал от техдолга?

Кто устал планировать техдолг?

Кто устал ручками выпиливать техдолг?

Есть ребята из платформы, которые устали?

Мы устали

Устали от



Ручного поиска проблем

Устали от



Ручного поиска проблем



Однотипных вопросов и анонсов
для команд

Устали от



Ручного поиска проблем



Долгого и болезненного внедрения массовых изменений



Однотипных вопросов и анонсов для команд

Устали от



Ручного поиска проблем



Долгого и болезненного внедрения массовых изменений



Однотипных вопросов и анонсов для команд



Ручного контроля интеграций


Сегодня будет плохо:
A: физически
B: физически и морально
C: морально
морально и физически

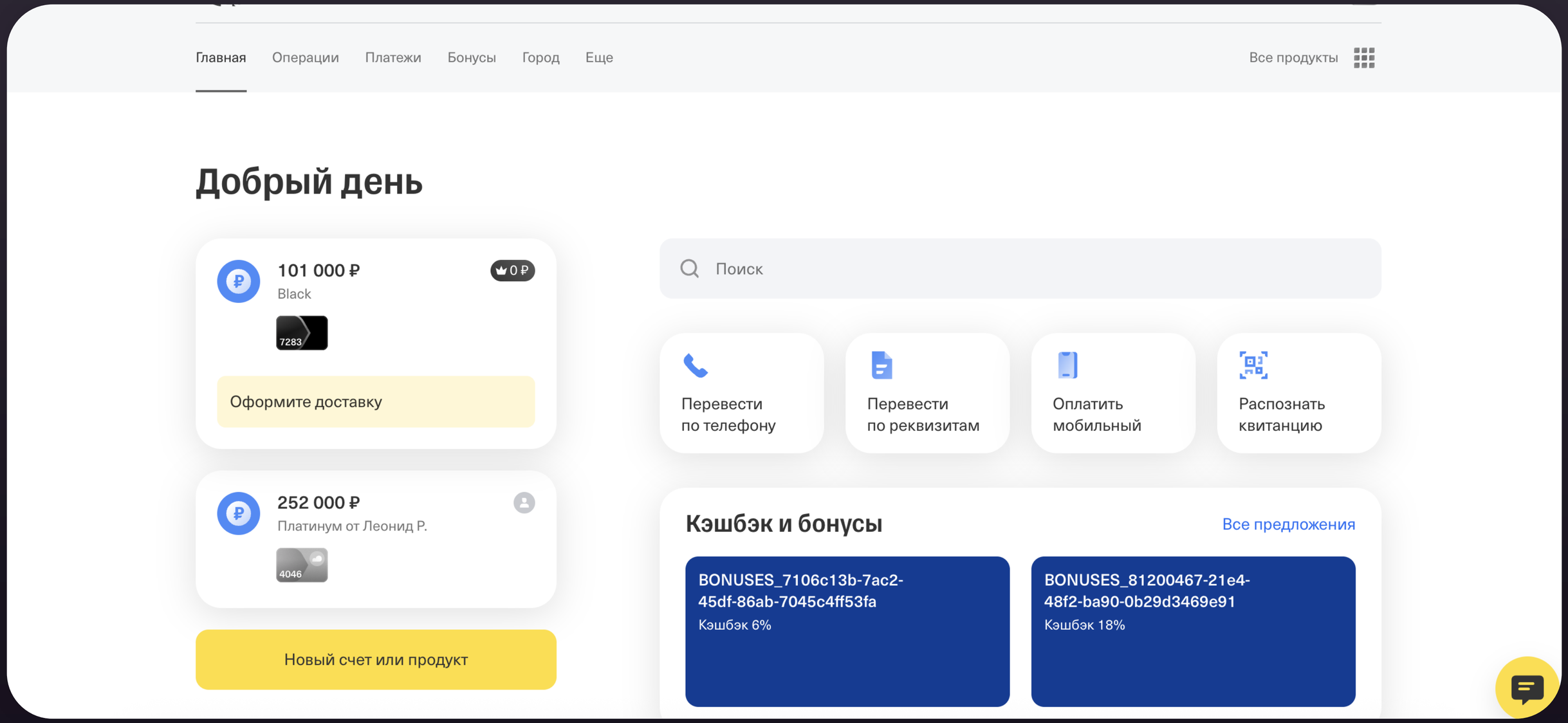


Мы сделали Партизана, который работает за нас

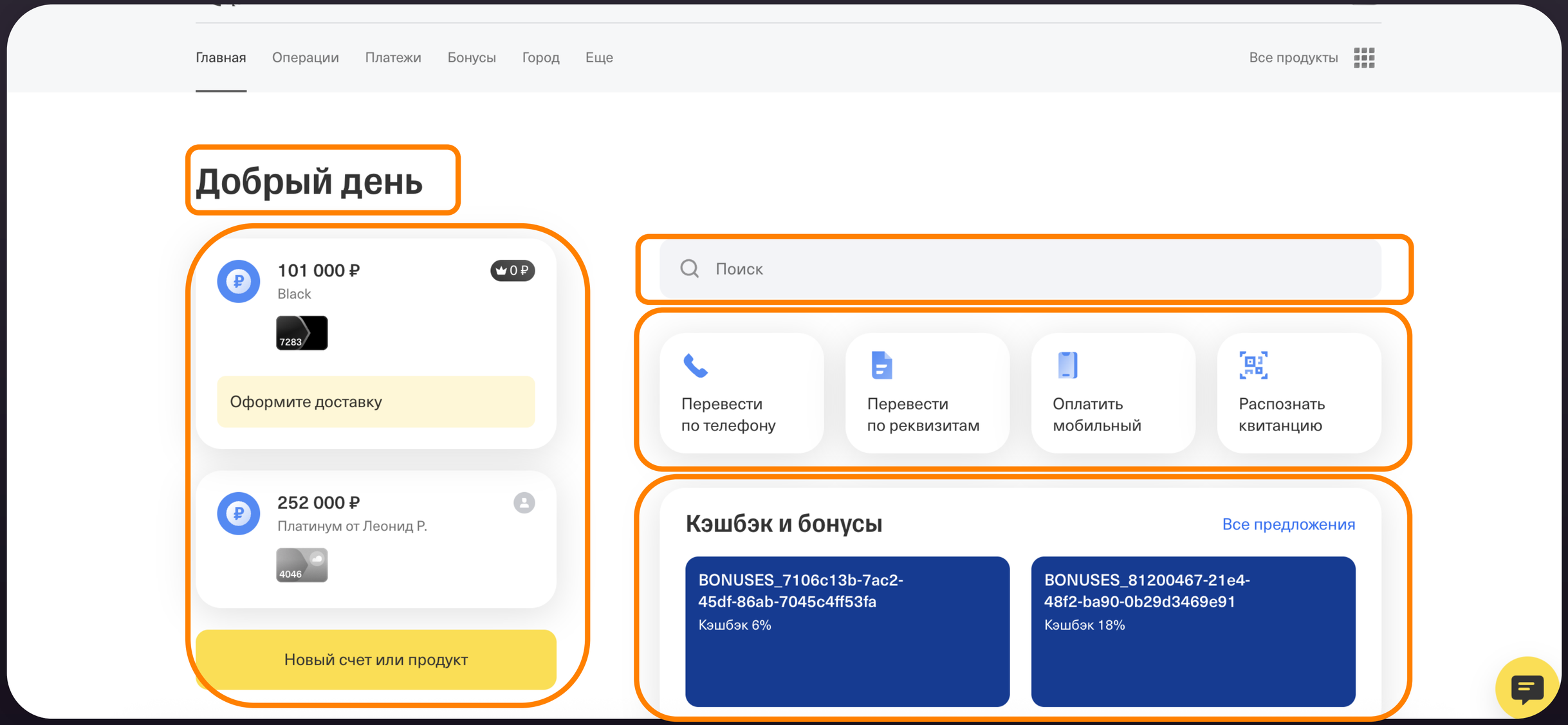


Как рост стал угрозой

Команды создают микрофронты



Команды создают микрофронты



Этапы жизни нашего проекта



Нам срочно нужен новый личный кабинет

Делаем одну репу

Что-то микрофронтов много, давайте разъезжаться на репы

Проект взлетел, появляется больше команд

Ура, нас уже больше 100

Этапы жизни нашего проекта



Нам срочно нужен новый личный кабинет

Делаем одну репу

Что-то микрофронтов много, давайте разъезжаться на репы

Проект взлетел, появляется больше команд

Ура, нас уже больше 100

Этапы жизни нашего проекта



Нам срочно нужен новый личный кабинет

Делаем одну репу

Что-то микрофронтов много, давайте разъезжаться на репы

Проект взлетел, появляется больше команд

Ура, нас уже больше 100

Этапы жизни нашего проекта



Нам срочно нужен новый личный кабинет

Делаем одну репу

Что-то микрофронтов много, давайте разъезжаться на репы

Проект взлетел, появляется больше команд

Ура, нас уже больше 100

Этапы жизни нашего проекта



Нам срочно нужен новый личный кабинет

Делаем одну репу

Что-то микрофронтов много, давайте разъезжаться на репы

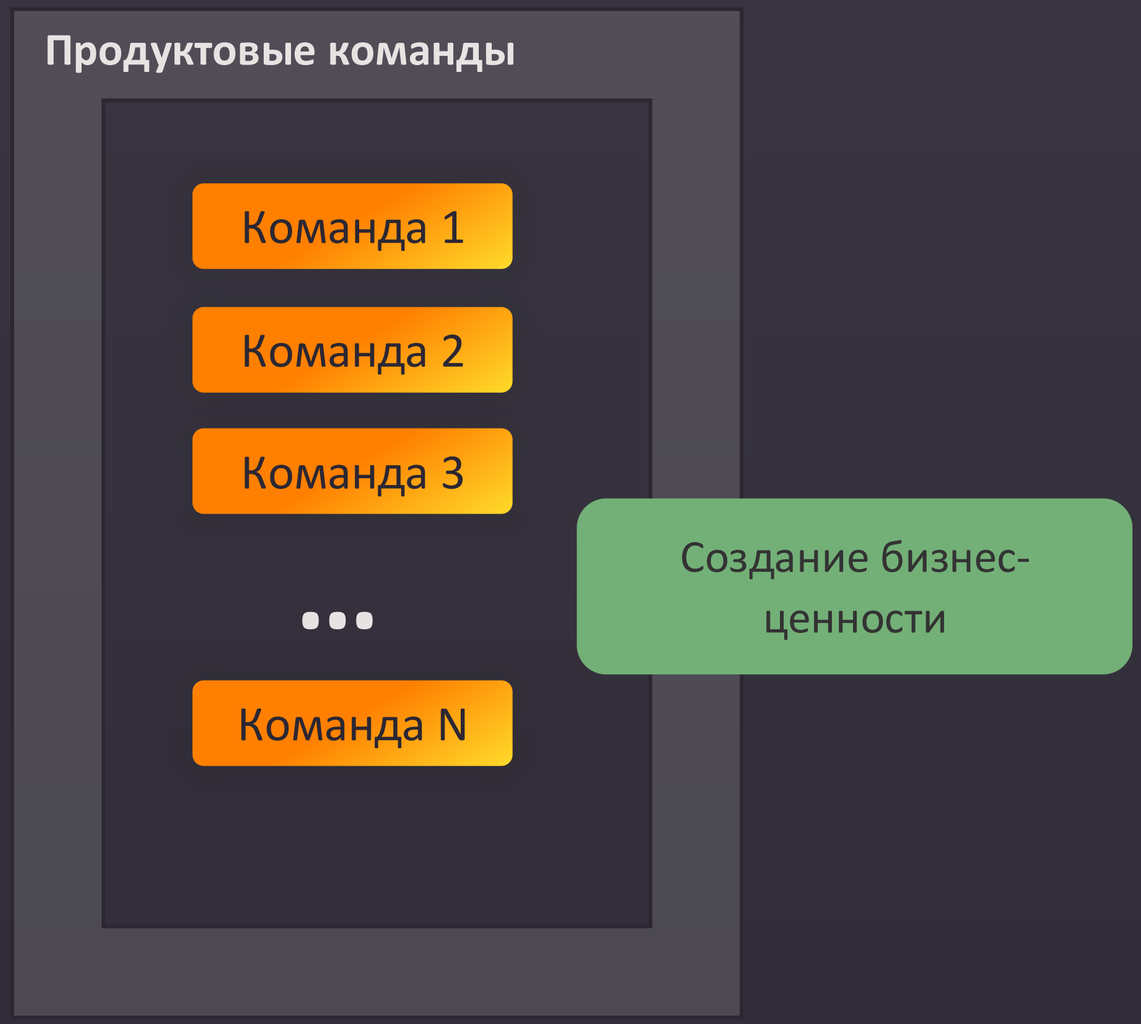
Проект взлетел, появляется больше команд

Ура, нас уже больше 100

Команд много, а я одна



Команд много, а я одна



Создание бизнес-ценности

Команд много, а я одна

Продуктовые команды

Команда 1

Команда 2

Команда 3

...

Команда N

Команда платформы



Команд много, а я одна

Продуктовые команды

Команда 1

Команда 2

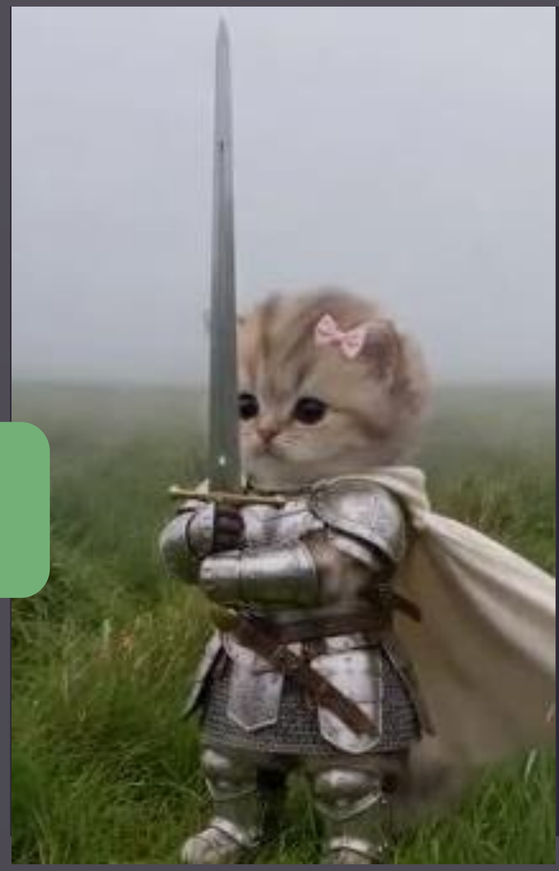
Команда 3

...

Команда N

Все остальное, чтобы бизнес-ценность доставлялась быстро

Команда платформы





Привет проблемы роста!

Привет проблемы роста!

01

Тяжело контролировать состояние реп



Нашли проблему в одной репе, как проверить что ее нет в остальных репах?

Привет проблемы роста!

01

Тяжело контролировать состояние реп

02

Тяжело эффективно раскатывать изменения



Нужно сделать общее изменение – напрягаем 100+ разработчиков


МУСЯ
МУСЯ ТЫ КУДА ЗАЛЕЗЛА

Привет проблемы роста!

01

Тяжело контролировать состояние реп

02

Тяжело эффективно раскатывать изменения

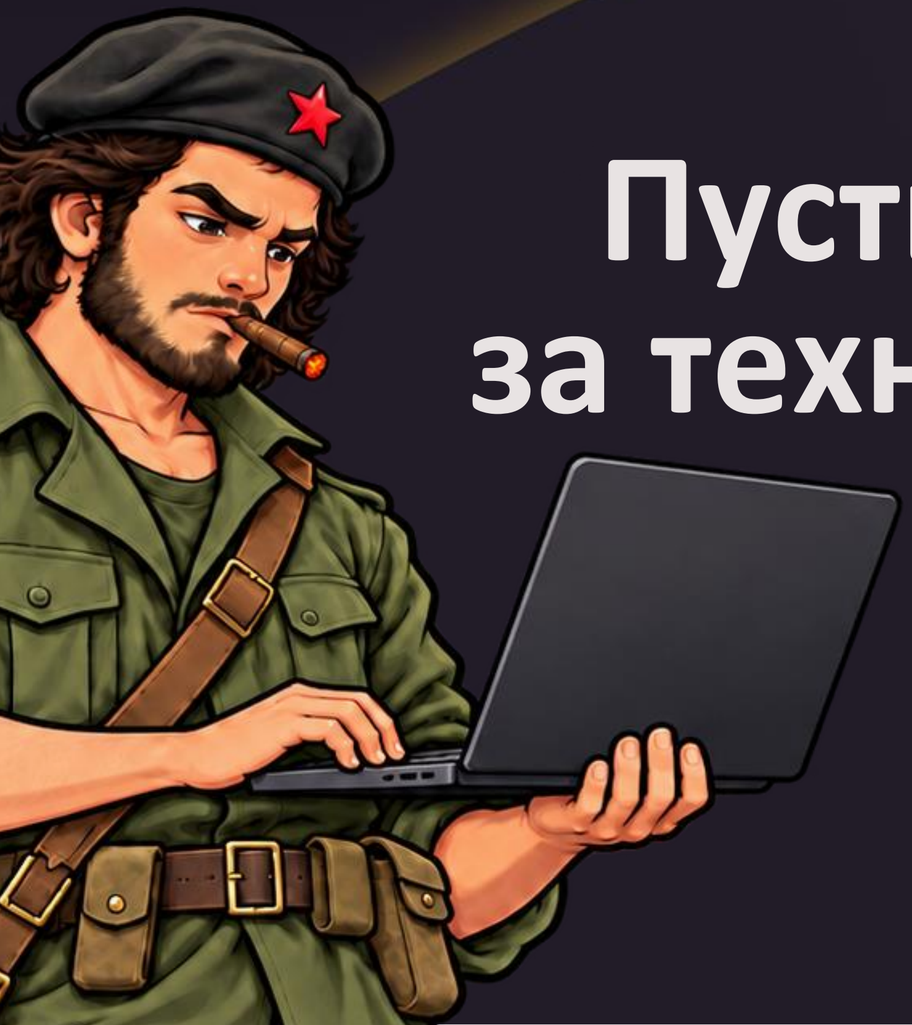
03

Тяжело приоритезировать техдолг



Лид продуктовой команды принимает решения о техдолге доверяя собственной интуиции


ПЕКАРНЯ
Одинаково непонятные
задачи техдолга



Пусть Партизан следит за техническим качеством

Требования к Партизану



Список команд с их репозиториями

Требования к Партизану



Список команд с их репозиториями



Функционал создания проверки
и запуска на репах

Требования к Партизану



Список команд с их репозиториями



Итоговой сформированный отчет по проверкам



Функционал создания проверки и запуска на репах

Требования к Партизану



Список команд с их репозиториями



Итоговой сформированный отчет по проверкам



Функционал создания проверки и запуска на репах



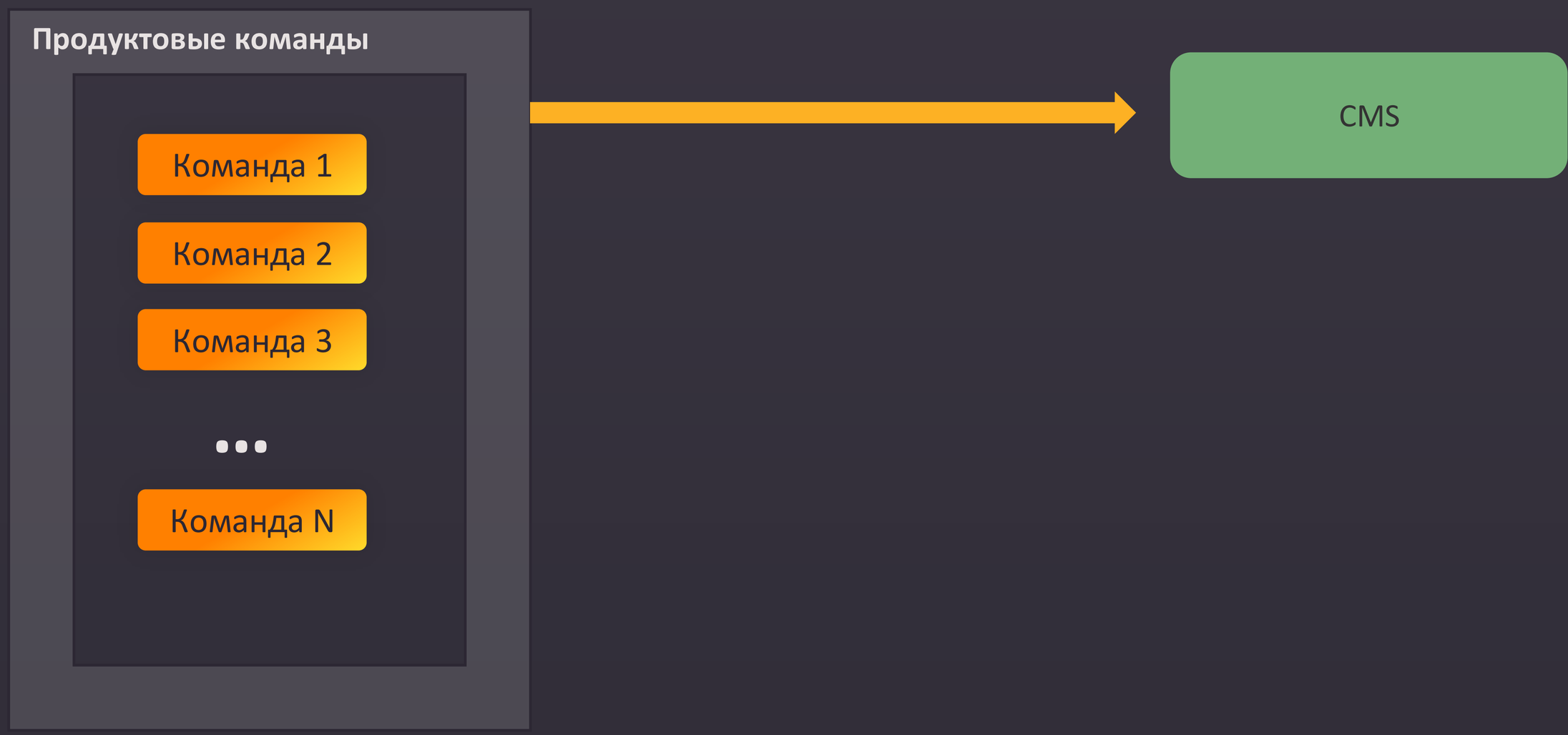
Красивый UI для отображения, чтобы у людей не выпали глаза

Получаем список команд

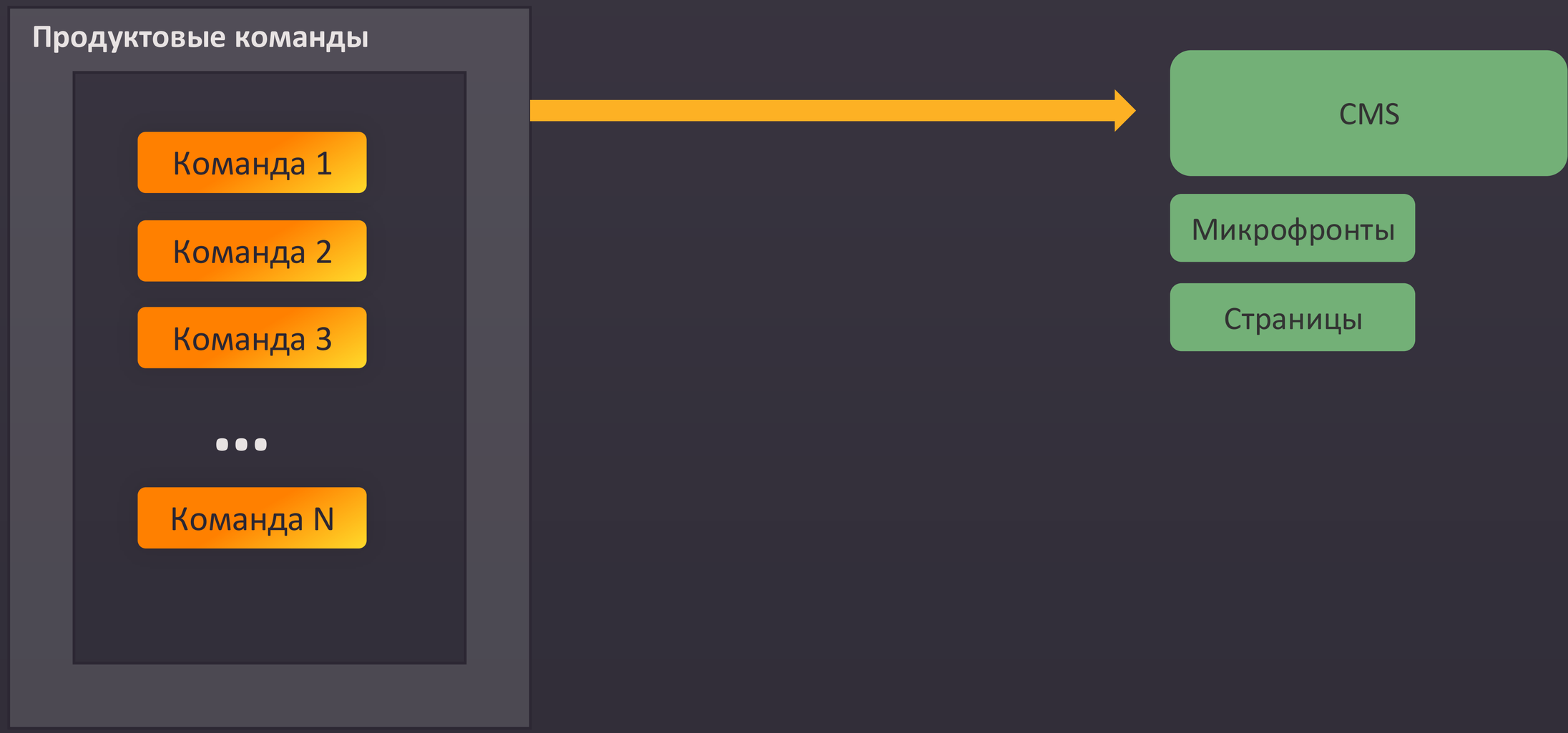
Информация о командах



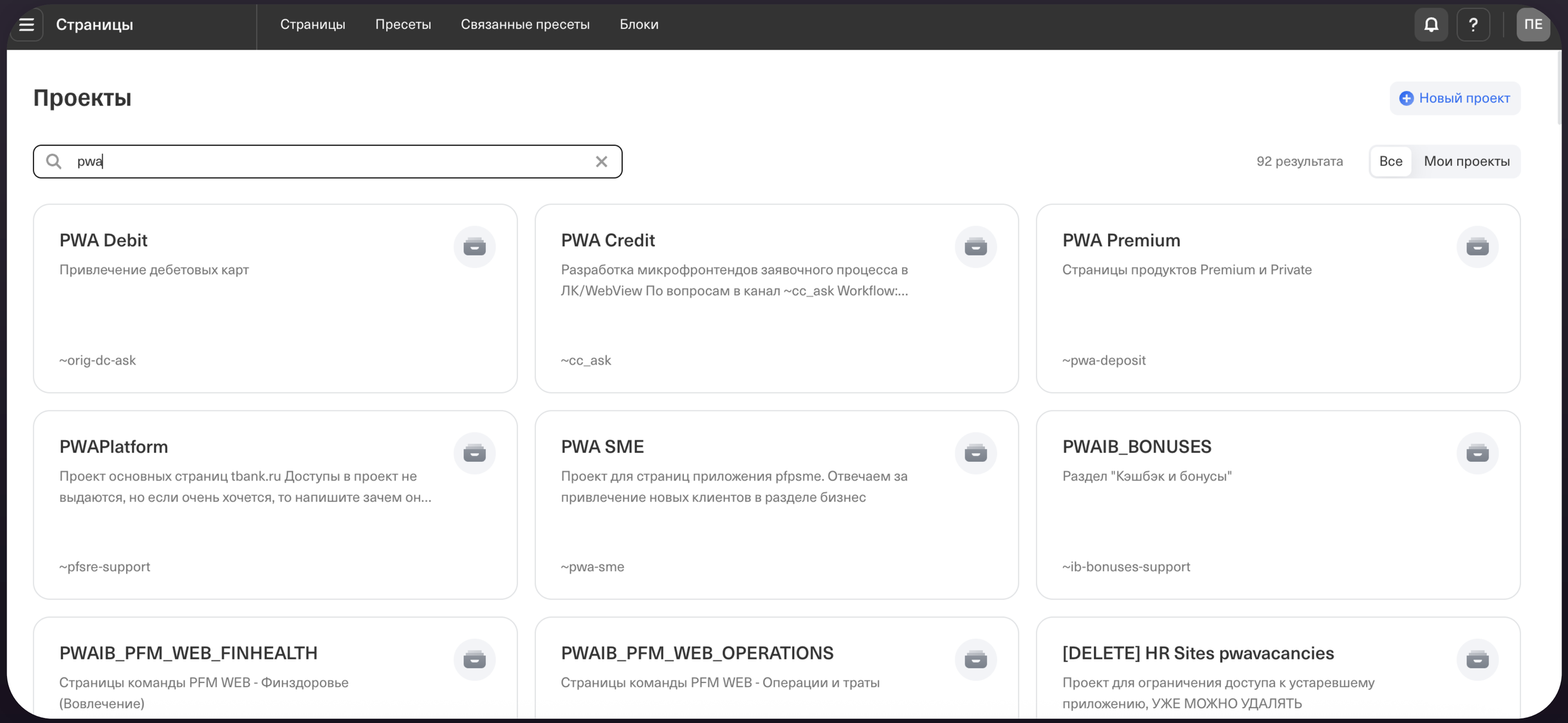
Информация о командах



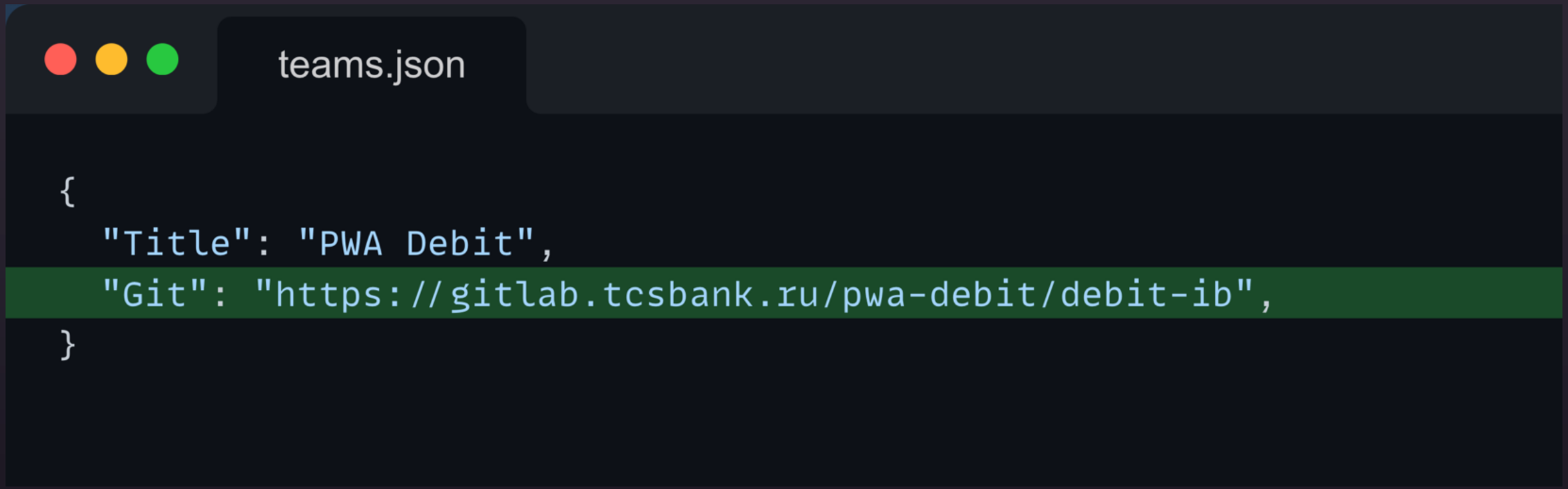
Информация о командах



Список команд



Список команд



Реализуем проверки

Контракт проверок



Passed

Проверка успешно пройдена на репозитории



Failed

Проверка не прошла



Skipped

Проверка не применима к репе



Организация техчеков

В каждой папке – проверка с выполняемым файлом

Функция проверки

check.ts

```ts
export const check: CheckFn = async (repoDir) => {
  const content = fs.readFileSync(`${repoDir}/.gitlab-ci.yml`, "utf8");

  if (!content.includes("SUPER")) {
    return [
      {
        checkName: name,
        description,
        status: "failed",
        meta: { reason: "Ты не супер", gitlabCiYml: content },
      },
    ];
  }

  return [{ checkName: name, description, status: "passed" }];
};
```

Функция проверки

check.ts

```ts
export const name = "check";
export const description = "Прекрасен ли я?";

export const check: CheckFn = async (repoDir) => {
  const content = fs.readFileSync(`${repoDir}/.gitlab-ci.yml`, "utf8");

  if (!content.includes("SUPER")) {
    return [
      {
        checkName: name,
        description,
        status: "failed",
        meta: { reason: "Ты не супер", gitlabCiYml: content },
      },
    ];
  }

  return [{ checkName: name, description, status: "passed" }];
};
```

Функция проверки

check.ts

```ts
export const name = "newCheck";
export const description = "Проверка того, а прекрасен ли я?";
export const fixInstruction = "Подключите все, что можете и не можете"
export const weight = 5;
export const impact = "Так у вас будет сразу все, это хорошо";
```

Функция проверки

check.ts

```ts
export const name = "newCheck";
export const description = "Проверка того, а прекрасен ли я?";
export const fixInstruction = "Подключите все, что можете и не можете"
export const weight = 5;
export const impact = "Так у вас будет сразу все, это хорошо";
```

Функция проверки

check.ts

```ts
export const name = "newCheck";
export const description = "Проверка того, а прекрасен ли я?";
export const fixInstruction = "Подключите все, что можете и не можете"
export const weight = 5;
export const impact = "Так у вас будет сразу все, это хорошо";
```

Пример 1 – Проверка, что CI гарантирует качество

Джобы с тестами должны блокировать MR

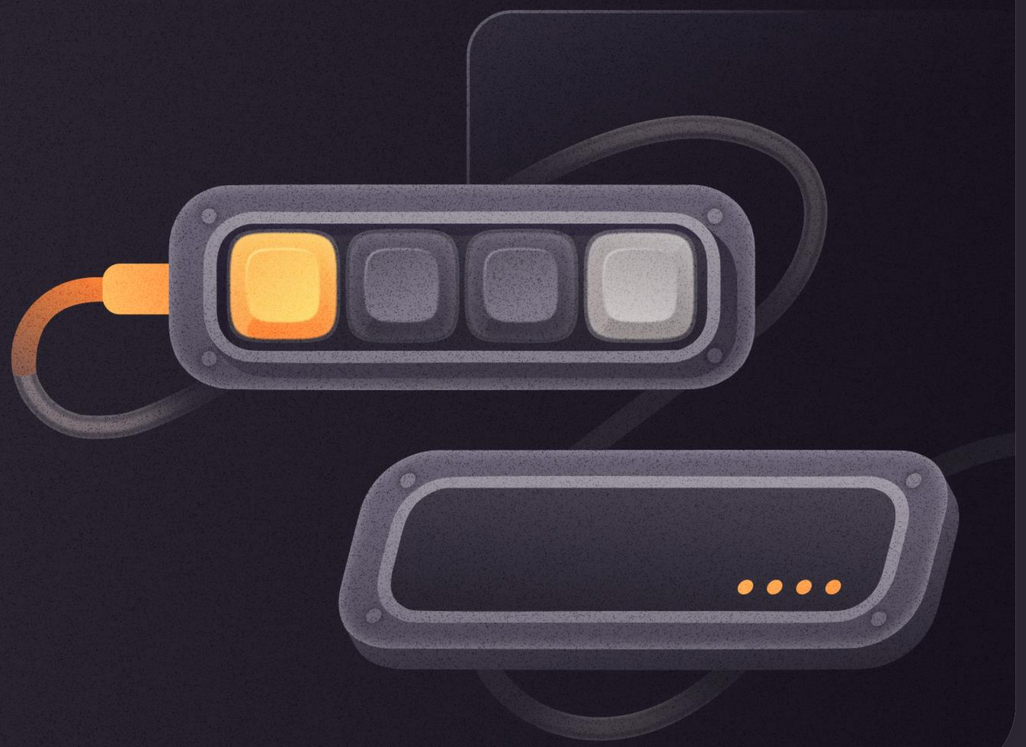
Парсим yaml-файлы

Используем библиотеку yaml





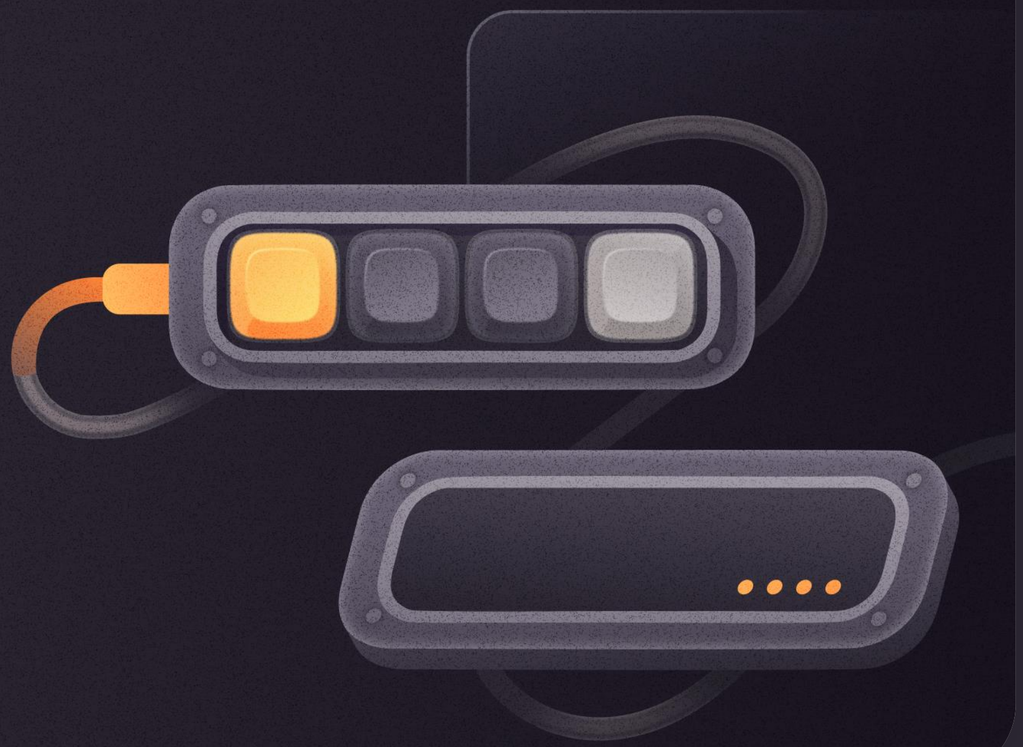
Возможности yaml



Чтение

Одного файла, нескольких, с сохранением форматирования

Возможности yaml



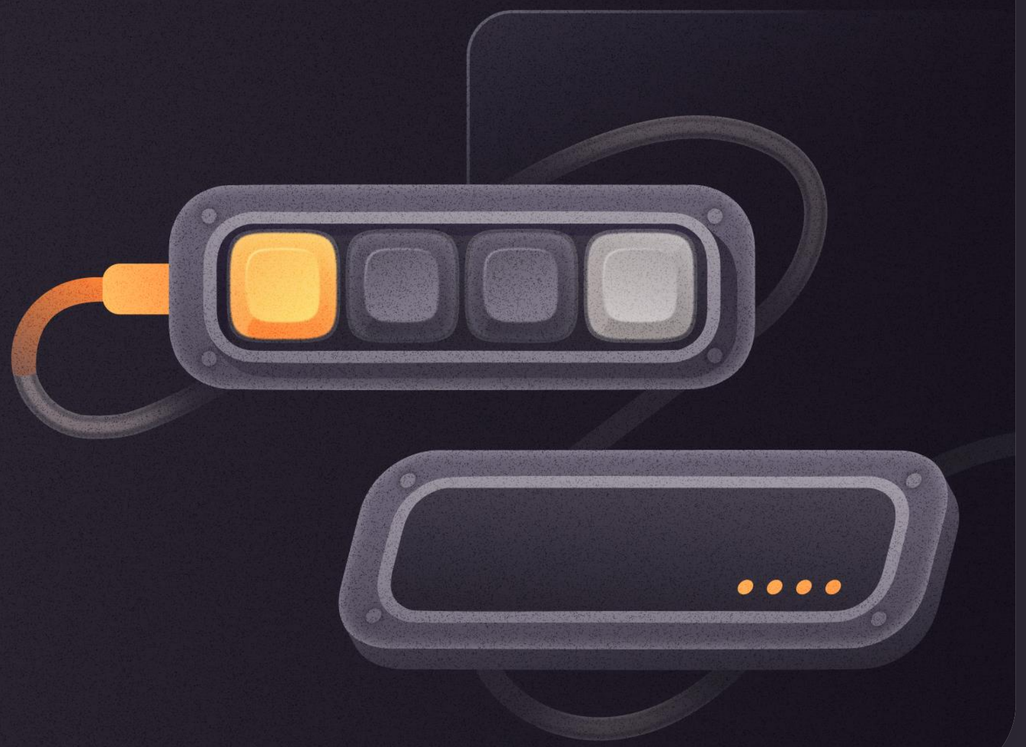
Чтение

Одного файла, нескольких, с сохранением форматирования

Поиск

Знает особенности yaml файлов

Возможности yaml



Чтение

Одного файла, нескольких, с сохранением форматирования

Поиск

Знает особенности yaml файлов

Модификация

Можно редактировать yaml удобно

Алгоритм проверки



Парсим YAML репы

Алгоритм проверки

Отбор джоб MR



Парсим YAML репы

Алгоритм проверки



Парсим YAML репы

Отбор джоб MR

Разворачивание конфига каждой джобы

Алгоритм проверки



Парсим YAML репы

Отбор джоб MR

Разворачивание конфига каждой джобы

Отбираем джобы allow_failure: true

Алгоритм проверки



Парсим YAML репы

Отбор джоб MR

Разворачивание конфига каждой джобы

Отбираем джобы allow_failure: true

Формирование лога

Ищем джобы MR

check.ts

```ts
const runsInMR = (job: Record<string, any>): boolean => {
  const rules = job.rules;
  const only = job.only;

  if (Array.isArray(rules)) {
    for (const rule of rules) {
      if (typeof rule === "object" && rule !== null) {
        const ifCondition = rule.if;
        if (ifCondition && typeof ifCondition === "string") {
          if (
            ifCondition.includes('$CI_PIPELINE_SOURCE == "merge_request_event"') ||
            ifCondition.includes("$CI_PIPELINE_SOURCE == 'merge_request_event'") ||
            ifCondition.includes("merge_request")
          ) {
            return true;
          }
        }
      }
    }
    return false;
  }
```

Ищем джобы MR

check.ts

```ts
onst runsInMR = (job: Record<string, any>): boolean => {
  const rules = job.rules;
  const only = job.only;

  // Если есть rules, проверяем их
  if (Array.isArray(rules)) {...}

  // Если есть only
  if (Array.isArray(only)) {
    return only.includes("merge_requests");
  }

  if (typeof only === "string") {
    return only === "merge_requests";
  }

  // Если нет ограничений (no rules, no only), джоба запускается всегда
  if (!only && !rules) {
    return true;
  }

  return false;
};
```

Проверяем, что джоба обязательна

check.ts

```ts
jobs.forEach((job) => {
  // config — полностью свёрнутое представление джобы (extends + !reference)
  const config = job.resolve().value;

  // Добавляем джобы, которые:
  // - запускаются в MR
  // - имеют allow_failure: true
  if (runsInMR(config) && isAllowFailure(config.allow_failure)) {
    violations.push(job.name);
  }
});
```

Проверяем, что джоба обязательна

check.ts

```ts
const isAllowFailure = (value: unknown): boolean ⇒
  value === true || (typeof value === "object" && value !== null);
```

Формируем результат

check.ts

```ts
jobs.forEach((job) => {
  const config = job.resolve().value;

  if (
    runsInMR(config) &&
    isAllowFailure(config.allow_failure)
  ) {
    violations.push(job.name);
  }
});

if (violations.length > 0) {
  return [
    {
      checkName: name,
      description,
      status: "failed",
      meta: {
        violations,
        reason: `Найдены джобы, которые должны блокировать пайплайн, но имеют allow_failure: true или могут быть скипнуты:\n${violations.join(";\n")}`,
      },
    } as SubCheckResult,
  ];
}
```

Результат работы

check.json

```
{
  "name": "ciBlockingJobs",
   "checkName": "ciBlockingJobs",
   "description": "Джобы с проверками, не зависящими от внешних систем, должны блокировать пайплайн MR",
   "status": "failed",
   "meta": {
       "violations": [
         "jest",
         "playwright",
         "storyshot-pages",
         "test-storyshots",
         "test-hydration",
         "test-ssr"
       ],
     "reason": "Найдены джобы, которые должны блокировать пайплайн, но имеют allow_failure: true или могут быть скипнуты:
   },
   "repoName": "lfsweb/bipka"
}
```

Задачка 2 – проверить, что в команду подключены проверки доступности

Что для этого нужно

01 В test-runner Storybook включена проверка доступности

Что для этого нужно

01 В test-runner Storybook включена проверка доступности

02 Проверка не отключена целиком

Что для этого нужно

01 В test-runner Storybook включена проверка доступности

02 Проверка не отключена целиком

03 Не игнорируются конкретные части проверки

Придется парсить AST дерево TS кода

Парсим код с помощью babel

01 @babel/parser – парсинг кода в AST

Парсим код с помощью babel

01 @babel/parser – парсинг кода в AST

02 @babel/traverse – обход и поиск узлов

Парсим код с помощью babel

01 @babel/parser – парсинг кода в AST

02 @babel/traverse – обход и поиск узлов

03 @babel/types – проверка типов узлов

Парсим код с помощью babel

01 @babel/parser – парсинг кода в AST

02 @babel/traverse – обход и поиск узлов

03 @babel/types – проверка типов узлов

04 @babel/generator – AST в код

Алгоритм проверки



Парсинг импортов
в test-runner

Алгоритм проверки

Если test-runner не включен, то
все остальное неважно



Парсинг импортов
в test-runner

Алгоритм проверки

Если test-runner не включен, то
все остальное неважно



Парсинг импортов
в test-runner

Проверяем, что не отключили
проверку a11y целиком

Алгоритм проверки



Парсинг импортов
в test-runner

Если test-runner не включен, то
все остальное неважно

Проверяем, что не отключили
проверку a11y целиком

Проверяем, что не отключена
часть правил a11y

Алгоритм проверки



Парсинг импортов
в test-runner

Если test-runner не включен, то
все остальное неважно

Проверяем, что не отключили
проверку a11y целиком

Проверяем, что не отключена
часть правил a11y

Собираем результаты ошибок

Парсинг кода в AST

check.ts

```ts
export const processFilesWithAST = async <T>(
  repo: RepoDir,
  pattern: string,
  processor: (ast: t.File, filePath: string) => Promise<T> | T,
): Promise<T[]> => {
  const glob = new Glob(pattern, {
    cwd: repo,
    absolute: true,
  });

  const results: T[] = [];

  for await (const filePath of glob) {
    try {
      const content = await fs.readFile(filePath, "utf-8");

      const ast = parse(content, {
        sourceType: "module",
        plugins: ["typescript", "jsx"],
      });
```

Обход узлов

check.ts

```ts
export function scanTreeForA11yDisabled(ast: t.File): Match[] {
  const matches: Match[] = [];

  traverse(ast, {
    ObjectExpression(path) {
      const a11yDisabledpMatches = checkA11yDisabled(path);
      const htmlValidatorDisabledMatches = checkHtmlValidatorDisabled(path);

      matches.push(...a11yDisabledpMatches, ...htmlValidatorDisabledMatches);
    },
  });

  return matches;
}
```

Поиск узлов

check.ts

```ts
import traverse, { type NodePath } from "@babel/traverse";
import * as t from "@babel/types";

const checkA11yDisabled = (path: NodePath<t.ObjectExpression>): Match[] => {
  const parentPath = path.parentPath;
  const result: Match[] = [];

  if (
    parentPath?.isObjectProperty() &&
    t.isIdentifier(parentPath.node.key) &&
    parentPath.node.key.name === "a11y"
  ) {
    for (const prop of path.node.properties) {
      // a11y.disable = true
      if (
        t.isObjectProperty(prop) &&
        t.isIdentifier(prop.key) &&
        prop.key.name === "disable" &&
        t.isBooleanLiteral(prop.value) &&
        prop.value.value === true
      ) {
        result.push(generateMatch(prop));
      }
  }

  return result;
};
```

Преобразование обратно

check.ts

```ts
import traverse, { type NodePath } from "@babel/traverse";
import * as t from "@babel/types";

const checkA11yDisabled = (path: NodePath<t.ObjectExpression>): Match[] => {
  const parentPath = path.parentPath;
  const result: Match[] = [];

  if (
    parentPath?.isObjectProperty() &&
    t.isIdentifier(parentPath.node.key) &&
    parentPath.node.key.name === "a11y"
  ) {
    for (const prop of path.node.properties) {
      // a11y.disable = true
      if (
        t.isObjectProperty(prop) &&
        t.isIdentifier(prop.key) &&
        prop.key.name === "disable" &&
        t.isBooleanLiteral(prop.value) &&
        prop.value.value === true
      ) {
        result.push(generateMatch(prop));
      }
  }

  return result;
};
```

Преобразование обратно

check.ts

```ts
import generate from "@babel/generator";
import * as t from "@babel/types";

import { Match } from "../types";

export const generateMatch = (node: t.Node): Match => {
  return {
    line: node.loc?.start.line ?? 0,
    column: node.loc?.start.column ?? 0,
    code: generate(node).code,
  };
};
```

Инструменты для поиска

Кейс	Инструмент	Почему
Простой строковый паттерн	grep	Это просто
Анализ YAML/JSON конфигов	Соответствующий парсер	Структурированный доступ
Нужен контекст кода (импорты, вызовы, объекты)	AST	Точность, понимание семантики

Запуск и отчет

Запуск техчеков



Клонируем все репы

Запуск техчеков

Парсим папку с чеками
и загружаем их



Клонируем все
репы

Запуск техчеков

Парсим папку с чеками
и загружаем их



Клонируем все
репы

Каждый техчек прогоняем
на каждой репе

Запуск техчеков



Клонируем все репы

Парсим папку с чеками и загружаем их

Каждый техчек прогоняем на каждой репе

Формируем JSON отчета

Запуск техчеков



Клонируем все репы

Парсим папку с чеками и загружаем их

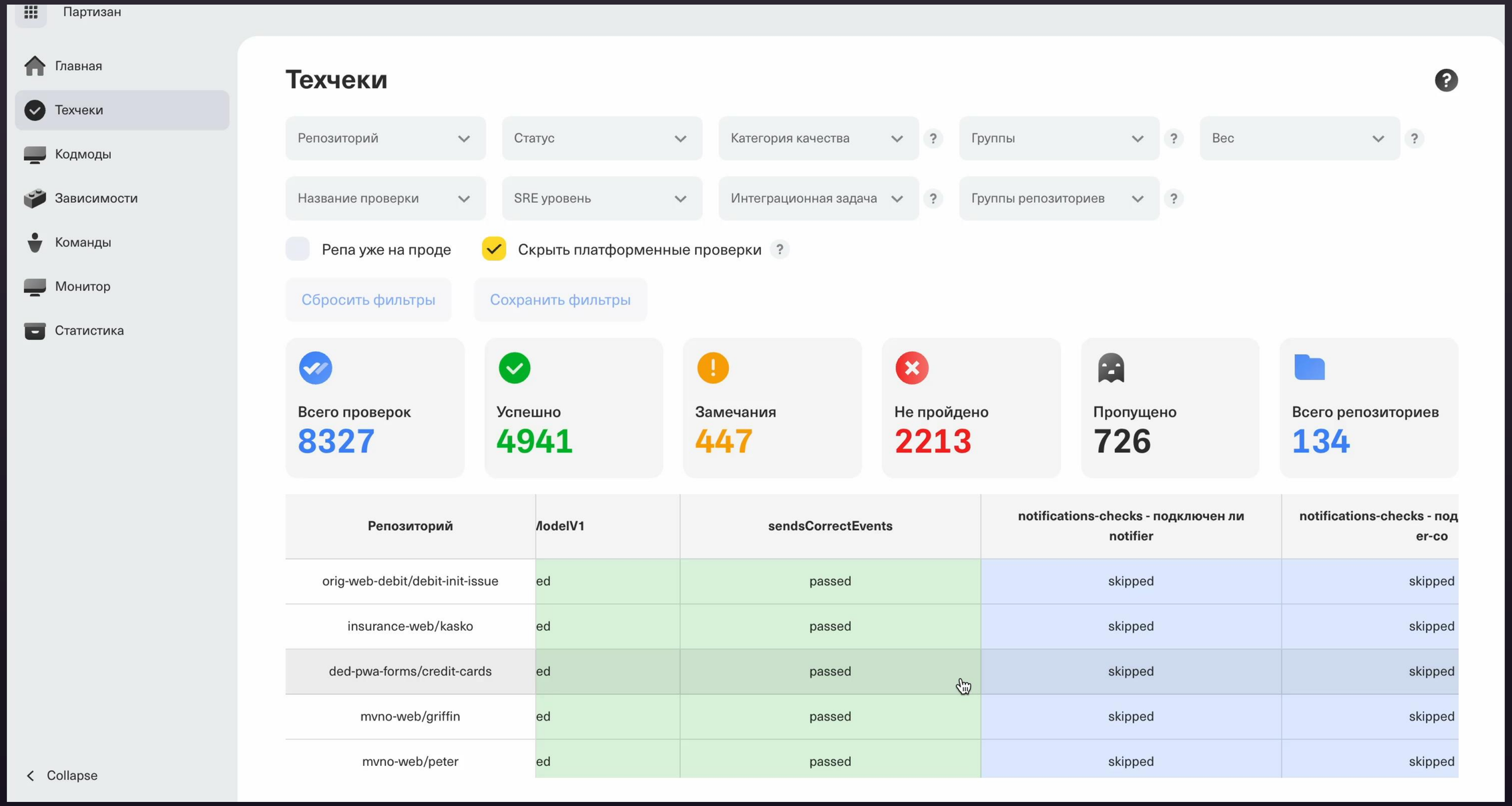
Каждый техчек прогоняем на каждой репе

Формируем JSON отчета

Вспоминаем, что мы фронтендеры и наводим красоту

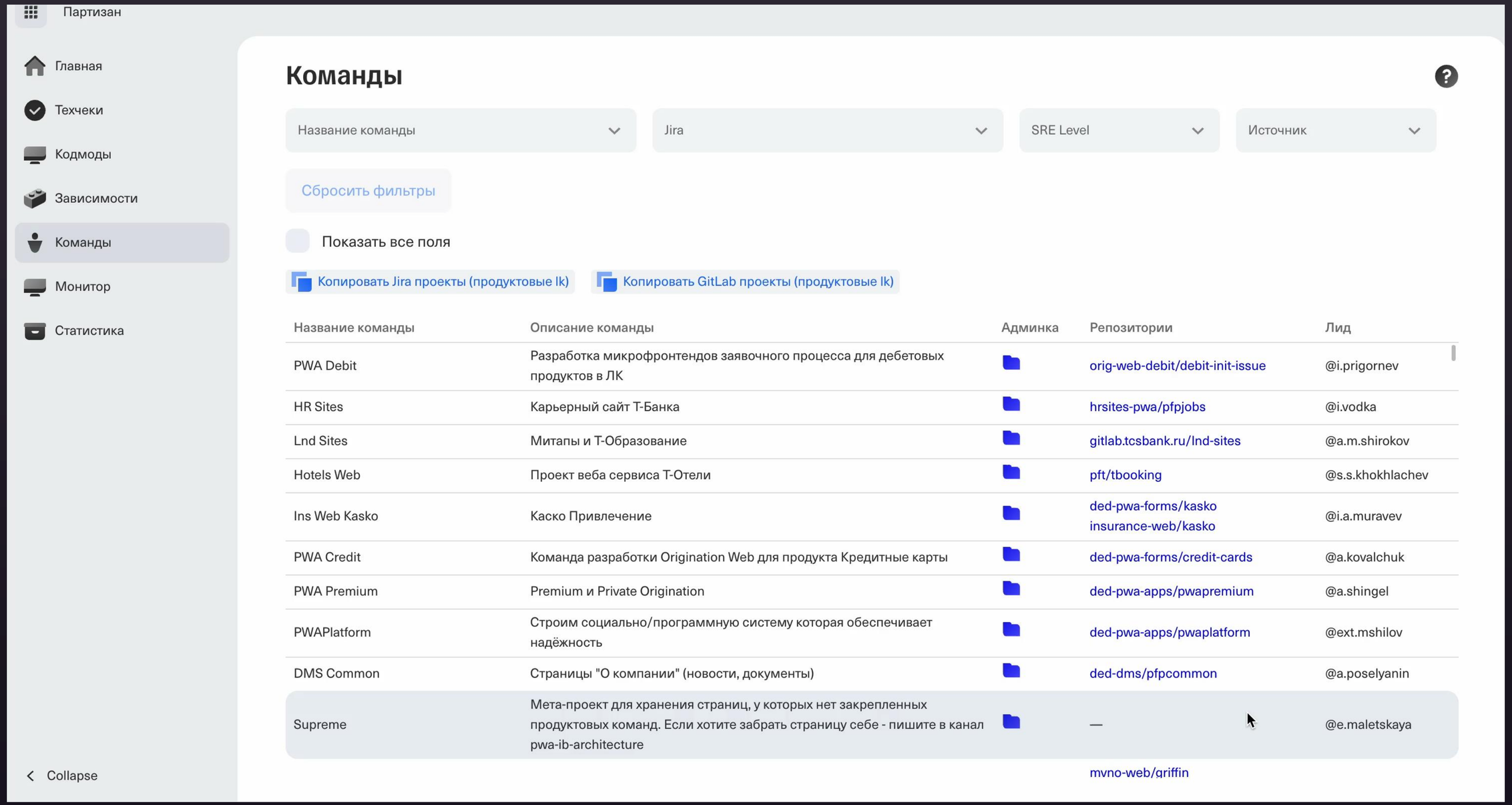
Запускаем каждую ночь

Сценарий – мы хотим понять состояние реп

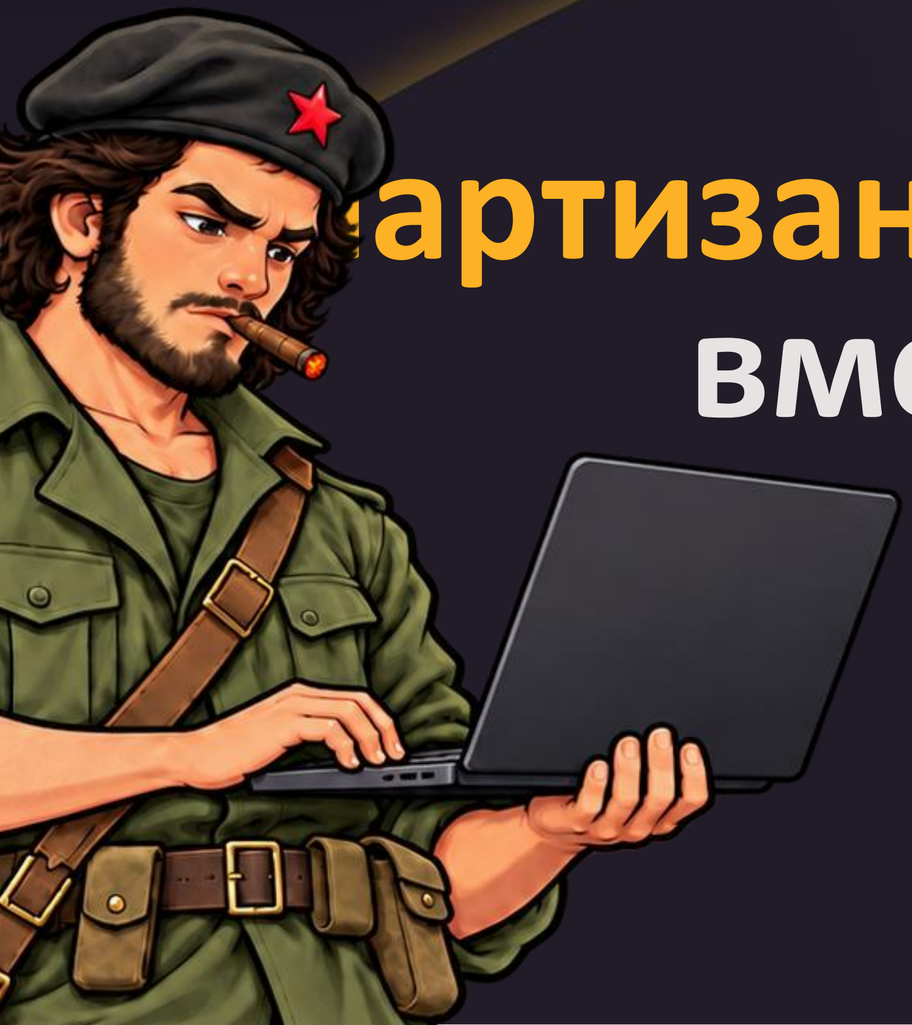


Репозиторий	ModelV1	sendsCorrectEvents	notifications-checks - подключен ли notifier	notifications-checks - под er-co
orig-web-debit/debit-init-issue	ed	passed	skipped	skipped
insurance-web/kasko	ed	passed	skipped	skipped
ded-pwa-forms/credit-cards	ed	passed	skipped	skipped
mvno-web/griffin	ed	passed	skipped	skipped
mvno-web/peter	ed	passed	skipped	skipped

Сценарий – лид хочет понять свой техдолг



Партизан следит за вашим качеством



Партизан делает изменения вместо кожаных

Требования к Партизану



Список команд с их репозиториями

Требования к Партизану



Список команд с их репозиториями



Инструмент для написания
функций, модифицирующих код

Требования к Партизану



Список команд с их репозиториями



Функционал для автоматического создания MR-ов с изменениями



Инструмент для написания функций, модифицирующих код

Требования к Партизану



Список команд с их репозиториями



Функционал для автоматического создания MR-ов с изменениями

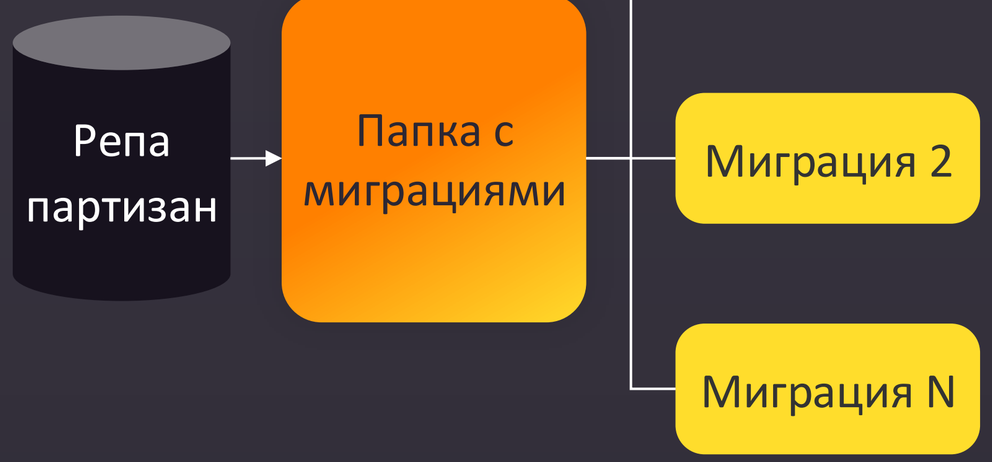


Инструмент для написания функций, модифицирующих код



Ну и опять красоты навести охота

Модификация кода



Организация миграций

В каждой папке – одна миграция кода

Функция миграции

run.ts

```ts
export const run = async () => {
  const pkg = JSON.parse(fs.readFileSync("package.json", { encoding: "utf8" }));

  console.log("@@name:", pkg.name);
};
```

Функция миграции

check.ts

```ts
import fs from "node:fs";

import { MigrationMeta } from "../../utils/migration-schema";

export const meta: MigrationMeta = {
  name: "Super migration",
  taskId: "HUBERT-00",
  changesType: "refactor",
};

export const run = async () => {
  const pkg = JSON.parse(fs.readFileSync("package.json", { encoding: "utf8" }));

  console.log("@@name:", pkg.name);
};
```

Переезд на новое API

Регулярочкой меняем API1 на API2

check.ts

```ts
export const transformStories: TransformFn = async (file: string, content: string) => {
  try {
    if (!content.includes(CLICKABLE_AREA_OLD)) {
      return undefined;
    }

    const updatedContent = content.replace(CLICKABLE_AREA_REGEX, `${CLICKABLE_AREA_NEW}(`);

    if (content !== updatedContent) {
      return prettierFormat(file, updatedContent);
    }

    return undefined;
  } catch (e) {
    console.error(`Ошибка при обработке файла ${file}:`, e);
  }
};
```

Оптимизация сборки Storybook

Что нужно оптимизировать

Мигрируем на современный аддон для стилей

Переезжаем на современный компилятор SWC

Оптимизируем бандлер Webpack

Отключаем неиспользуемые настройки

Как выглядит конфиг Storybook

config.ts

```ts
import type { StorybookConfig } from '@storybook/react-webpack5';

const config: StorybookConfig = {
  stories: getStories(),
  addons: [
    {
      name: '@storybook/addon-styling',
      options: {
        postCss: true,
        cssModules: {
          localIdentName: '[path][name]__[local]',
        },
      },
    },
    '@css-variables/storybook-addon',
  ],
  framework: {
    name: '@storybook/react-webpack5',
    options: {},
  },
  env: () => ({
    STORY_GENERATORS_DISABLE_MMB_BUILD: (!process.env.CI).toString(),
  }),
  staticDirs: [{ from: '../public/mm', to: '/mm' }],
  typescript: {
```

Оптимизация webpack – список оптимизаций

check.ts

```
function buildOptimizationEntries(
  paramName: string,
  major: SupportedMajor,
): Array<{ name: string; initializer: string }> {
  return [
    {
      name: "devtool",
      initializer: `${paramName}.mode === 'production' ? false : (${paramName}.devtool === false ? false : 'eval-cheap-module-source-map')`,
    },
    {
      name: "experiments",
      initializer: `${paramName}.mode === 'development' ? { ...${paramName}.experiments, lazyCompilation: { entries: false, imports: true } }`,
    },
    {
      name: "watchOptions",
      initializer: `{ ...${paramName}.watchOptions, ignored: ['**/node_modules/**', '**/.git/**', '**/public/storybook/**', '**/dist/**'] }`,
    },
    { name: "module", initializer: buildModuleInitializer(paramName, major) },
    { name: "output", initializer: `{ ...${paramName}.output, pathinfo: false }` },
    {
      name: "optimization",
      initializer: `{ ...${paramName}.optimization, removeAvailableModules: false, removeEmptyChunks: false }`,
    },
  ];
}
```

Снова придется работать с AST деревом

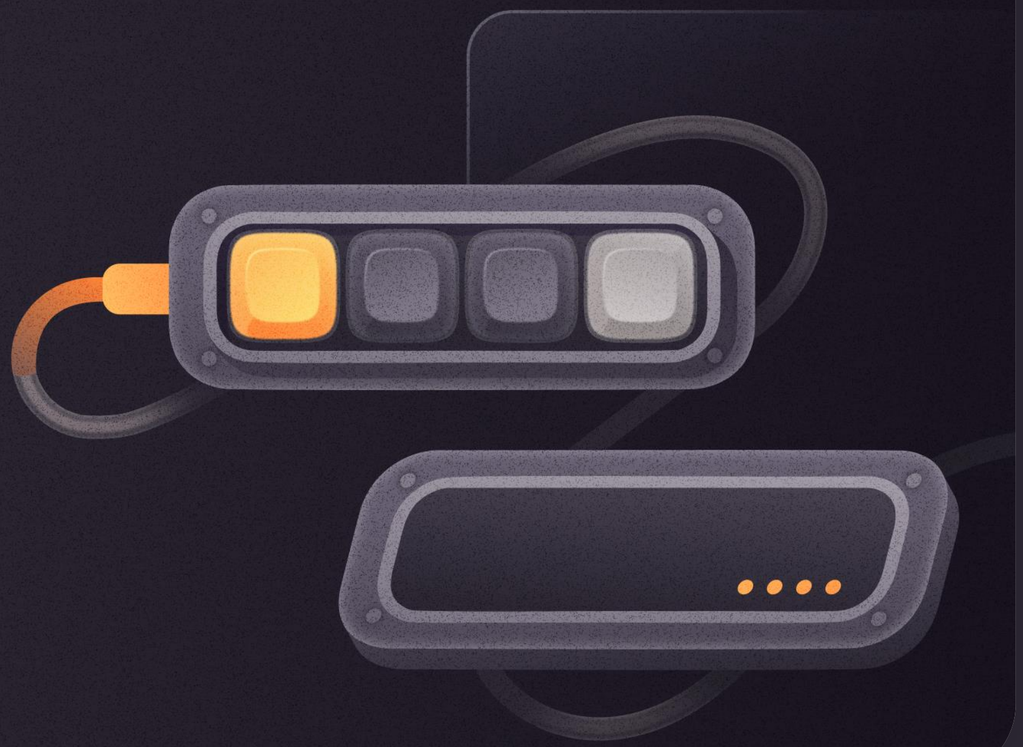
Модификация кода

Используем ts-morph



TS-MORPH

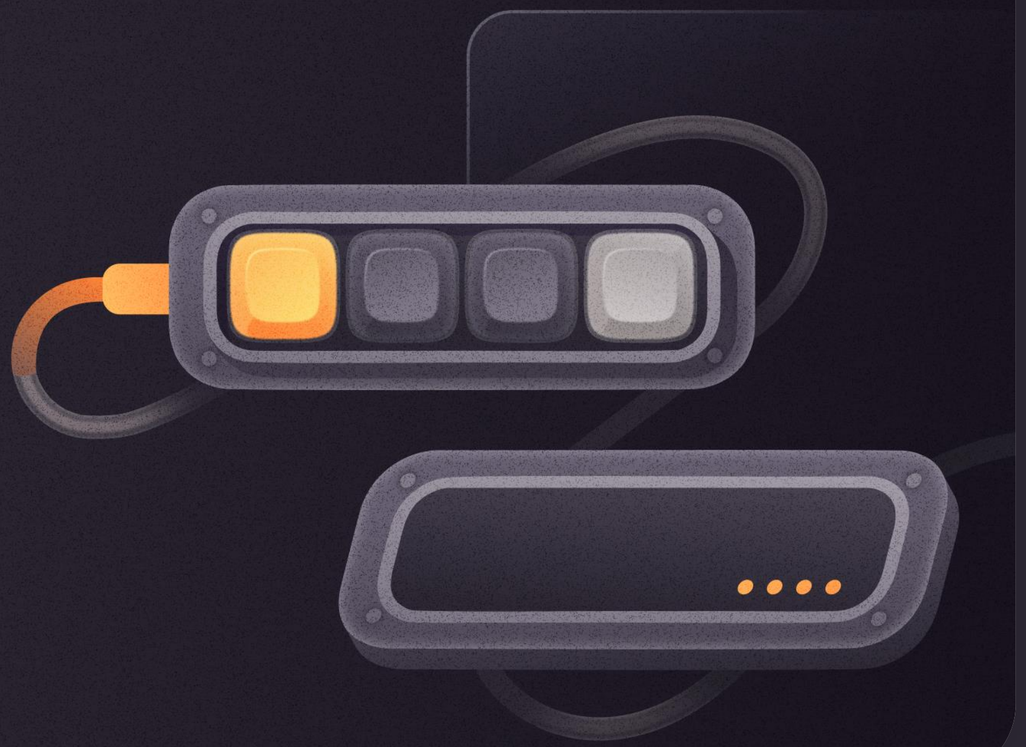
Возможности ts-morph



Чтение

Парсинг файлов, получение AST дерева

Возможности ts-morph



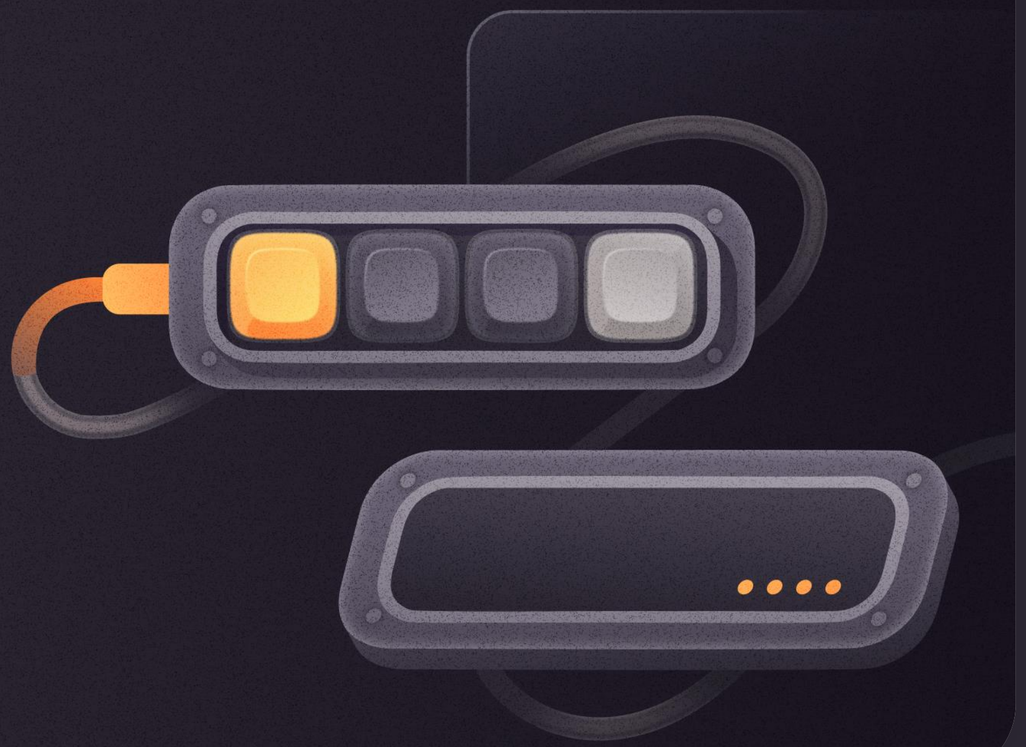
Чтение

Парсинг файлов, получение AST дерева

Навигация

Обход дерева вверх/вниз/вглубь

Возможности ts-morph



Чтение

Парсинг файлов, получение AST дерева

Навигация

Обход дерева вверх/вниз/вглубь

Модификация

Добавление, изменение, переименование, удаление узлов. И в целом все что может прийти в голову

Алгоритм работы



Определение
версии Storybook

Алгоритм работы



Определение версии Storybook

Апдейт настроек webpack

Алгоритм работы



Определение версии Storybook

Апдейт настроек webpack

Подключение swc

Алгоритм работы



Определение версии Storybook

Апдейт настроек webpack

Подключение swc

Мигрируем на новый аддон

Алгоритм работы



Определение версии Storybook

Апдейт настроек webpack

Подключение swc

Мигрируем на новый аддон

Запускаем TS

Настройка webpack - находим webpack в конфиге

run.ts

```
const existing = config.getProperty("webpackFinal");
```

Настройка webpack – ищем нужные опции

run.ts

```
if (!first.isKind(SyntaxKind.SpreadAssignment)) return false;
const spreadExpr = first.getExpression();
if (!spreadExpr.isKind(SyntaxKind.Identifier)
|| spreadExpr.getText() !== paramName) {
  return false;
}
```

Настройка webpack – анализируем содержимое поля

run.ts

```
    if (!first.isKind(SyntaxKind.SpreadAssignment)) return false;
    const spreadExpr = first.getExpression();
    if (!spreadExpr.isKind(SyntaxKind.Identifier)
    || spreadExpr.getText() !== paramName) {
      return false;
    }
```

Настройка webpack – вставляем недостающие опции

run.ts

```
for (const entry of buildOptimizationEntries(paramName, major)) {
    if (presentKeys.has(entry.name)) continue;
    obj.addPropertyAssignment(entry);
    added = true;
}
```

Запускаем проверку TS - создаем проект

check.ts

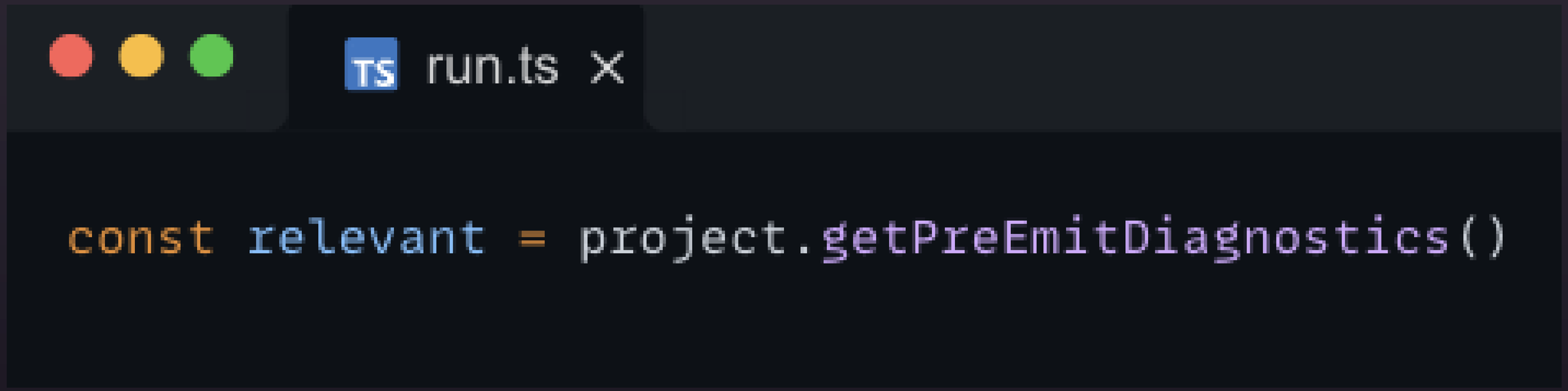
```ts
function typecheckStorybookConfig(file: string): { ok: true } | { ok: false; message: string } {
  try {
    const project = new Project({
      compilerOptions: {
        noEmit: true,
        skipLibCheck: true,
        allowJs: true,
        esModuleInterop: true,
        moduleResolution: ModuleResolutionKind.NodeJs,
        resolveJsonModule: true,
        target: ScriptTarget.ESNext,
        module: ModuleKind.ESNext,
        jsx: ts.JsxEmit.Preserve,
        strict: false,
      },
    });
}
```

Запускаем проверку TS – добавляем файл

check.ts

```ts
function typecheckStorybookConfig(file: string): { ok: true } | { ok: false; message: string } {
  try {
    const project = new Project({
      compilerOptions: {
        noEmit: true,
        skipLibCheck: true,
        allowJs: true,
        esModuleInterop: true,
        moduleResolution: ModuleResolutionKind.NodeJs,
        resolveJsonModule: true,
        target: ScriptTarget.ESNext,
        module: ModuleKind.ESNext,
        jsx: ts.JsxEmit.Preserve,
        strict: false,
      },
    });
    project.addSourceFileAtPath(file);
}
```

Запускаем проверку TS – запускаем диагностику



Запускаем проверку TS – ищем ошибки

run.ts

```ts
const relevant = project.getPreEmitDiagnostics()
.filter((d) => {
        const code = d.getCode();
        return code !== 2307 && code !== 2792;
    });
```

Запуск миграций



Клонируем все репы

Запуск миграций

Парсим папку с миграциями
и загружаем их



Клонируем все
репы

Запуск миграций

Парсим папку с миграциями
и загружаем их



Клонируем все
репы

Нужную миграцию применяем
к каждой репе

Запуск миграций



Клонируем все репы

Парсим папку с миграциями и загружаем их

Нужную миграцию применяем к каждой репе

Создаем MR с изменениями

Запуск миграций



Клонируем все репы

Парсим папку с миграциями и загружаем их

Нужную миграцию применяем к каждой репе

Создаем MR с изменениями

Вспоминаем, что мы фронтендеры и наводим красоту

Создание MR

run.ts

```
const url = `${GITLAB_HOST}/api/v4/projects/${projectId}/merge_requests`;

const response = await gitlabFetch(url, {
  method: "POST",
  headers: {
    "PRIVATE-TOKEN": accessToken,
    "Content-Type": "application/json",
});
```

Стартуем миграцию через UI Gitlab

Variables

Using `inputs` to control pipeline behavior offers improved security and flexibility. Consider updating your pipelines to use `inputs` instead.

Go to the pipeline editor | Learn more

Variable	NPM_CONFIG_REGISTRY	https://nexus.tcsbank.ru/repository/npm-all/

Конкретно для установки esbuild с флагом --no-optional - нужно явно указать registry

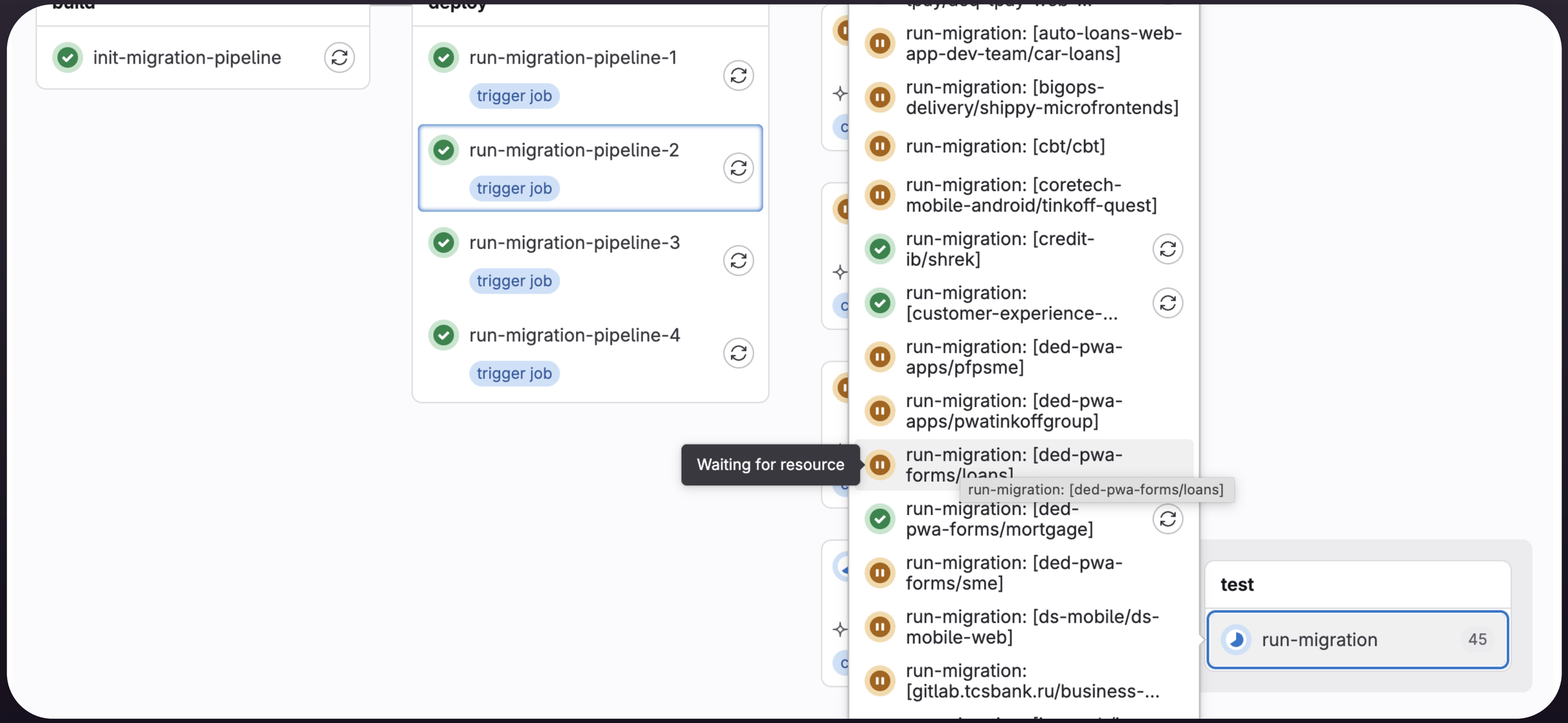
Variable	MIGRATION_NAME	SUPER-CODE
Variable	MIGRATION_REPOS	all
Variable	ACCESS_TOKEN	secret
Variable	Input variable key	Input variable value

Specify variable values to be used in this run. The variables specified in the configuration file as well as CI/CD settings are used by default. Learn more

Variables specified here are **expanded** and not **masked.**

New pipeline | Cancel

Параллельный запуск для скорости



Добавляем UI в Партизан

Список кодмодов

Выберите задачу

Показать кодмоды без открытых MR

Сбросить фильтры

Задача	Описание
BLOCKSIB-3390	Новая функциональность: Подключение актуальных версий sensitive
HUBERT-0000	Рефакторинг: Добавляем src/api-methods/**/* в workspaces package.json
HUBERT-1797	Рефакторинг: Подготовка к React 19
HUBERT-1844	Рефакторинг: подключение общего конфига renovate от hubert
HUBERT-2032	Рефакторинг: Migration: Replace and remove dp auth commands in pipelines
HUBERT-2047	Рефакторинг: change rules for run contract tests
HUBERT-2048	Рефакторинг: Добавление when: manual в джобу net-access-mr
HUBERT-2054	Рефакторинг: update package rules in renovate config
HUBERT-2120	Рефакторинг: Нестрогие зависимости для платформенных пакетов
HUBERT-2146	Рефакторинг: update package rules in renovate config
HUBERT-2176	Рефакторинг: подключает skipper к скриншотному тестированию
HUBERT-2371	Новая функциональность: Добавление в skipper.config.js директории API-методов в run_affected
HUBERT-2400	Рефакторинг: Подключения spa-переходов в e2e тестах
HUBERT-2413	Исправление: Поднимает версию подключаемых джоб, для исправления проблем доступа к образу kafka
HUBERT-2429	Новая функциональность: запуск SSR-тестов на всех сторях кроме info/docs/a11y
HUBERT-2447	Рефакторинг: добавление @swc/jest в jest тесты

Добавляем UI в Партизан

Общая информация

ID задачи: HUBERT-2371

Описание: Новая функциональность: Добавление в skipper.config.js директории API-методов в run_affected

Автор: group_70240_bot

Статистика MR

Всего MR: 101 | Замержено: 62 | Открыто: 32 | Успешных: 3 | Упавших: 29 | Закрыто: 7

Напомнить о MR

Информация о MR-ах

Статус MR | Статус Pipeline | С открытыми тредами | Сбросить фильтры

Репозиторий	MR	Статус MR	Статус pipeline	Открытые треды
pwa-ib/transfer-to-request	https://gitlab.tcsbank.ru/pwa-ib/transfer-to-request/-...	Merged	Success	Нет тредов
finhealth-web/finhealth-web-microblocks	https://gitlab.tcsbank.ru/finhealth-web/finhealth-web...	Merged	Success	Нет тредов
nonclients-web/blocks	https://gitlab.tcsbank.ru/nonclients-web/blocks/-/me...	Merged	Success	Нет тредов



Партизан боролся с Gitlab

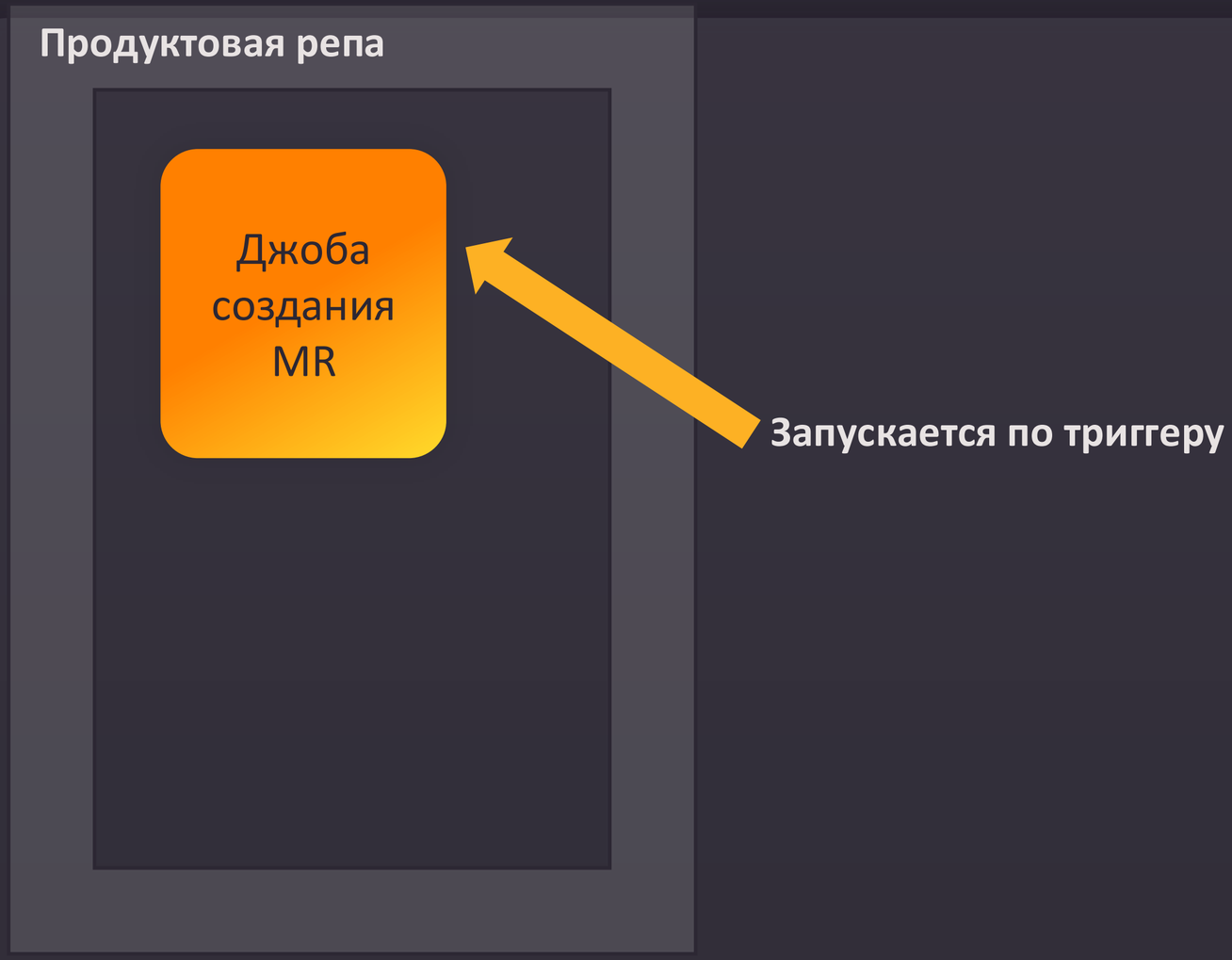
Из-за массового запуска мр-ов под одной личной учеткой их банил Gitlab

Пусть все делается на уровне команд

Перенос запуска в команды



Перенос запуска в команды



Перенос запуска в команды

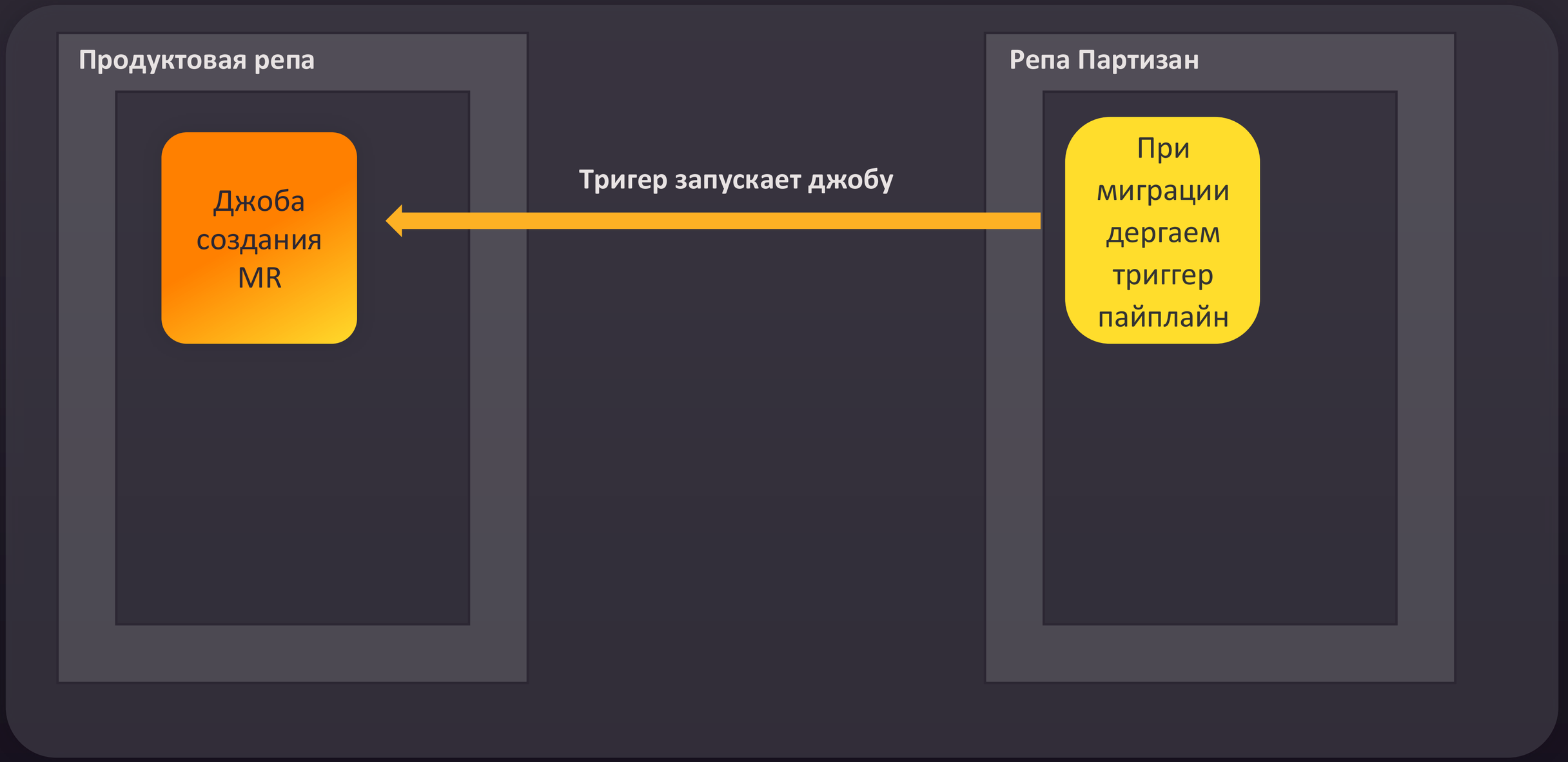
Продуктовая репа

Джоба создания MR

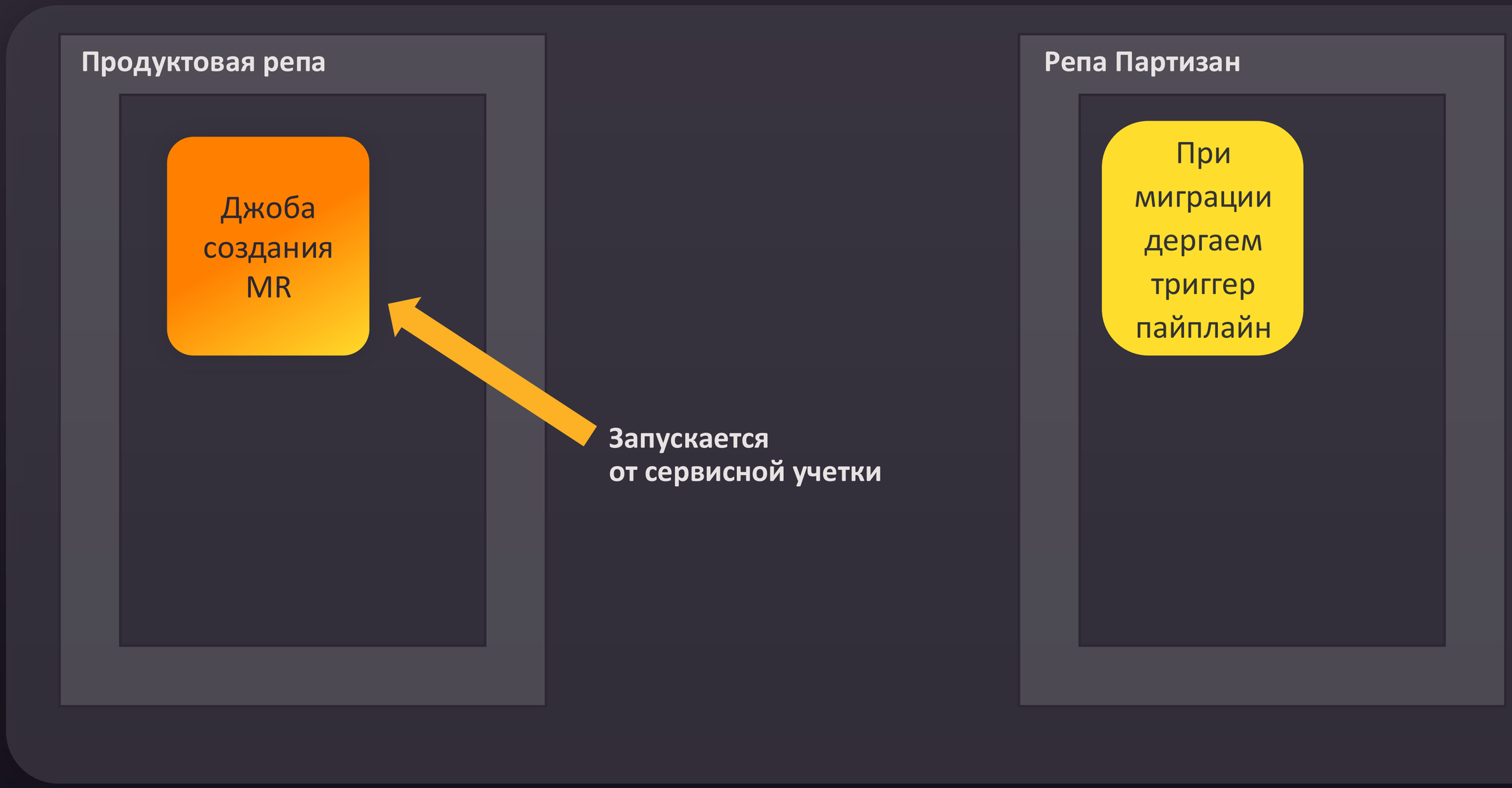
Репа Партизан

При миграции дергаем триггер пайплайн

Перенос запуска в команды



Перенос запуска в команды



Партизан триггерит пайплайн

check.ts

```ts
export const triggerPipeline = async ({
  projectPath,
  accessToken,
  body = {},
}: {
  projectPath: ProjectPath;
  accessToken: AccessToken;
  body: unknown;
}) => {
  const projectId = await repoPathToId(projectPath, accessToken);

  const url = `${GITLAB_HOST}/api/v4/projects/${projectId}/pipeline?ref=master`;

  reportSignale.debug("Trigger pipeline", url, body);

  const response = await gitlabFetch(url, {
    method: "POST",
    headers: {
      "PRIVATE-TOKEN": accessToken,
      "Content-Type": "application/json",
    },
    body: JSON.stringify(body),
  });
}
```

Проверяем, что пайп запустился

check.ts

```
const PIPELINE_FINAL_STATUSES: PipelineStatus[] = ["success", "failed", "canceled", "skipped"];

    for (let i = 0; i < PIPELINE_ITERATIONS_LIMIT; i++) {
      const delay = PIPELINE_APROXIMATE_DURATION * (i + 1);
      await sleep(delay);
      const pipelineInfo = await getPipeline({
        accessToken,
        projectPath,
        pipelineId: createdPipeline.id,
      });
      reportSignale.debug(`Pipeline after ${delay} ms`, pipelineInfo);
      if (pipelineInfo && PIPELINE_FINAL_STATUSES.includes(pipelineInfo.status)) {
        break;
      }
    }
```

Джоба в CI команд открывает MR

run.yml

```yaml
platform-create-mr:
  rules:
    - if: $PLATFORM_MR_SOURCE_BRANCH
  script:
    - |
      git push \
        -o merge_request.create \
        -o merge_request.target="${TARGET_BRANCH}" \
        -o "merge_request.title=${MR_TITLE}" \
        "https://oauth2:${ACCESS_TOKEN}@gitlab.tcsbank.ru/${CI_PROJECT_PATH}.git" \
        "$PLATFORM_MR_SOURCE_BRANCH" \
        --force \
        --no-verify
```

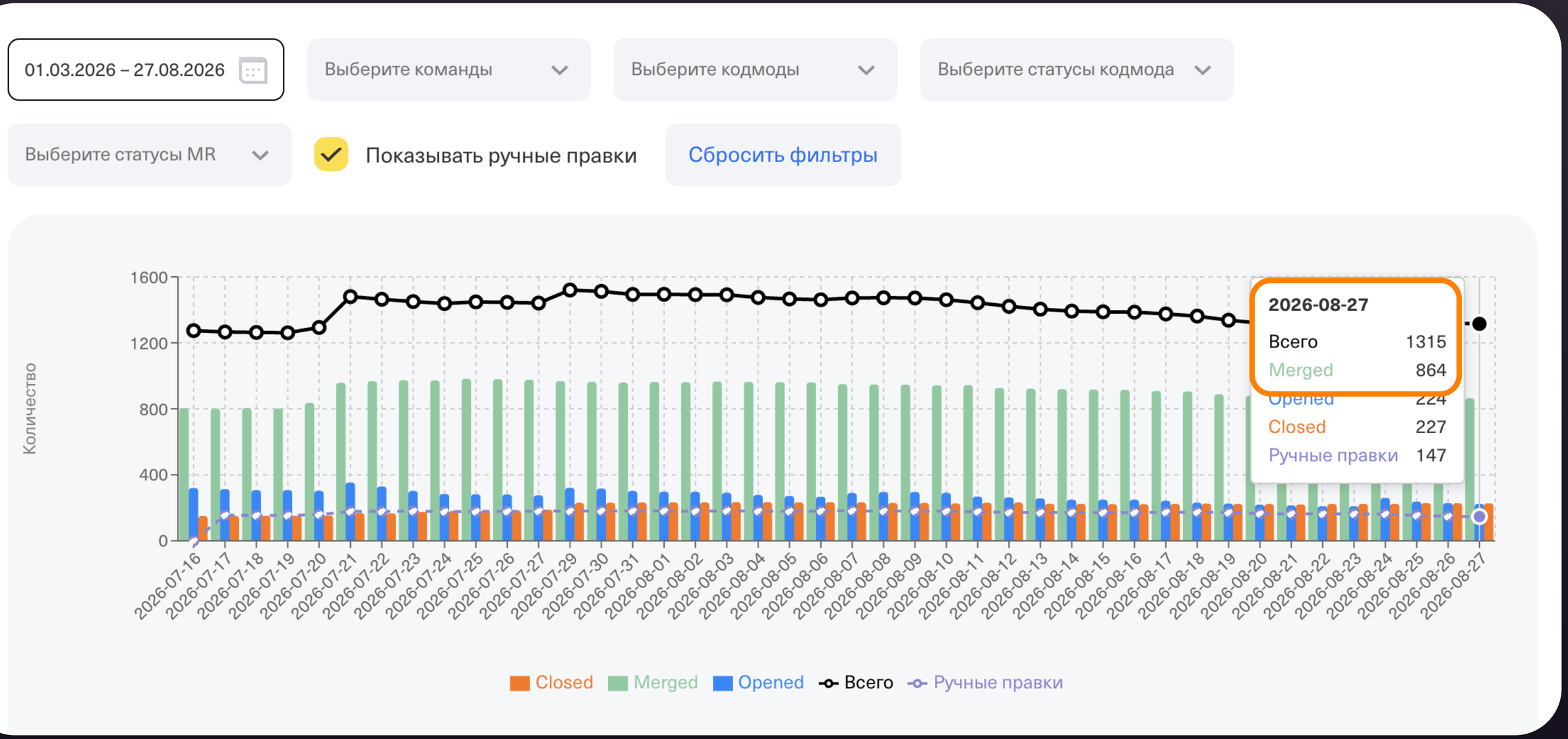
Проверяем, что MR открылся

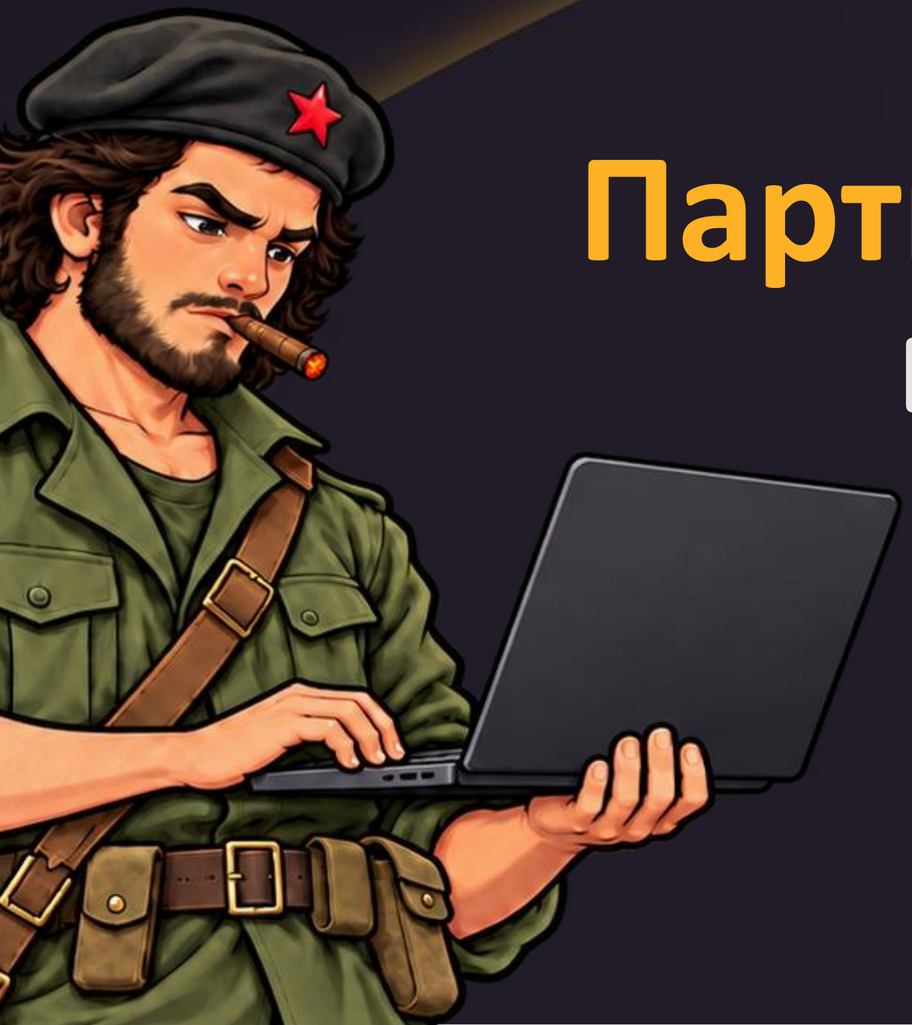
check.ts

```ts
const mr = (await getMergeRequests({ accessToken, projectPath, branchName }, 10))
      .filter((el) => el.source_branch === branchName)
      .at(0);

    if (!mr) {
      throw new Error(`MR с ветки ${branchName} не найден!`);
    }
```

Умеем раскатывать массовые изменения кода





Партизан внедрился в процессы

Связь технического качества с продуктовым

Техническое качество

Как считаем

Процент успешных/все проверки = процент соответствия тех. качеству

Продуктовое качество

Как считаем

Есть SLA на количество багов на задачу, скорость фикса багов. Выполненные требования/все требования = процент качества

Связь технического качества с продуктовым

Рейтинг команд

Название команды	Общий %	Техчеки %	Зависимости %	Кодмоды %	Рейтинг качества %
ib-edd-main	70.5	63.5	86	100 (12/12)	40 (2/5)
ib-invest-pension	52.2	51.9	75.4	68.8 (11/16)	40 (2/5)
ib-myhome	65.3	67.3	75.7	85 (17/20)	60 (3/5)
ib-invest-investbox	89.3	86.5	97.1	100 (12/12)	80 (4/5)
PWA CBT	82.7	86.5	82.8	87.5 (14/16)	80 (4/5)
PWAIB_SHOPPING_AUTO_PARTS	80.8	80.8	0	0 (0/0)	80 (4/5)
ib-bravo	74.2	73.4	76.8	100 (13/13)	80 (4/5)
ib-jobboard	73.4	69.2	85.9	100 (1/1)	80 (4/5)
ib-mediaplatform	69.1	71.2	61.3	0 (0/0)	80 (4/5)
ib-spend-advisor	68.3	71.2	65.1	80 (4/5)	80 (4/5)
PWAIB_MATCH	68.1	78.8	71.1	50 (6/12)	80 (4/5)
ib-socialweb	65.9	59.6	83.1	100 (4/4)	80 (4/5)
ib-cartman	63.6	59.4	78.4	80 (4/5)	80 (4/5)

Прозрачность исправлений

Прозрачность исправлений

teams.json

```
{
    "Title": "PWA Debit",
    "Git": "https://gitlab.tcsbank.ru/pwa-debit/debit-ib",
    "Jira": "https://jira3.tcsbank.ru/secure/RapidBoard.jspa?rapidView=300278&projectKey=ODC
    "Меншен дежурного в Time": "@debit-web-duty"
}
```

Линкуем задачи

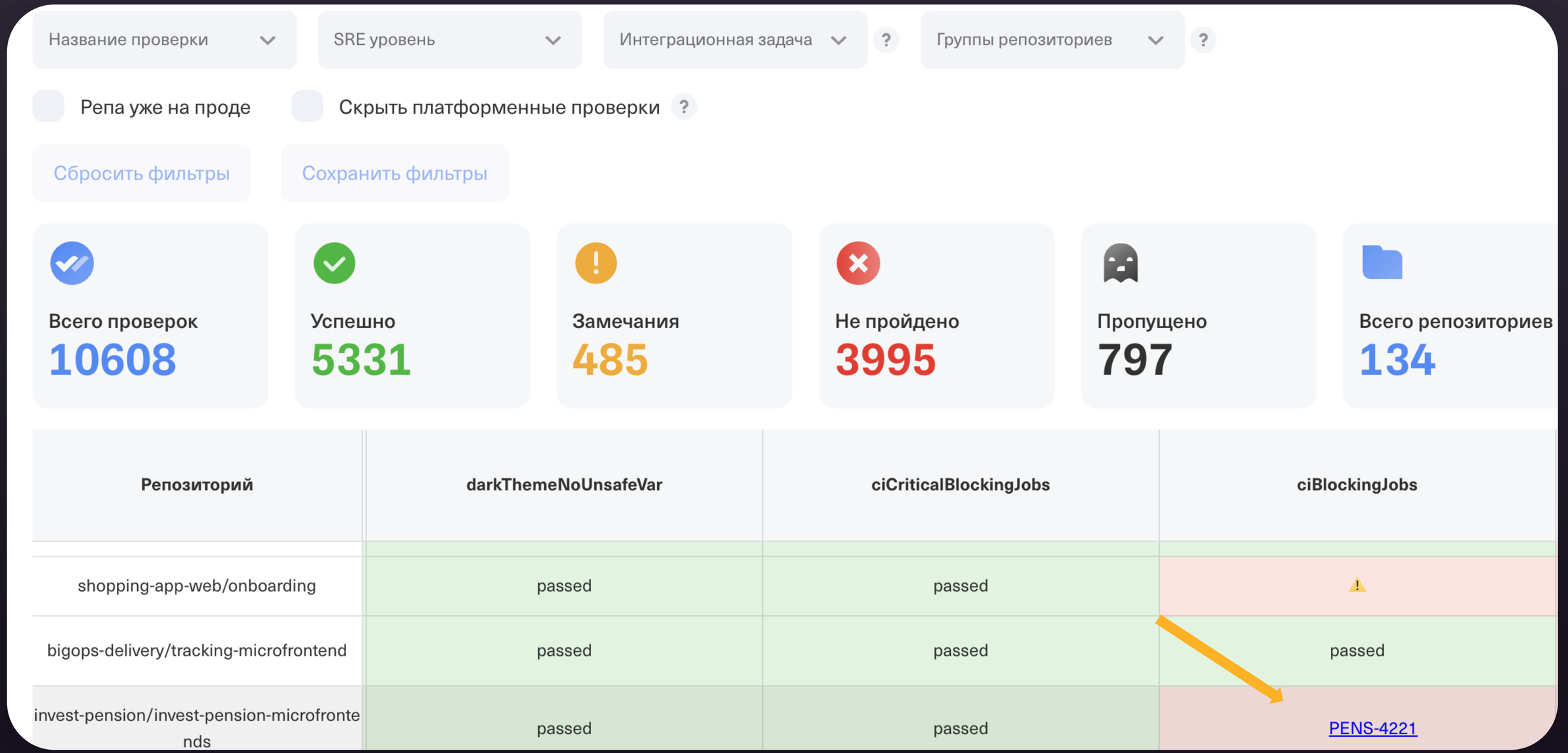
Сводка | Техчеки | Миграции | Зависимости | Рейтинг качества | Использование блоков

Выберите проверки | Выберите группы | Выберите категории | Показать пройденные | Скрыть внутренние проверки

Сбросить фильтры

Проверка	Описание	Импакт	Статус	Категория	Вес	Jira
validPackageVersio...	Проверка актуальности версий ключевых пакетов (@hubert, @tools-for-blocks, @boxy)	Использование устаревших версий пакетов может привести к отсутствию новых фич и багфиксов	failed	code-quality	0	Для линковки назначьте на задачу метки: `partisan-checks, validPackageVersions`
notifications-check...	Подключен ли notifier из micro di	Иначе нотификации могут накладываться друг на друга	failed	ui	2	Для линковки назначьте на задачу метки: `partisan-checks, notifications-checks`
notifications-check...	Используется ли NotificationStack	Иначе нотификации могут накладываться друг на друга	failed	ui	2	Для линковки назначьте на задачу метки: `partisan-checks, notifications-checks`
		Пакеты @hubert/atom-* не должны использоваться напрямую в продуктовых репозиториях. Перейдите на				

Линкуем задачи



Регулярно пушим

Автоматические напоминания о миграциях

- lfsweb/checkout — !650 — @Roman Karandashov

📦 HUBERT-2413 — Исправление: Поднимает версию подключаемых джоб, для исправления проблем доступа к образу kafka (15 MR)

- edd-main/blocks — !1237 — @wintm-dev-duty
- insurance-web/kasko — !305 — @ins-web-kasko-duty
- invest-frontend/invest-advisory-boxy — !70 — @invest-catalog-duty
- insurance-web/property — !11 — @ins-web-property-duty
- mb-origination-welcome/welcome-onboarding-web — !266 — @Aleksandr Utbanov
- my-payments-core/mpc-web — !348 — @Boris Kunshenko , @Aleksey Skakun
- orig-web-vip/orig-wm-ib — !36 — @pwapremium
- bigops-delivery/shippy-microfrontends — !162 — @parceltracking_duty
- nfs-ls-tauto/tauto-supreme-microfrontends — !140 — @Roman Starostin
- pwa-ib/fry — !4276 — @pwa-ib-fry-duty
- bigops-delivery/tracking-microfrontend — !269 — @parcel-tracking-front
- invest-pension/invest-pension-microfrontends — !924 — @invest-pension-duty
- invest-frontend/invest-margin-boxy — !90 — @inv_pro_web_duty
- mfo-pwa/mfo-pwa — !149 — @Sergey Dmitriev
- travel-growth-inspiration/travel-inspiration-web — !94 — @Evgeniy Rasskazov

Когда увидели сообщение — поставьте реакцию ❤️ к сообщению

PM ⚠️ 🔔 Открытые MR-ы по кодмодам (11 кодмодов, 178 MR) [5/8]

📦 HUBERT-2447 — Рефакторинг: добавление @swc/jest в jest тесты (14 MR)

- top-cash-study-loans/study-ib-web — !374 — @Anton Novikov
- pwa-ib/bank — !436 — @-

Автоматические напоминания о техчеках

Elena Popova 5:08 PM

Привет, это напоминание о техчеке! #techcheck_reminder
Техчек: hasResolutions
Ссылка: https://pwa-ib-partisan.s3-msk.tinkoff.ru/partisan-dashboard/index.html#/techchecks/hasResolutions

Когда увидели сообщение — поставьте реакцию ❤️ к сообщению.
В тред, пожалуйста, скиньте заведенные задачи на исправление.

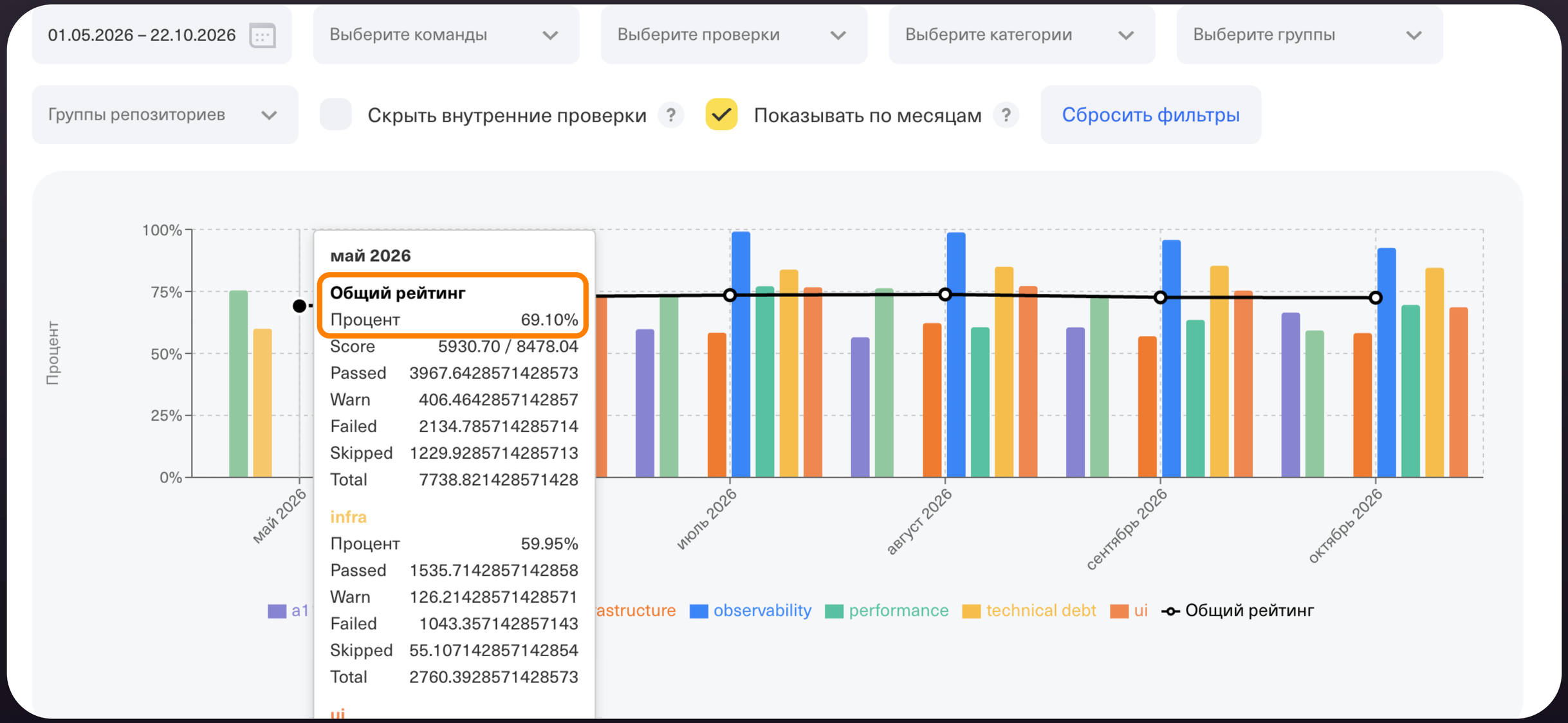
Список репозиториев:

- ❌ credit-ib/bravo @bnplmobileduty
- ❌ gta/gta-web @gta-ib
- ❌ tigr/tigr-web
- ❌ t-wallet/t-wallet @t-wallet-duty
- ❌ invest-terminal/crypto-terminal-mobile @Vyacheslav Goreev
- ❌ gtech-restrictions-ib-web/ib-restrictions-front @Egor Kalinin
- ❌ pwa-ib/pumba @ib-operations-dev-duty
- ❌ credit-ib/morty @trnnweb
- ❌ dibs/black @debit-product-web-duty
- ❌ dibs/saving @deposits-web-dev-support-duty

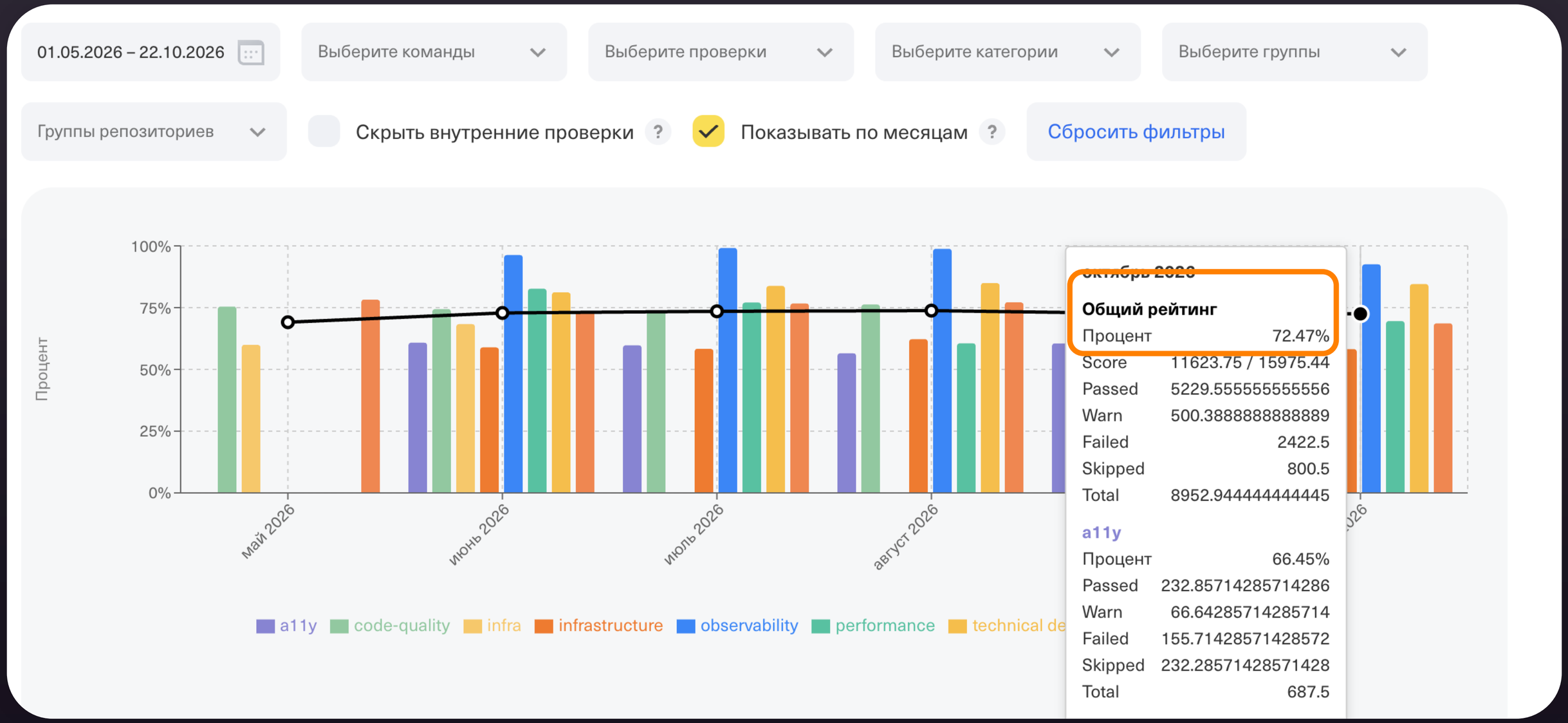
o IB: architecture

Мониторим в динамике

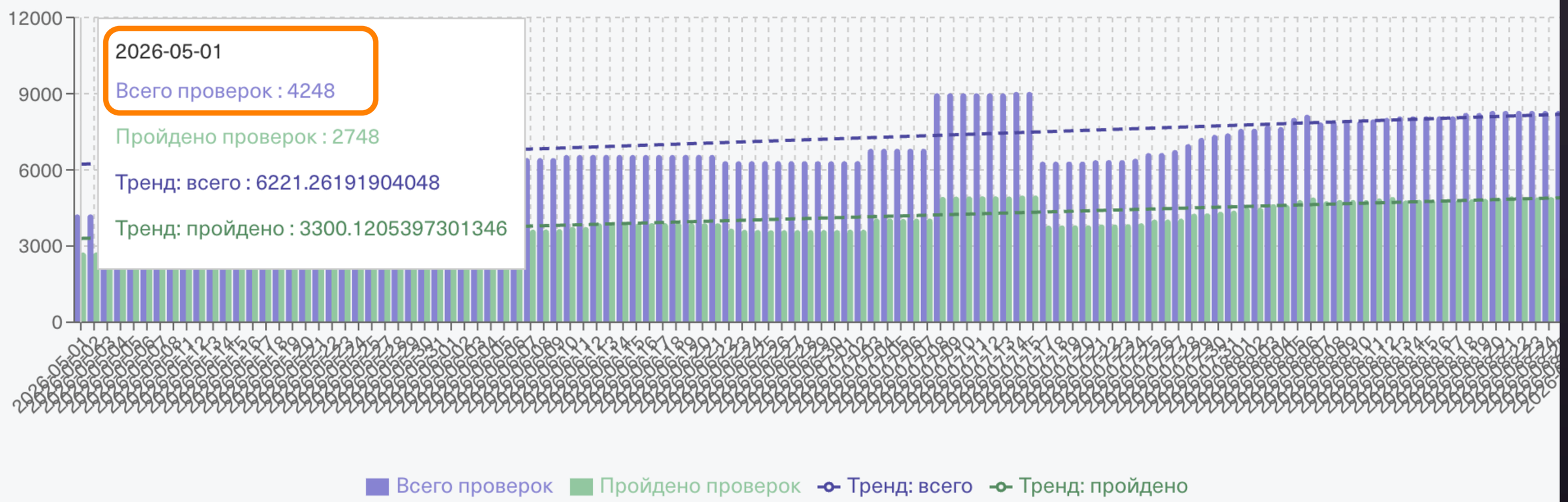
Статистика по тех качеству



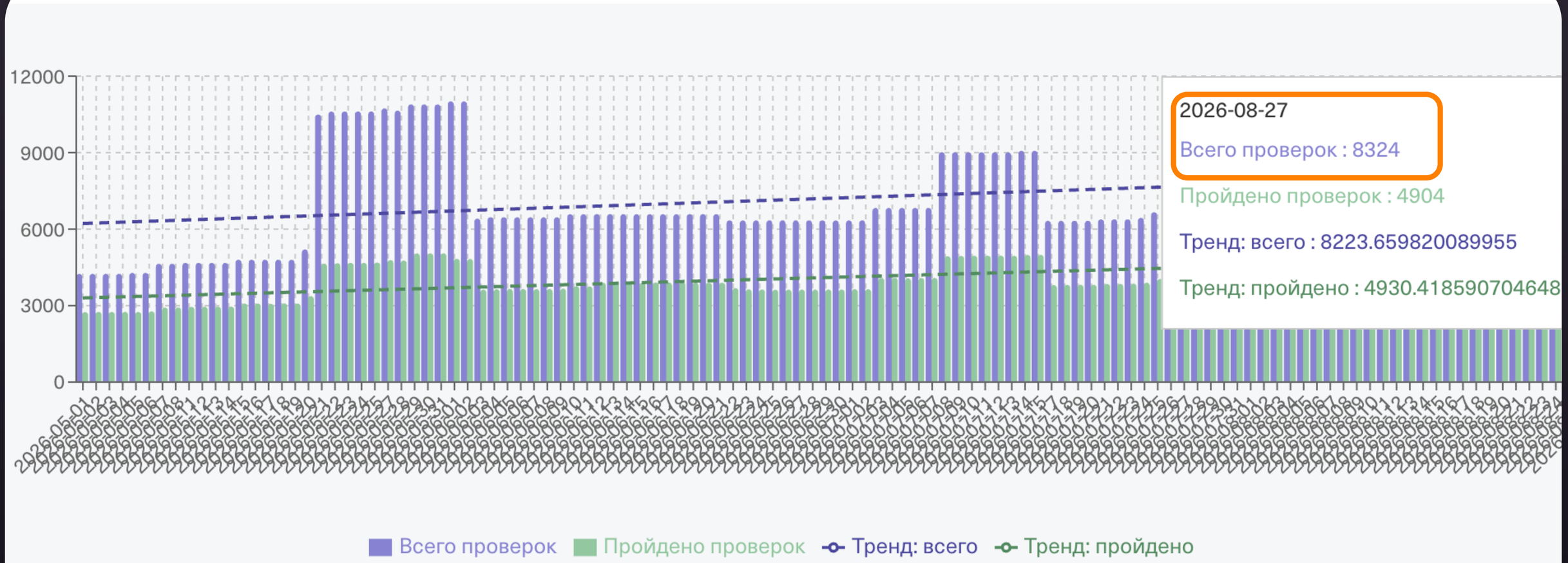
Статистика по тех качеству



Статистика по тех качеству



Статистика по тех качеству



Итого

Умеем определять важный техдолг и подсвечивать его командам

Умеем понимать, что становится с техкачеством с течением времени

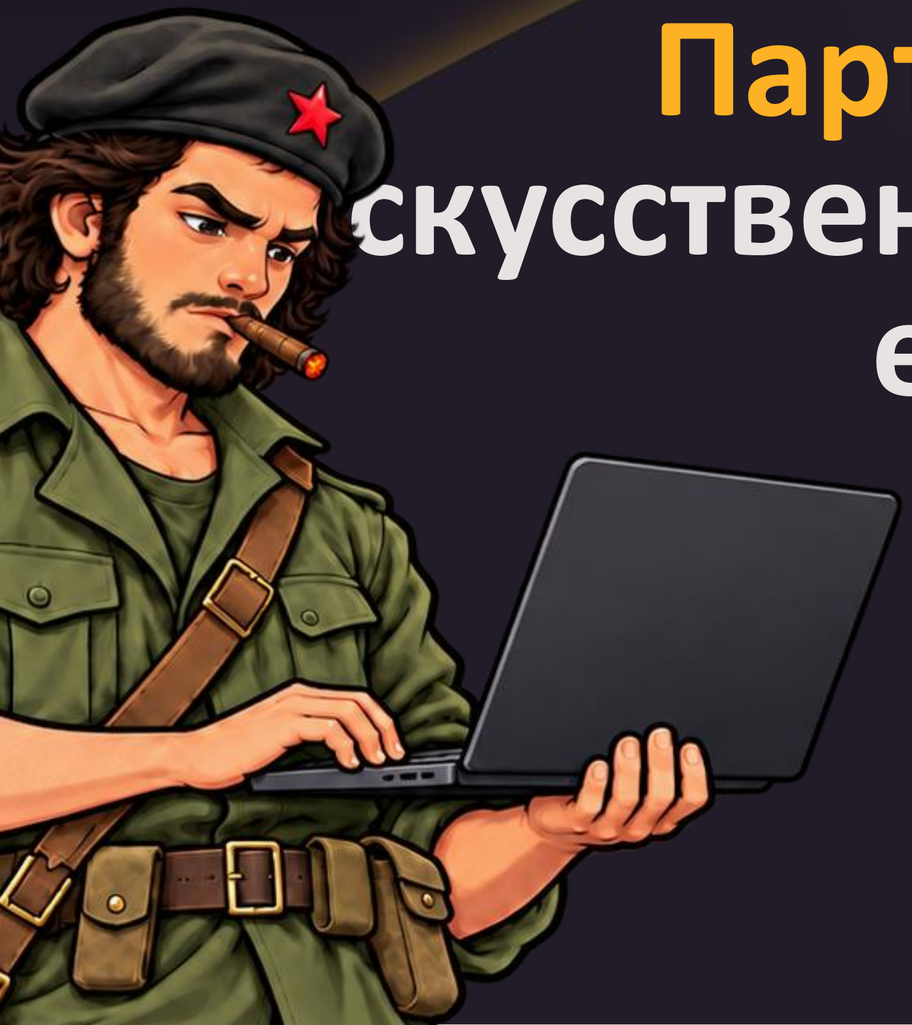
Умеем экономить время команд за счет миграций

Что дальше?

AI !

HR:
где вы видите себя через пять лет?
Я (прикидывая наиболее вероятные события):





Партизан использует скусственный интеллект вместо естественного

Задаем проверку с помощью промта

promt.md

```
You are auditing a codebase for compliance with the navigation rule:

> Все переходы по ссылкам должны осуществляться через `router.navigate` от supreme

All link navigations must use `router.navigate` from the supreme package.

## Task

Verify that this repository complies with the navigation rule.

## Check

Look for:

- `window.location.href`
- `window.location.assign`
- `window.location.replace`
- `location.href`
- `window.open()` for internal navigation
- `<a href="/mybank/...">`
- other direct navigations that should use `router.navigate`
- acceptable exceptions such as external URLs, downloads, `mailto:`, `tel:`, `target="_blank"`

## Output rules

Start with "routerForNavigation_PASSED" if no violations are found, otherwise start with "routerForNavigation_FAILED".

Do not use markdown.
Do not use code fences.
List violations below (and in general explain why they are violations)
```

Скармливаем ее AI

run.ts

```
const result = query({
    prompt: prompt,
    options: {
      cwd: repoDir ?? process.cwd(),
      authType: "openai",
      maxSessionTurns: 1000,
      logLevel: "debug",
      debug: false,
      env: {
        ...process.env,
        OPENAI_API_KEY: getEnv("AI_TOKEN")
      } as Record<string, string>,
    },
});
```

Парсим результат

run.ts

```
const result = String(successResultMessage?.result);
return [
        {
          status: result.includes("routerForNavigation_FAILED")
              ? "failed"
              : result.includes("routerForNavigation_PASSED")
                ? "passed"
                : "warn",
            meta: {
              result,
            },
          },
        ];
```

Все хорошо, но только очень долго для джоб в CI

Поэтому переносим запуск из Gitlab на выделенные песочницы

Но это уже другая история

Вывод

На инженерной смекалочке можно сделать инструмент, который улучшит жизнь 1000+ разработчикам