

Мобильный CI. Как выбрать и переехать



Mobile CI

Прокачиваем mobile CI

Teamcity IOS

Agent 1

Build PR

Build master

Build release

Support scripts

Teamcity Android

Agent 1

Build PR

Build master

Build release

Support scripts



По одному агенту на каждую OS оказалось **мало**, разработчики не дожидались долгого ответа от CI и мержили пулы в мастер.



Прокачиваем mobile CI

Teamcity IOS

Agent 1

Agent 2

Build PR

Build master

Build release

Support scripts

Teamcity Android

Agent 1

Agent 2

Build PR

Build master

Build release

Support scripts

Прокачиваем mobile CI

Teamcity IOS

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Teamcity Android

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts



На TeamCity IOS скапливалась большая очередь,
а максимальное количество агентов в бесплатной версии
уже **достигнуто**.

Прокачиваем mobile CI

Teamcity IOS

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Teamcity IOS 2

Agent 4

Agent 5

Teamcity Android

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Как использовать 2 сервера TeamCity для одних сборок?

- Никак
- Можно разделить сборки и одни запускать на одном сервере, другие на другом
- Так мы разделили сборку Build PR на сборку проекта и UI тесты
- Сборка шла на первом TeamCity, а UI-тесты на втором

... всё работает, круто всё работает, круто всё работает, круто

Прокачиваем mobile CI

Teamcity IOS Build

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Teamcity IOS UI-test

Agent 4

Agent 5

UI-test PR

Teamcity Android

Agent 1

Agent 2



Agent 3

Build PR

Build master

Build release

Support scripts



CI администрировался разработчиками самостоятельно, из-за чего часто хаотично менялся пайплайн, что приводило либо к случайному отключению проверок, либо к нестабильным сборкам.

Решение — забрать доступы у разработчиков и **передать** нашим **DevOps инженерам**, пусть делают красиво.

Прокачиваем mobile CI

Teamcity IOS Build

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Teamcity IOS UI-test

Agent 4

Agent 5

UI-test PR

Jenkins Android

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Переезд Android **длился очень долго** из-за отсутствия времени у наших DevOps инженеров и понимания особенностей сборок Android проектов, которых DevOps не знали.

Появилась идея, чтобы мобильным CI занимался человек, который в курсе специфик мобильных платформ и умеет в инфраструктуру. Компании **нужен** был **Mobile DevOps**, которого она заслуживает.



На этом моменте я пришел в Авиасейлс. Первое о чем меня просили мобильные команды — **навести порядок в СІ** и стабилизировать его.

Нерациональное использование агентов.

Очередь на IOS сборки была всегда больше, чем на полные прогоны UI тестов.

Двойная сборка тестов.

Из-за отдельных конфигураций, невозможно было переиспользовать сборку для IOS UI и Unit тестов.

XX2

Двойная работа при администрировании CI серверов. Приходилось вносить изменения в обоих TeamCity.

XX2

**Вся проблема в 2 серверах CI
для одной платформы**

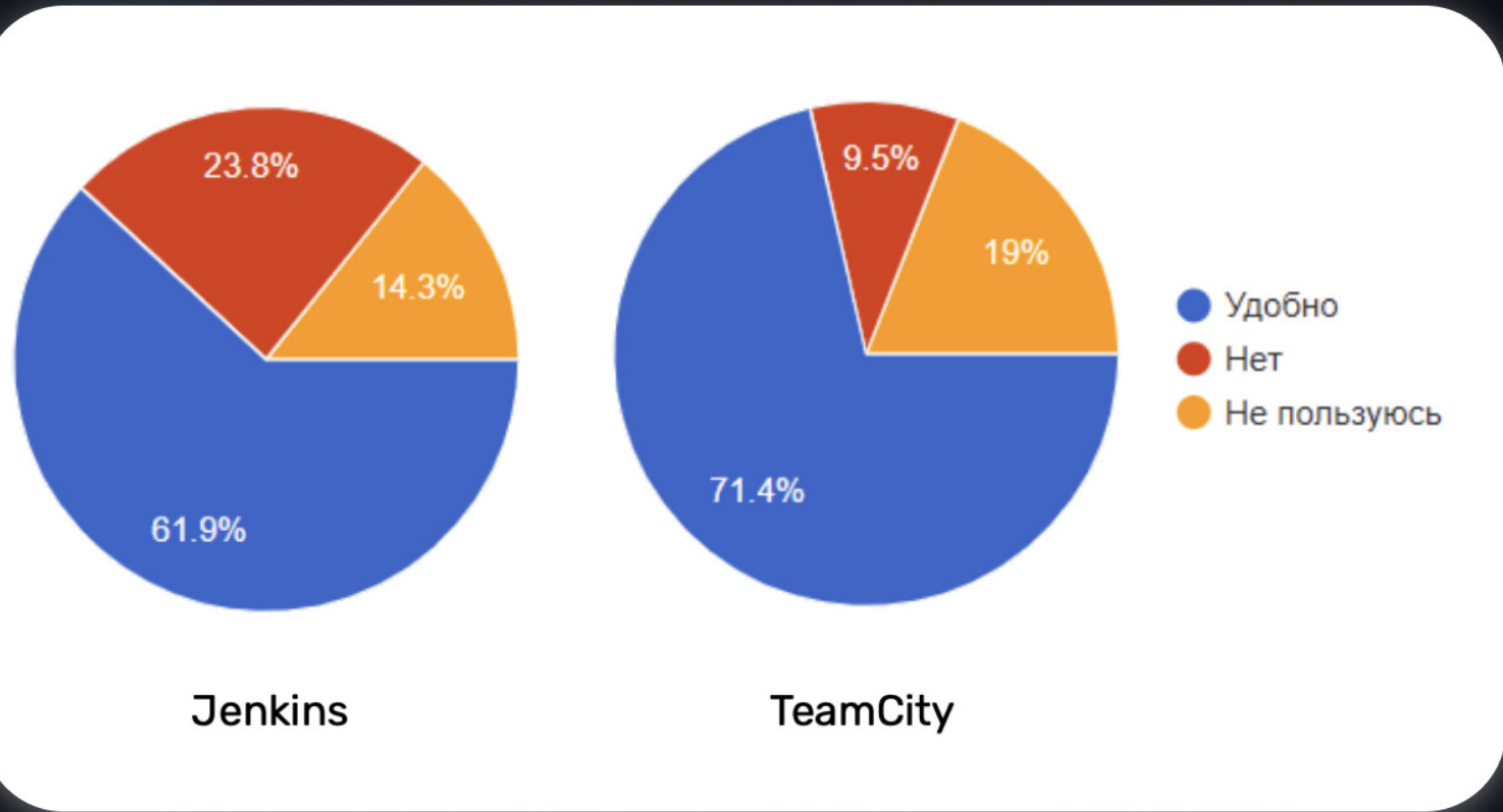


**Было принято решение
объединить все CI**



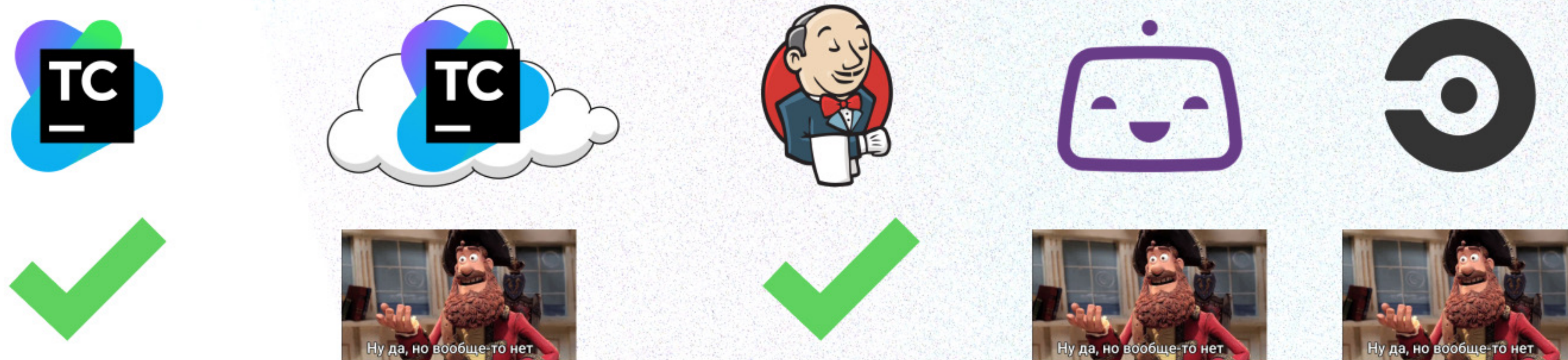
Чтобы не рассматривать зря текущие варианты,
надо **спросить разработчиков** насколько они приемлемы.

Удобно ли пользоваться?



Спросили у сообщества, показав наш список требований к CI.
Порекомендовали **self hosted** решения, потому что они
намного дешевле облачных решений.

Возможность прогонять UI тесты на нескольких девайсах параллельно



сё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает

IOS

Ресурс	CPU	RAM	Цена
Agent 1	8	8	20\$
Agent 2	6	16	20\$
Agent 3	6	16	20\$
Agent 4	6	16	20\$
Agent 5	6	8	20\$
Agent 6	6	16	20\$
Agent 7	2	8	20\$
Teamcity host	8	32	80\$
Всего			220\$

сё работает, круто

о всё работает, круто

Стоимость нашей инфраструктуры

Включает все используемые ресурсы CI мобильных команд

620\$/мес

сё работает, круто всё работает, круто всё работает, круто вс



Цель — рассчитать сколько стоят
такие же ресурсы на разных CI



**Стоимость инфраструктуры
на TeamCity**

1120\$/мес

сё работает, круто всё работает, круто всё работает, круто вс



**Стоимость инфраструктуры
на TeamCity Cloud**

1220\$/мес

сё работает, круто всё работает, круто всё работает, круто всё



**Стоимость инфраструктуры
на Jenkins**

540\$/мес

сё работает, круто всё работает, круто всё работает, круто всё



**Стоимость инфраструктуры
на Bitrise**

2500\$/мес

сё работает, круто всё работает, круто всё работает, круто всё



**Стоимость инфраструктуры
на Circle CI**

1000\$/мес

сё работает, круто всё работает, круто всё работает, круто всё



Так что в итоге с деньгами?



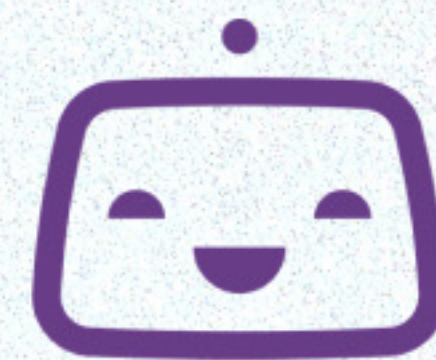
1120\$



1220\$



540\$



2500\$



1000\$

сё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает, круто

Android уже был **на Jenkins**, поэтому работы по переезду в два раза меньше, чем могло бы быть.

IOS CI

Teamcity Build

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Teamcity Test

Agent 4

Agent 5

UI-test PR

UI-test full

Agent 6

Agent 7

Настройка IOS под

- Обновление OS
- Обновление Xcode
- Настройка сертификатов
- Установка симуляторов

сё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает, круто всё работает

Настраиваем Jenkins ноды

Teamcity Build

Agent 1

Agent 2

Agent 3

Build PR

Build master

Build release

Support scripts

Teamcity Test

Agent 4

Agent 5

UI-test PR

UI-test full

Jenkins

Node 6

Node 7

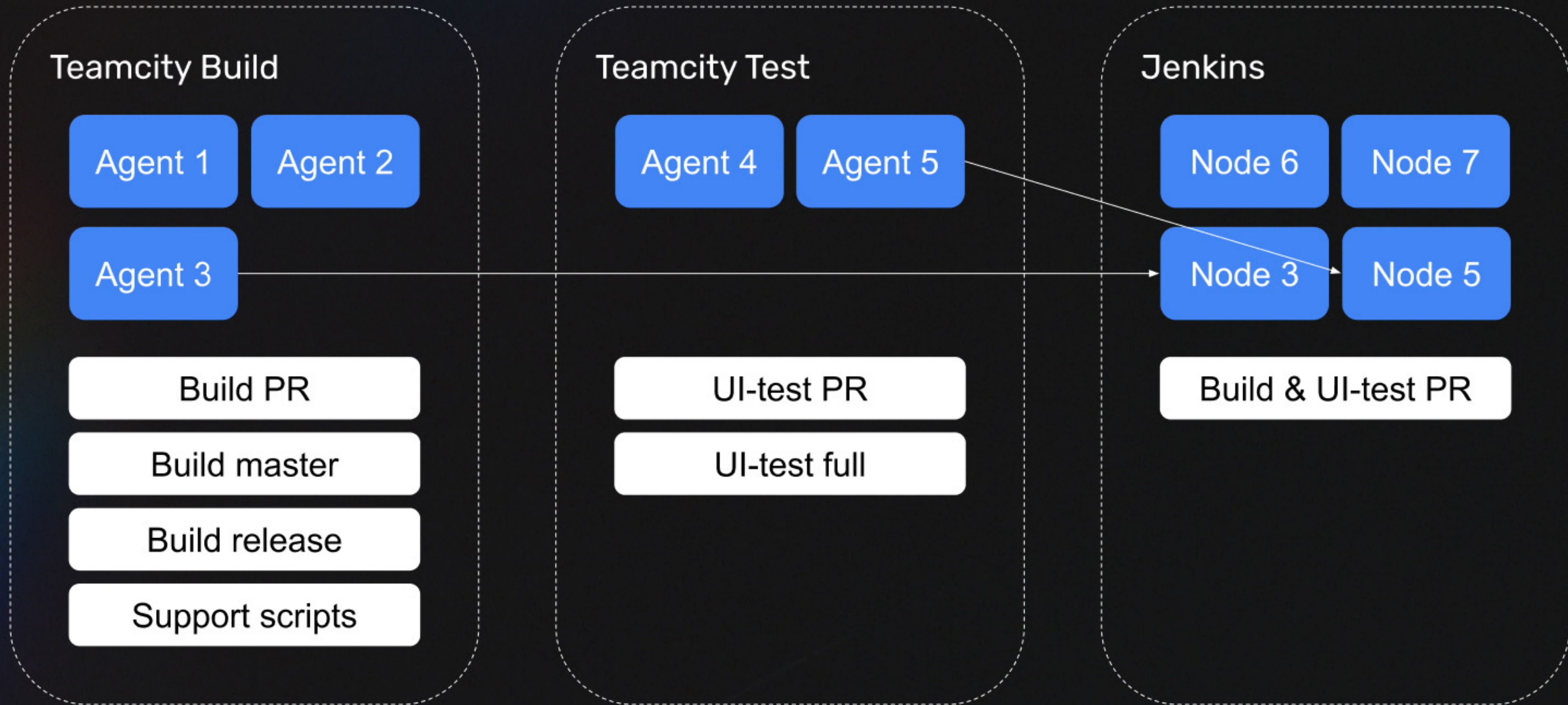
Сначала перенесем самые **важные и частые** по запуску пайплайны. Это поможет быстрее отказаться от выбранного CI, если что-то не будет работать.

Объединение Build PR и UI-test PR

- Общий checkout проекта
- Сборка будет занимать одну ноду
- Возможность не прогонять UI-тесты, если упал линтер
- Возможность параллельно прогонять Unit и UI-тесты на одной ноде
- Сборка тестов выполняется один раз для Unit и UI-тестов

... всё работает, круто всё работает, круто

Настраиваем новые ноды на Jenkins



PR пайплайн на Jenkins

Teamcity Build

Agent 1

Agent 2

Build master

Build release

Support scripts

Teamcity Test

Agent 4

UI-test full

Jenkins

Node 6

Node 7

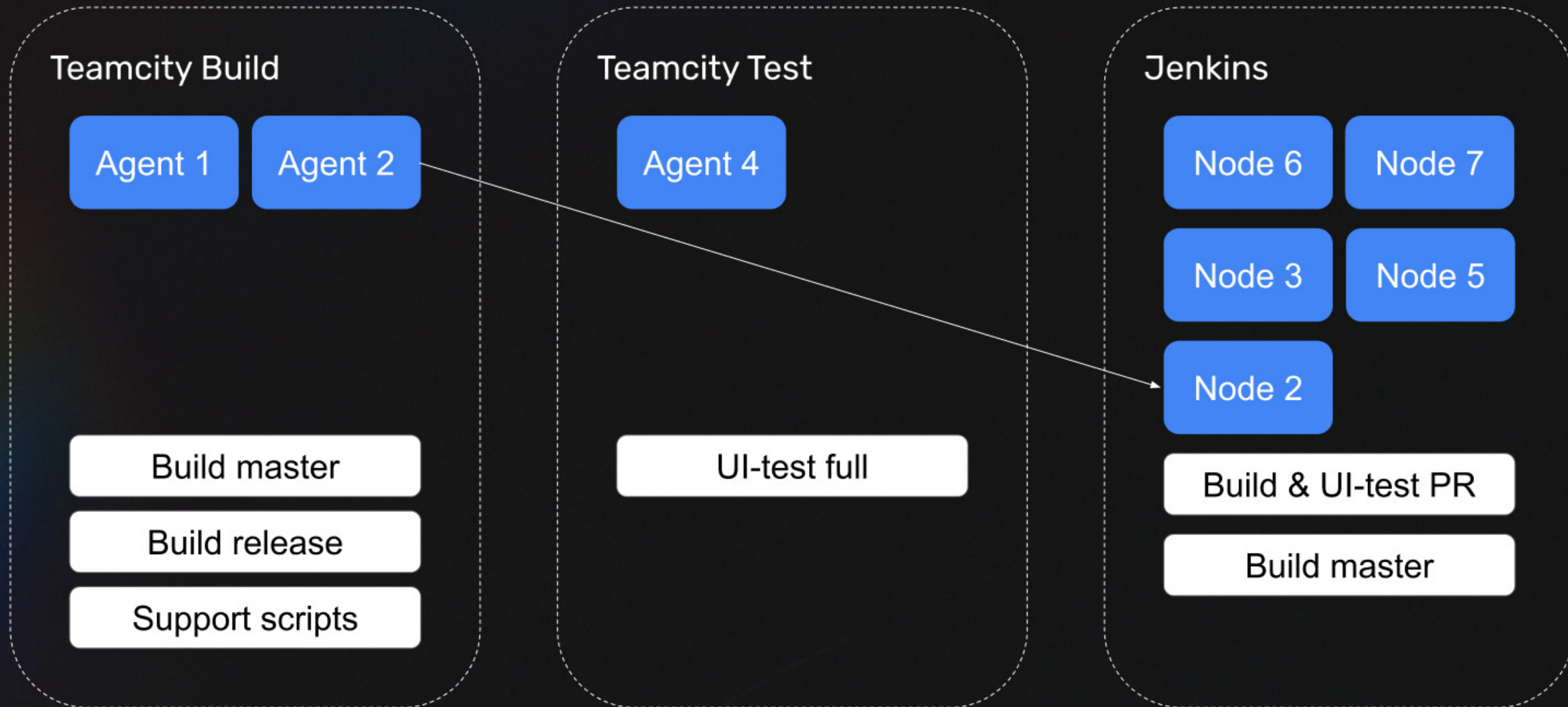
Node 3

Node 5

Build & UI-test PR

Несмотря на стабильную работу пайплайнов на Jenkins, мы не удаляем конфигурации с TeamCity и не удаляем TeamCity агенты с машин, потому что если **что-то пойдет не так**, можно будет быстро **откатиться**.

Настраиваем мастер пайплайн на Jenkins



Мастер пайплайн на Jenkins

Teamcity Build

Agent 1

Build release

Support scripts

Teamcity Test

Agent 4

UI-test full

Jenkins

Node 6

Node 7

Node 3

Node 5

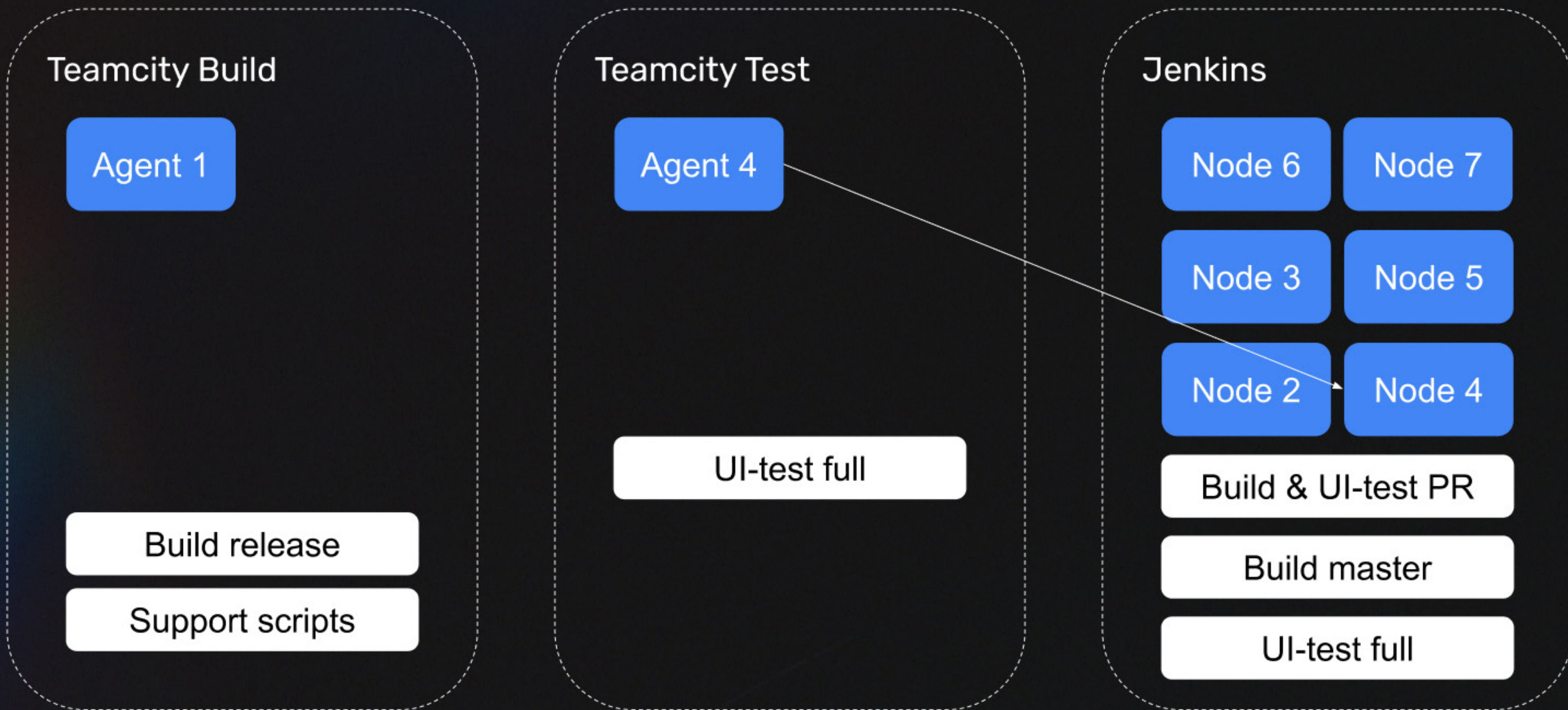
Node 2

Build & UI-test PR

Build master

Есть возможность перенести пайплайн полного прогона UI-тестов и **избавиться** от одного **TeamCity** сервера.

Настраиваем UI-test full пайплайн на Jenkins



UI-test full пайплайн на Jenkins

Teamcity Build

Agent 1

Build release

Support scripts

Jenkins

Node 6

Node 7

Node 3

Node 5

Node 2

Node 4

Build & UI-test PR

Build master

UI-test full

Настраиваем Build release пайплайна на Jenkins

Teamcity Build

Agent 1

Build release

Support scripts

Jenkins

Node 6

Node 7

Node 3

Node 5

Node 2

Node 4

Build & UI-test PR

Build master

UI-test full

Build release

Build release пайплайн на Jenkins

Teamcity Build

Agent 1

Support scripts

Jenkins

Node 6

Node 7

Node 3

Node 5

Node 2

Node 4

Build & UI-test PR

Build master

UI-test full

Build release

Настраиваем Support scripts на Jenkins

Teamcity Build

Agent 1

Support scripts

Jenkins

Node 6

Node 7

Node 3

Node 5

Node 2

Node 4

Build & UI-test PR

Support scripts

Build master

UI-test full

Build release

Support scripts на Jenkins

Teamcity Build

Agent 1

Jenkins

Node 6

Node 7

Node 3

Node 5

Node 2

Node 4

Build & UI-test PR

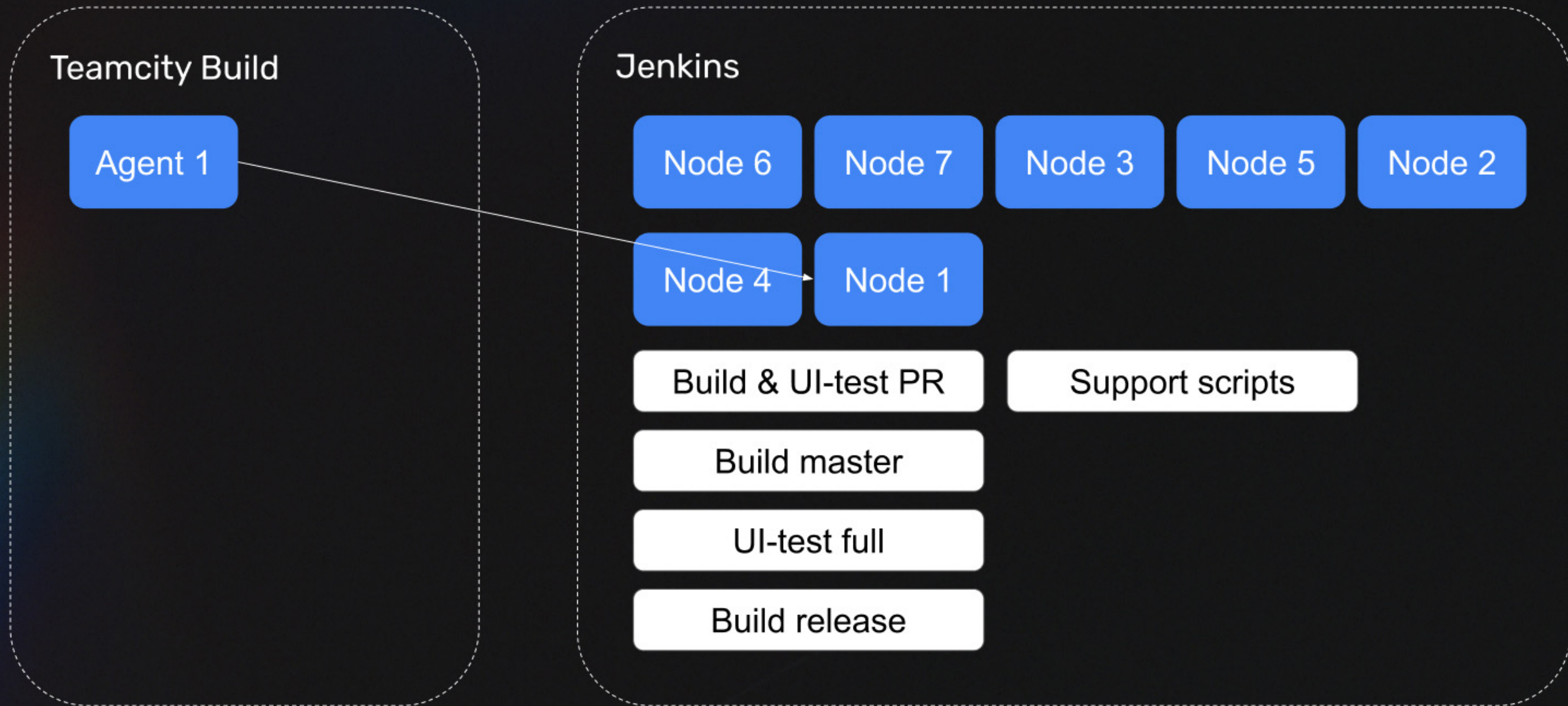
Support scripts

Build master

UI-test full

Build release

Подключаем последнюю ноду на Jenkins



Переезд закончен!

Jenkins

Node 6

Node 7

Node 3

Node 5

Node 2

Node 4

Node 1

Build & UI-test PR

Build master

UI-test full

Build release

Support scripts

О чем помнить при переезде?

- Сначала настроить пайплайн
- Проверить его в тихие часы, чтобы не блочить работу
- Настроить ноду, оставляя ее на старом CI
- Проверить все пайплайны на новой ноде в тихие часы
- Только после этого подключить полноценно в новый CI
- Не гасить старый CI сразу после переезда

Сотает, круто всё работает, круто

Что мы выиграли?

- Администрировать надо только один CI
- Сборки IOS стали оптимальнее за счет переиспользования сборок и возможности распараллеливать работу
- Можем «бесплатно» наращивать мощности
- Получили бесплатную экосистему плагинов
- Наш код и секретные данные не передается стороннему CI

Создает, круто всё работает, круто всё работает, круто

Спасибо!

Севастьян Жуков,

Mobile DevOps

tg: **@SiberianDroid**

jobs.aviasales.ru

