

Запрыгиваем на релизный поезд. Как мы ускоряли путь задач фронтенда на production

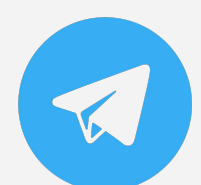
Юрий Голубев

ведущий frontend разработчик в VK

Frontend

FrontOps

Tools



t.me/Ansile



vk.com/ansile

Юрий Голубев

Ведущий frontend разработчик в VK

- 5+ лет работаю в VK над продуктами Mail.ru и решаю различные задачи в команде RnD
- перенёс на новый деплой все проекты Mail.ru
- ускорил процесс деплоя нового проекта фронта в прод с двух недель до 2 дней
- интересуюсь не только фронтенд-разработкой, но и всем, что заходит в смежные области и не очень: от систем сборки до автотестирования

Что будет в докладе

- что мы хотим от деплоя?
- отправная точка
- почему поезд
- путь внедрения в наши проекты
- как понять, что сделали лучше?
- как внедрить у вас без регистрации и смс
- пункт назначения

ЧТО МЫ ХОТИМ?

О проблемах

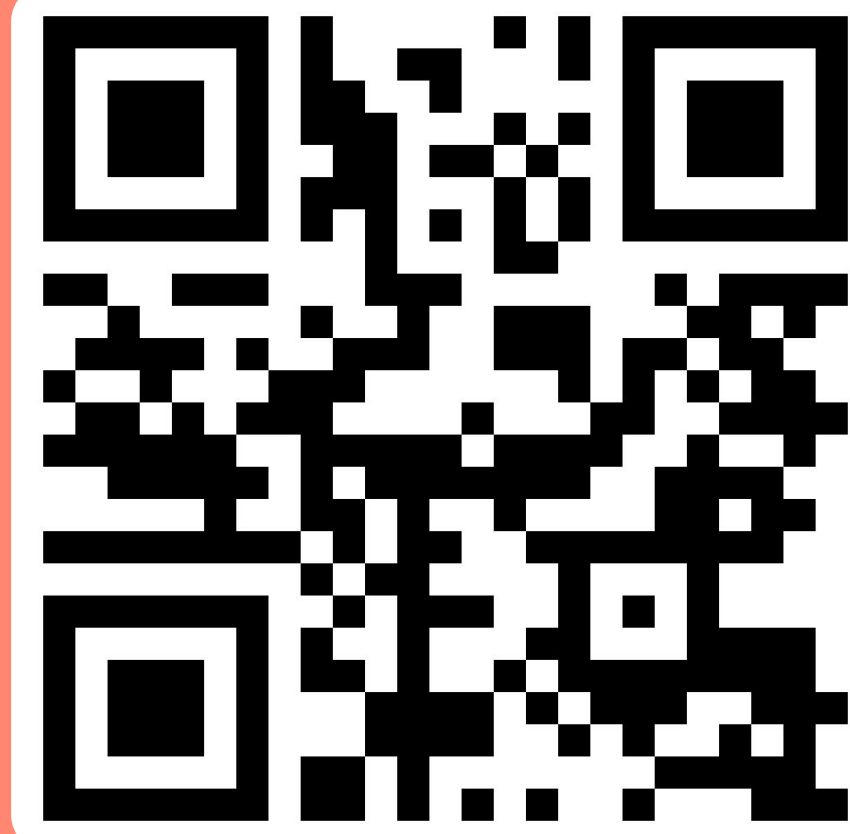
- если мы катимся медленно → цикл выкатки двух задач может занимать 2 недели
- задачи долго простаивают → теряется контекст
- задачи долго лежат и при этом мерзятся другие задачи → больше времени на фикс конфликтов, больше шанс багов из-за конфликтов

Отправная точка



Отправная точка

- Самописная система деплоя —
- Gitlab CI
- Выкатываемся итерациями



Про то, как
было раньше

Критерии выбора решения

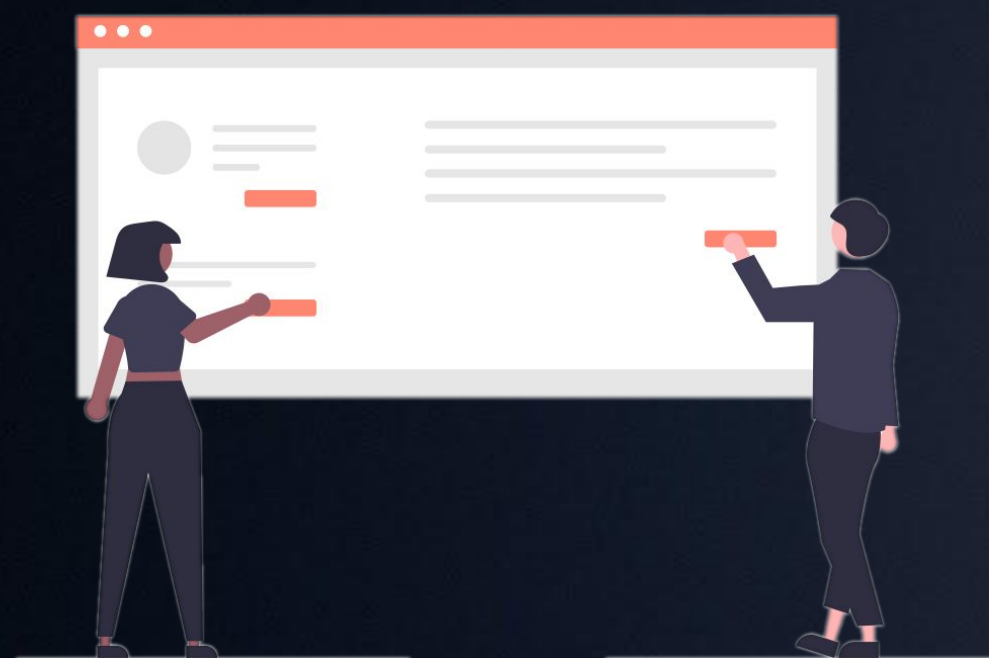
- сохранение качества / uptime
- возможный уровень автоматизации
- время на roundtrip
- риски от человеческой ошибки
- порог вхождения
- время на поиск источника бага в релизе
- время на восстановление нормального deploy flow после отката задачи

Критерии выбора решения



Ограничения

Мы делаем frontend!



Необходимо ручное
тестирование релиза



Сервера не бесконечные



Нужен ручной мониторинг
после выкатки

Варианты

- просто автоматизация сборки релиза (автомерж, etc)
- релизы из мастера по расписанию (батчинг)
- merge train с выкаткой по 1 таске



Автоматизация сборки

- ✓ легко начать (не меняется процесс)
- потеря контекста
- нужны инициативность / доп процесс для скорости
- релиз собрали, а новые задачи ждут

Релизы из мастера по расписанию

- ✔ предсказуемые тайминги
- потеря контекста
- релиз собрали, а новые задачи ждут
- если нашли багу в релизе, процесс встаёт

понятные варианты



поезд...

Что такое merge train?

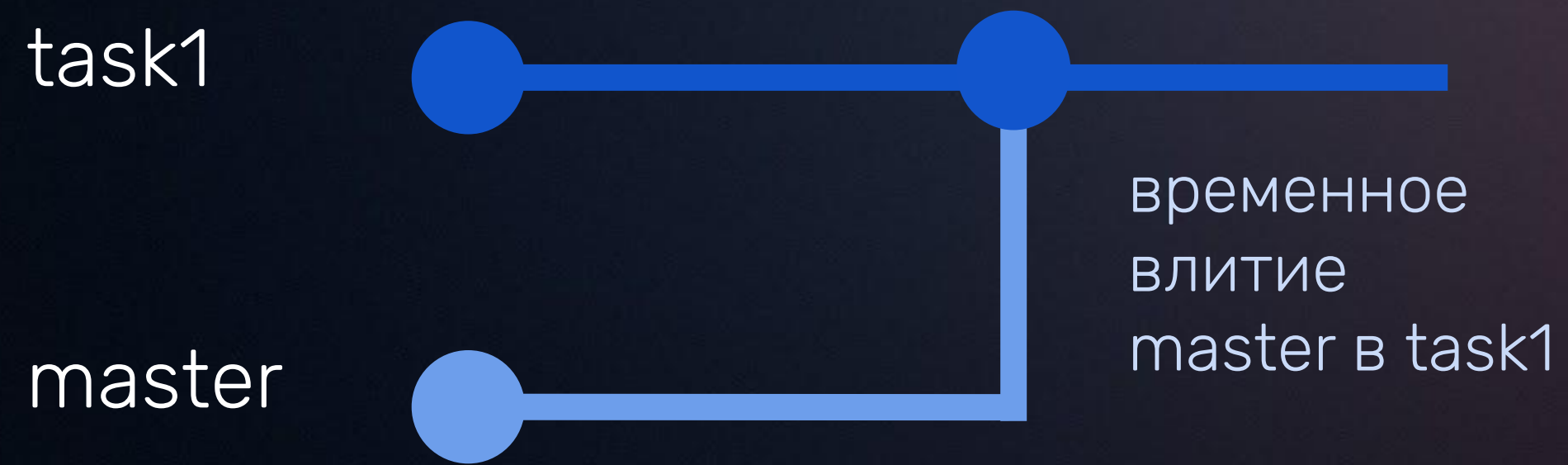
Представим что мы хотим выкатить задачу...

master



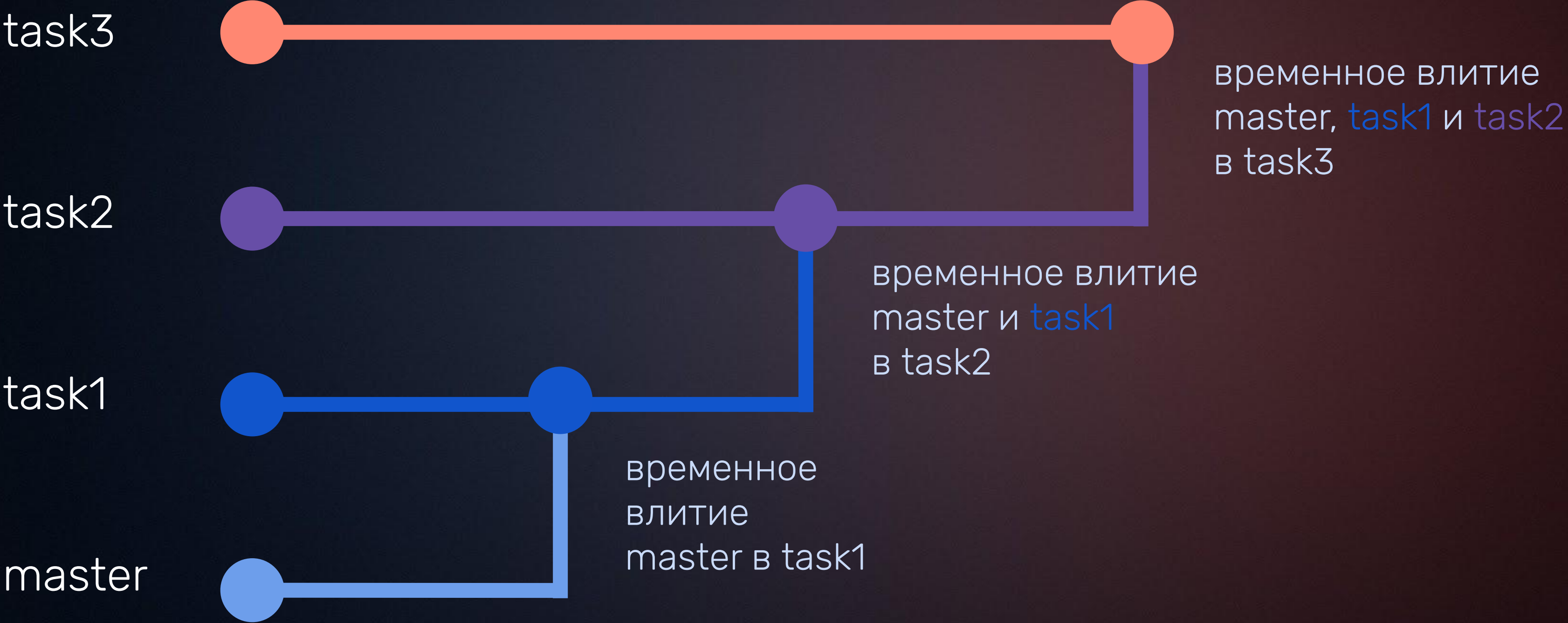
Что такое merge train?

А потом ещё одну...



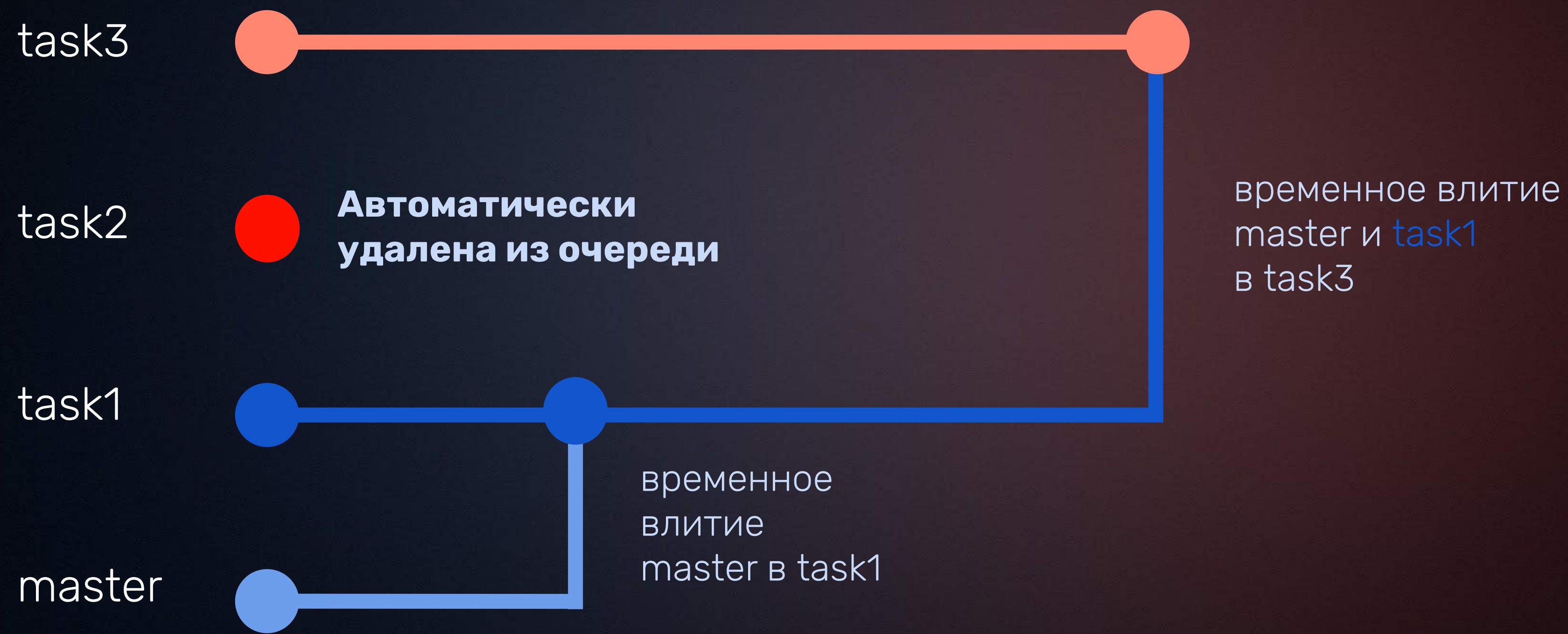
Что такое merge train?

А потом ещё одну...



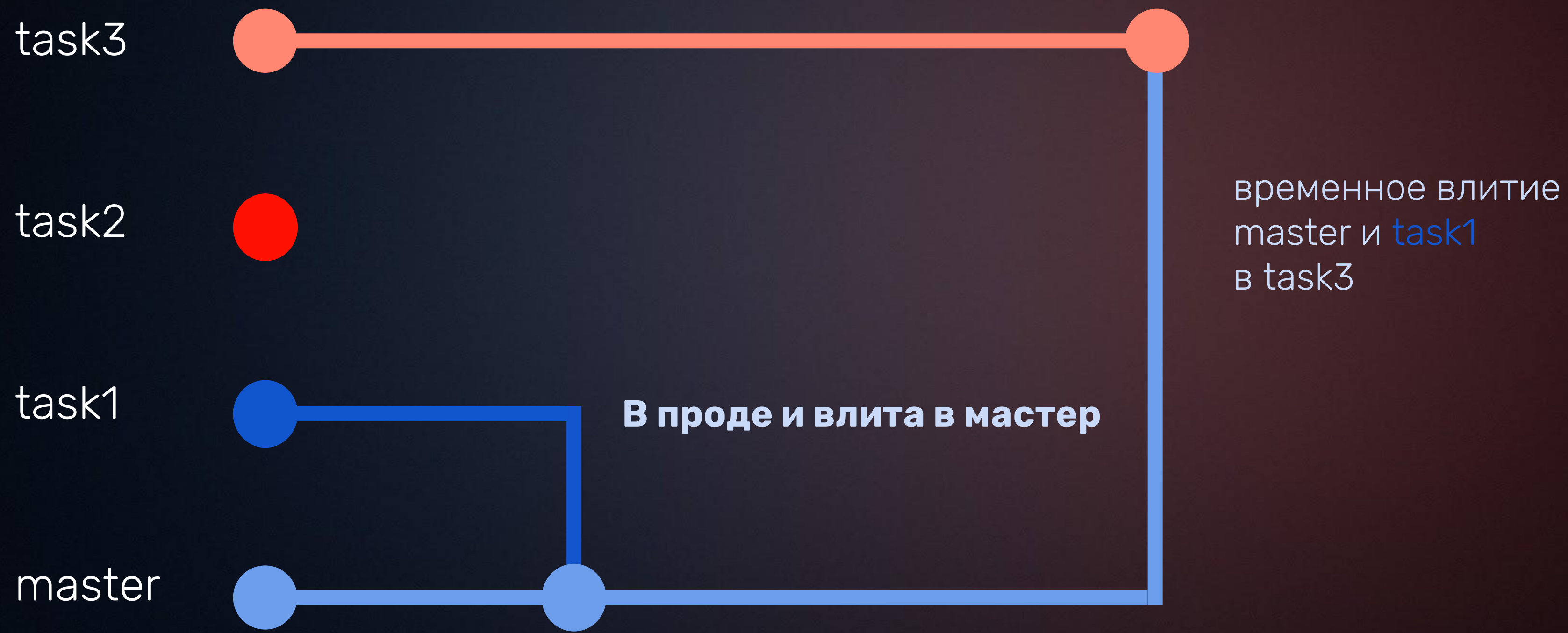
А если что-то пойдет не так?

В задаче 2 упали автотесты :/



Выкатка по одной задаче

В задаче 1 прошли тесты 🎉



Выкатка по одной задаче

В задаче 3 прошли тесты 🎉

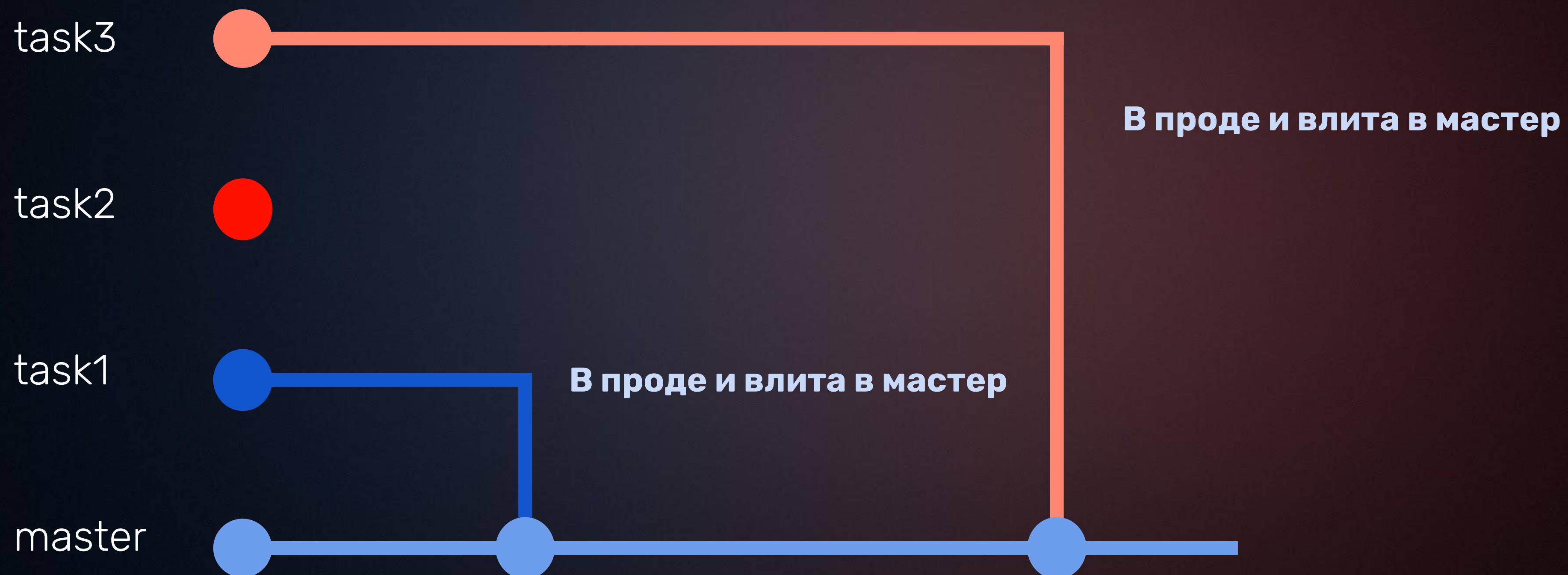


Схема выкатки – до

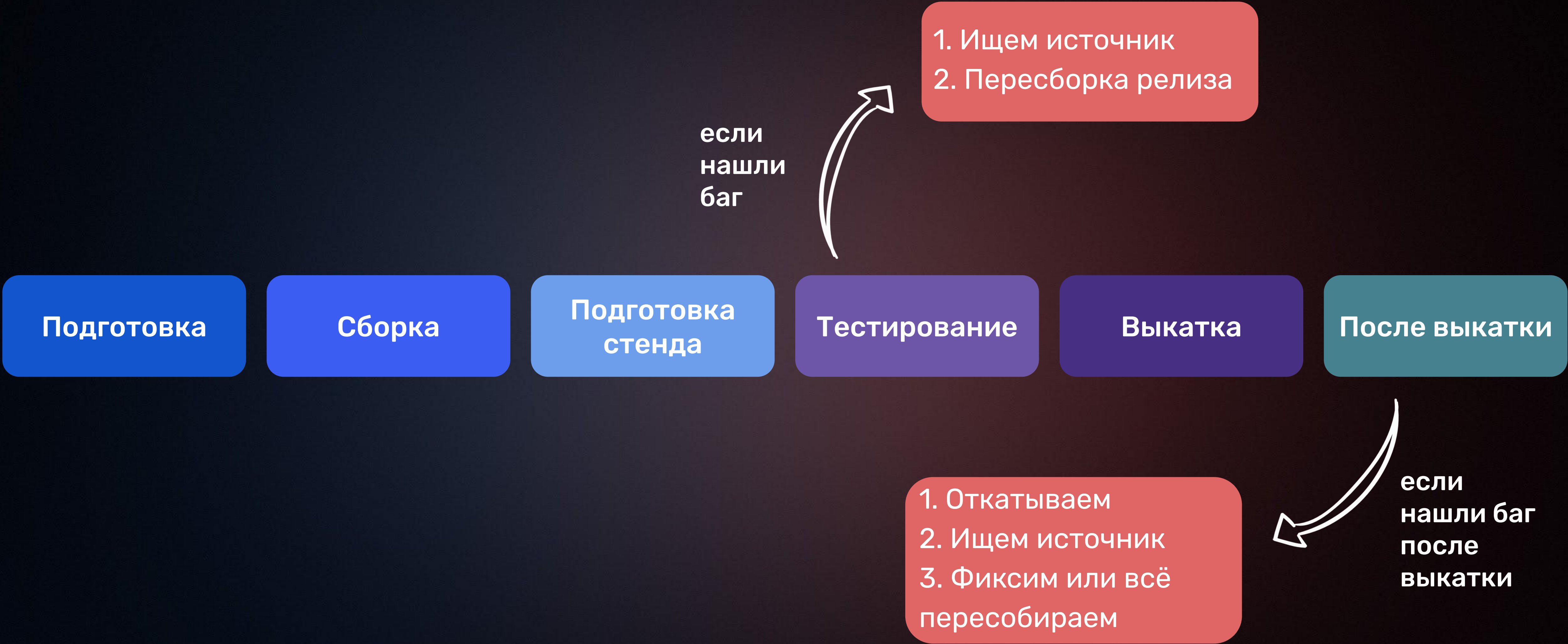
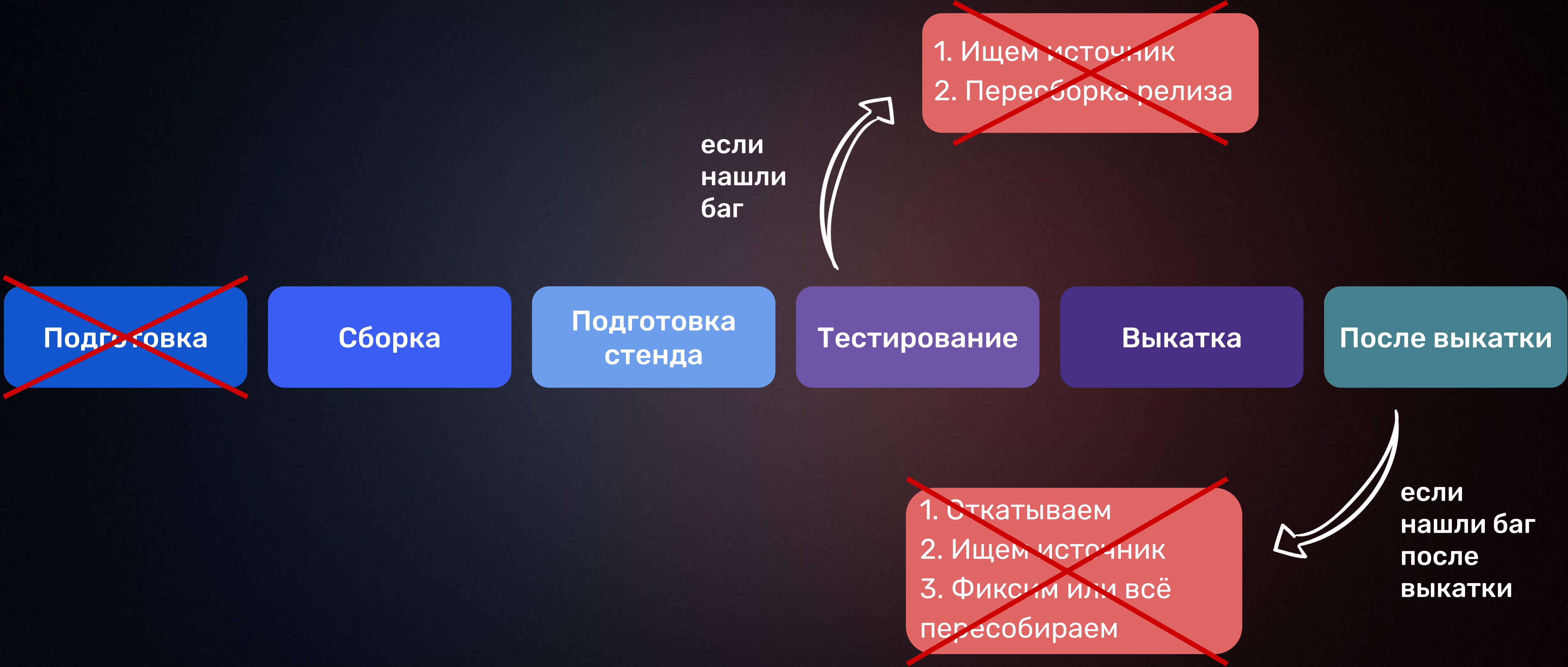


Схема выкатки – что меняется?



~~1. Ищем источник
2. Пересборка релиза~~

если
нашли
баг

~~Подготовка~~

Сборка

Подготовка
стенда

Тестирование

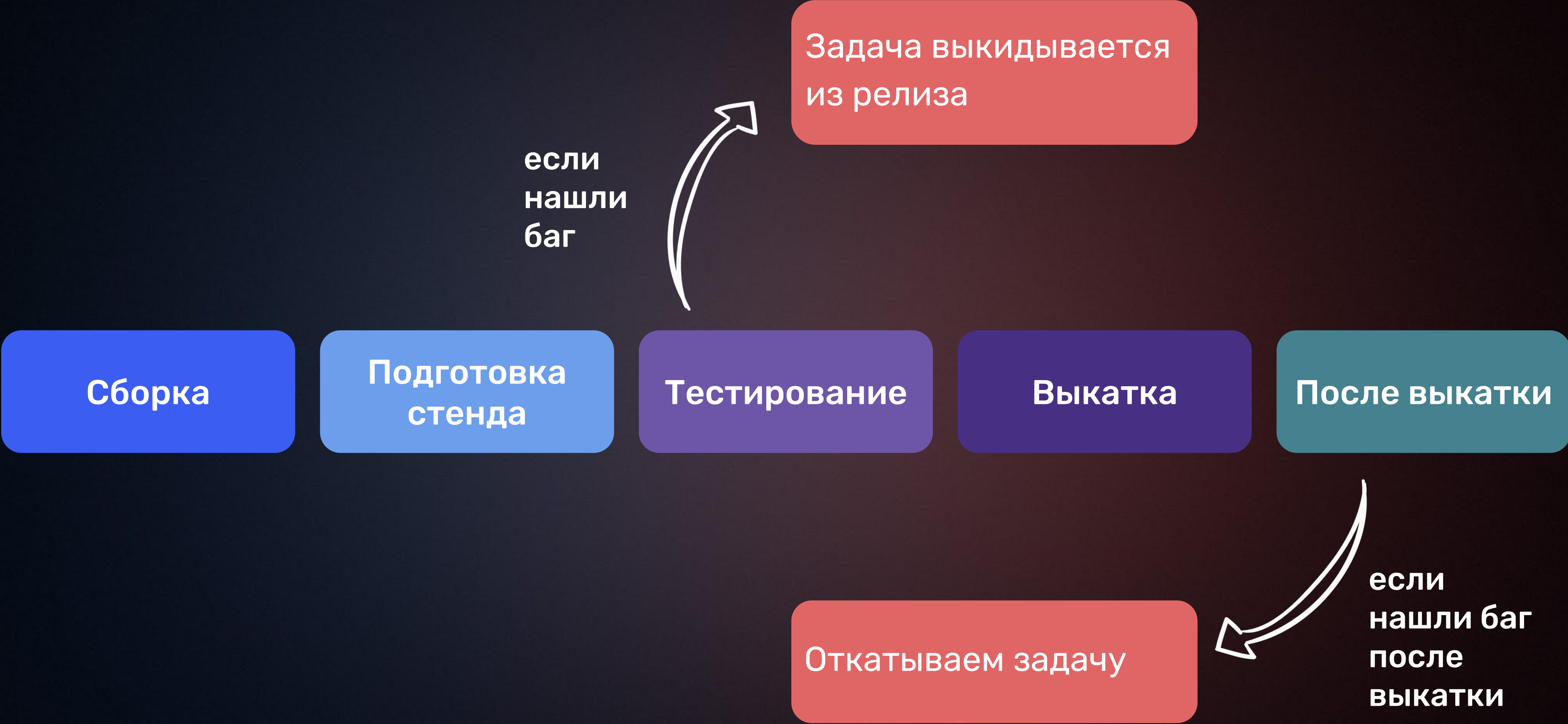
Выкатка

После выкатки

~~1. Откатываем
2. Ищем источник
3. Фиксим или всё
пересобираем~~

если
нашли баг
после
выкатки

Схема выкатки – после



Merge train

- ✓ ответственность за задачу
- ✓ сохранение контекста
- ✓ всегда катим что-то, даже в худшем случае
- нужно много инфраструктуры, чтобы завести
- сбросы браузерного кэша на каждую задачу

История паровозика,
который смог 



Research...



- Хотим быстро всё обкатать и помочь продукту → берем новый проект с быстрым релизным циклом – Заметки
- Хотим минимум ручных действий → нужна интеграция с jira
- Хотим меньше сбросов кэша → нужны чанки и хэши
- **Вроде всё обсудили, можно делать?**

... and Development



Никита

привет!
[ВОТ ТУТ](#)
пайплайн мержа таски в мастер как будто завис

edited 17:34

9

Марина

эх опять застряло
ну ладно подожду как в прошлый раз

19:11

Александр

марин

18:43

может нам всем в тибет?

18:43

карму почистить

18:43

вы собираете все вообще проблемы=)))

18:43

Марина

о да мне точно нада

18:43

Александр

причем это к трейну ноль отношения

18:44

Марина

оно до сих пор не доехало, что делать, кому можно написать?

18:38

Марина

[ВОТ ЭТОТ ПАЙП](#) застрял чего-то
хотя следующий раскатился и бот сообщение прислал

17:39

Где мы свернули не туда? 🤔

Что теперь говорят юзеры?

Александр

Merge Train отличная фича, молодцы что вкрутили, всё пыхтит и вертится просто на ура 💪

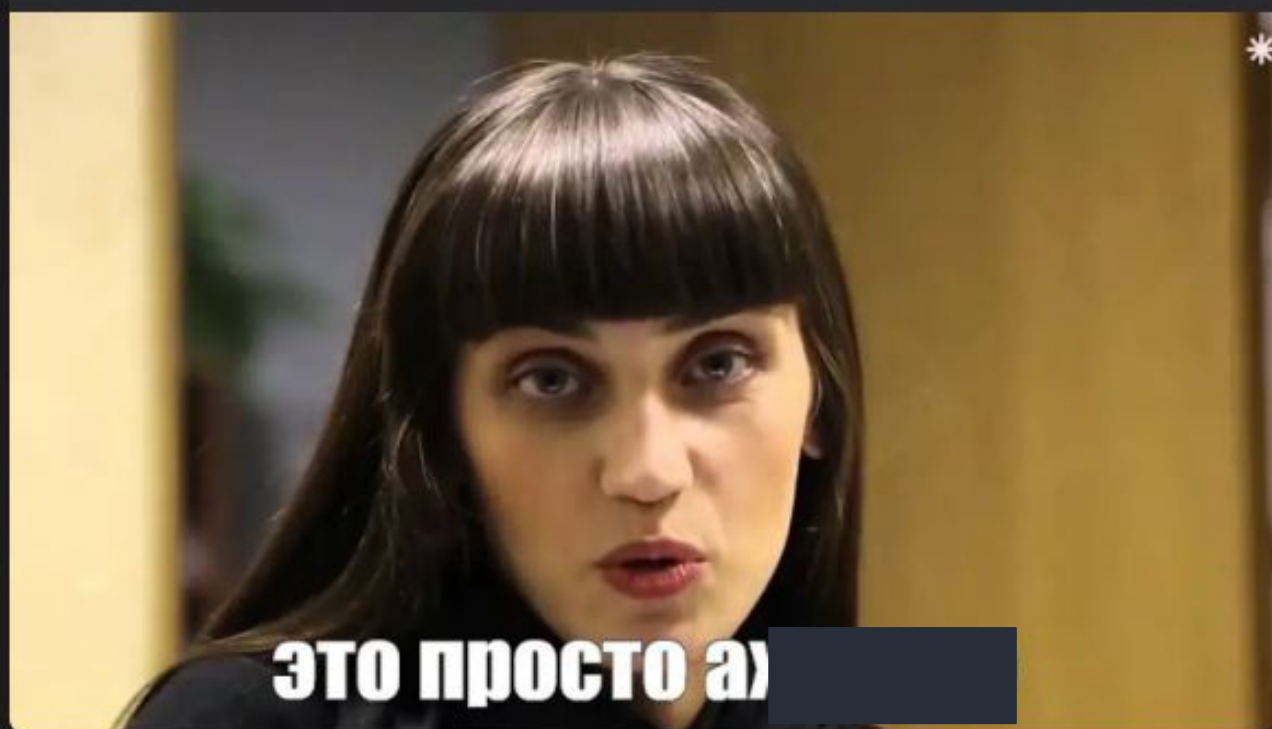
22:45

Герман

Быстро. Четко. Падают автотесты.

22:44

Денис



[Дмитрий Жук](#) [Юрий Голубев](#) спасибо вам большое за это чудо высоких технологий!

11:21



2

Артём

Удобно, что каждый член команды может катить задачи.

Однако, для максимальной эффективности нужно иметь стабильные стенды - тогда все работает как часы

22:44

Сергей

Очень круто)

Раньше в свои задачи нужно было постоянно мастер подливать, в трейне все автоматом подливается. Аналогично с реквестом с тестами, не нужно руками следить и вливать


То есть ушло много рутинной работы, которую ранее надо было руками делать

22:45

Катимся дальше



Облако – ДО

- другой проект в jira
 - куда больше тестов
 - куда больше задач
 - почти* allgreen
 - релизы по 15 задач за раз, могут катиться неделю
- 



zapi добавил(а) комментарий - 2023-12-05 18:36

- Host: cloud27
- Branch: cloudweb-16022
- Total: 5889
- Passed: 5888
- Failed: 1
- Skipped: 0

Reply · Удалить · 🗨️

Failed: 1 19:42

Облако – ПОСЛЕ

- allgreen
- правки (унификация) в jira
- выкатка 13 задач в день (и это не предел)



Никита

Итоги дня

Выкатили:

[CLOUDWEB-16280](#)

[CLOUDWEB-15886](#)

[CLOUDWEB-16405](#)

[CLOUDWEB-16333](#)

[CLOUDWEB-16383](#)

[CLOUDWEB-15825](#)

[CLOUDWEB-16436](#)

[CLOUDWEB-16610](#)

[CLOUDWEB-16362](#)

[CLOUDWEB-16374](#)

[CLOUDWEB-16454](#)

[CLOUDWEB-16551](#)

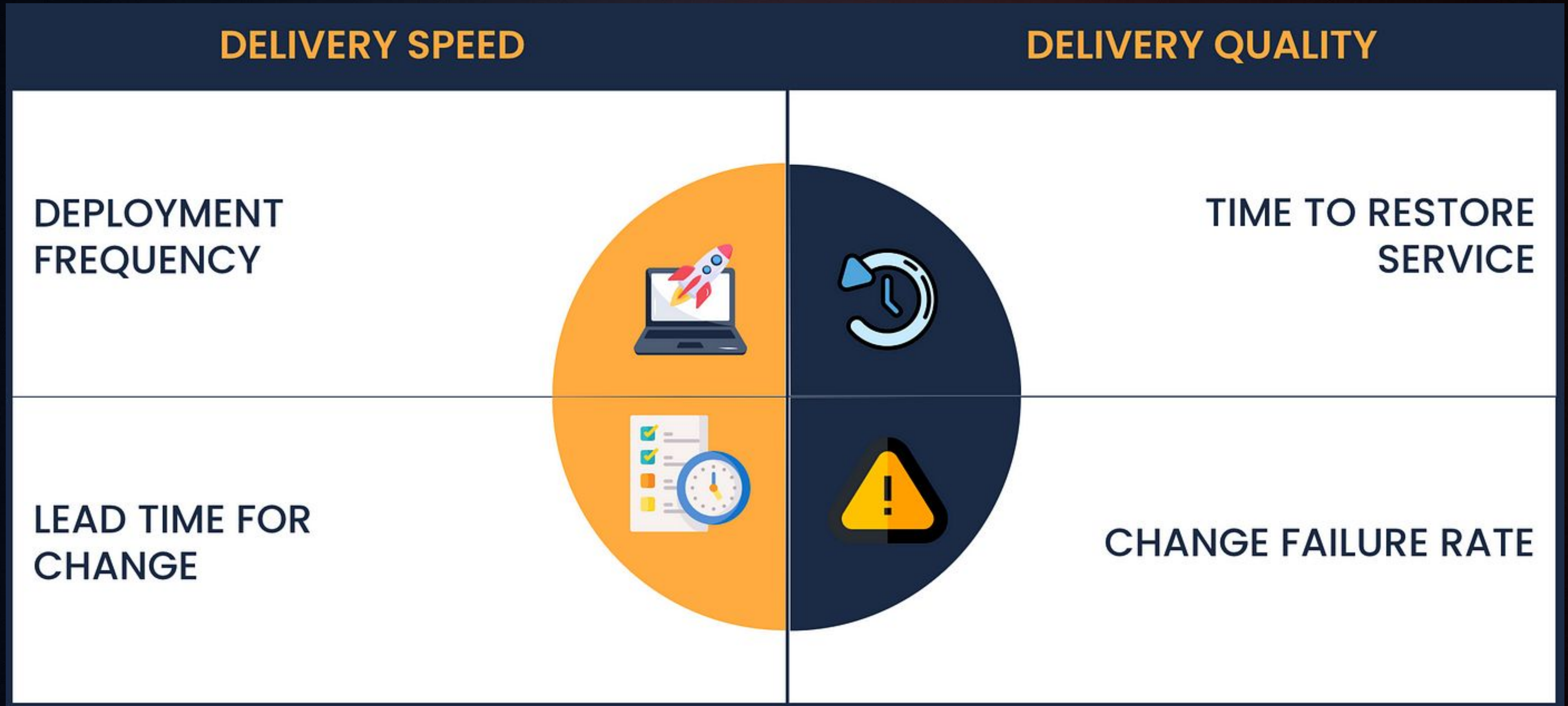
[CLOUDWEB-16600](#)

Лэндинги:

[CLOUDWEB-15564](#)

А точно ли стало хорошо?

Что есть DORA?



DORA - Lead Time For Changes

1200 часов => 30 часов



Процесс

- Разработчик сам катит свою задачу
- Если нужно разгрести большую очередь – это задача для ~~суперфронтендера~~ дежурного

**Пункт назначения —
ваш проект** 

Ваш проект

- каждый проект уникален
- ресерчим а потом решаем
- реалистично оцениваем свои силы

Baseline

- стабильные автотесты
- стабильный CI
- корректный git flow
- >1 тестового стенда (параллелизация)

Как сделать ещё лучше

- меньше ручного тестирования – меньше блокировок
- вырезать ручное вообще ✂️
- стабильность, стабильность и ещё раз стабильность

Вы не проходите

- проекты с внешними ограничениями
(ревью в сторах, etc)
- проекты с очень высокими требованиями к качеству
(банк, биржа)
- вещи которые нельзя откатить
(интегрируемое ПО...)



Запрыгиваем на релизный поезд. Как мы ускоряли путь задач фронтенда на production

Юрий Голубев

ведущий frontend разработчик в VK

Frontend

FrontOps

Tools



Предыдущий
доклад про деплой