

How technical excellence helps in LeSS adoption



Anton Bevzuk
Dodo Pizza
Chief Agile Officer

The plan

- Why engineering practices?
- Deep dive into
 - Pair Programming
 - Test Automation
 - Continuous Integration
- Q&A

What is Dodo Pizza?

11 countries

372 pizzerias

200 orders/min

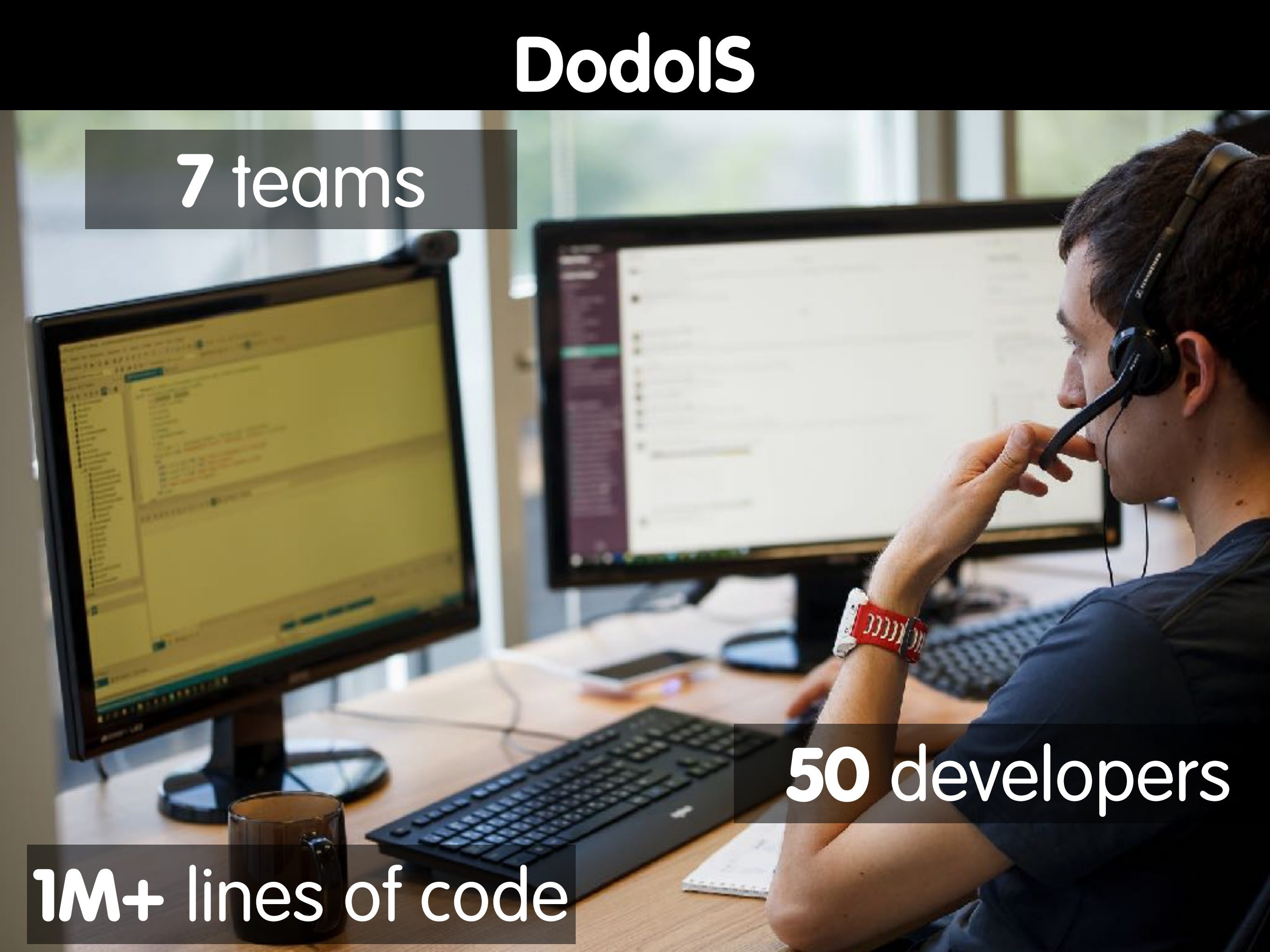


DodoIS

7 teams

50 developers

1M+ lines of code

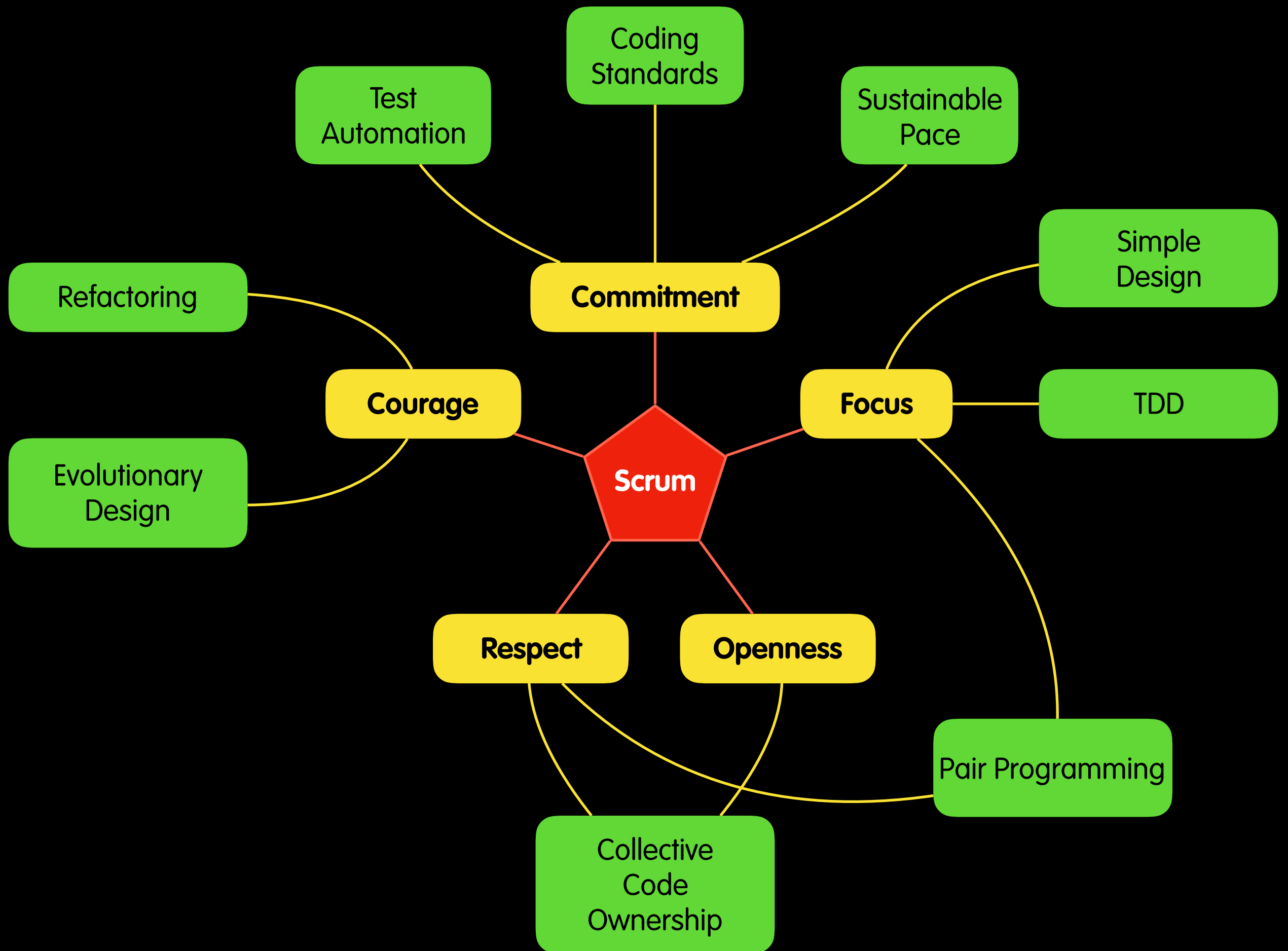


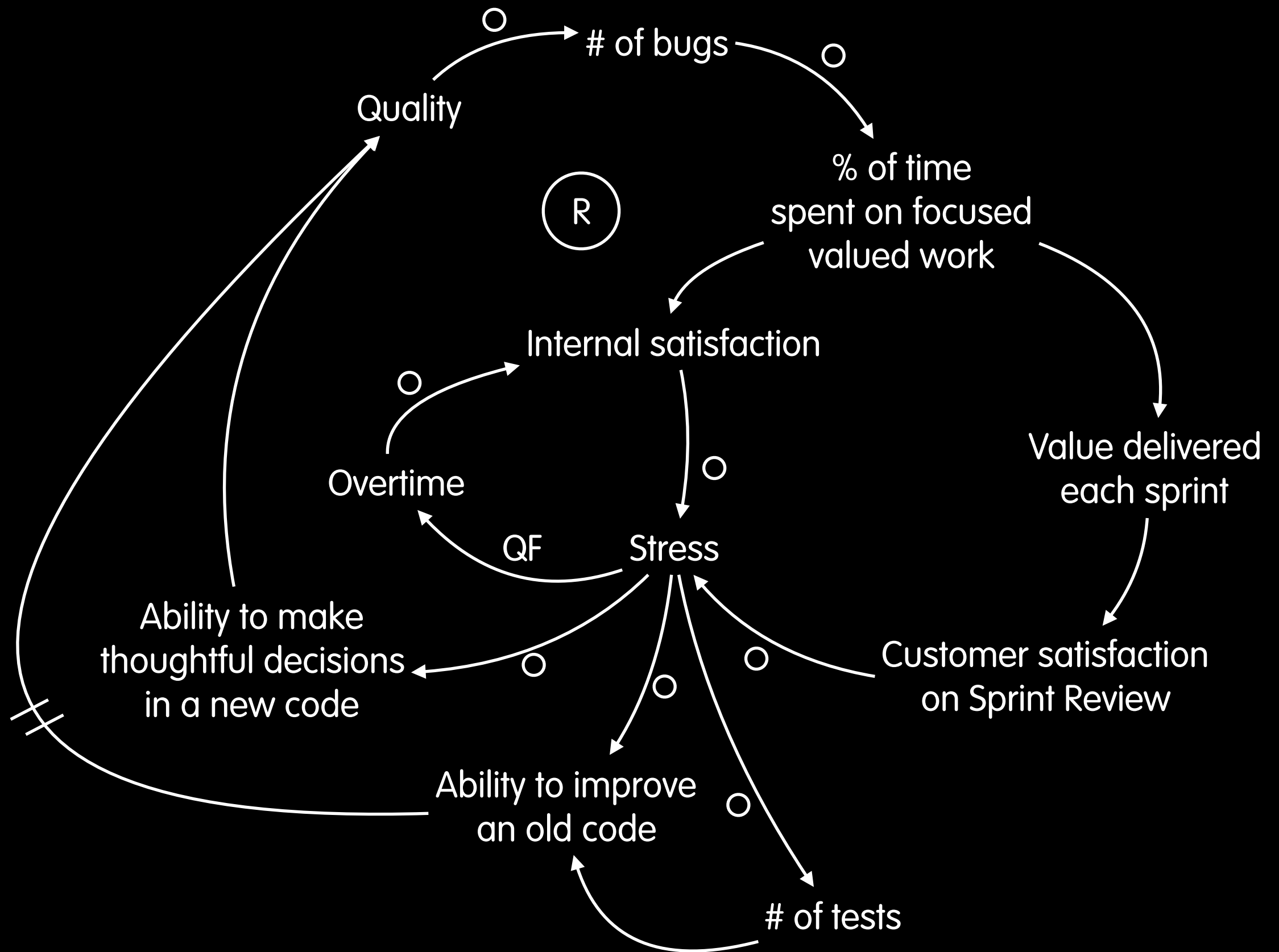
Scrum as well as LeSS
does not work

without
engineering practices

Continuous attention
to technical excellence
and good design
enhances agility

agilemanifesto.org





Test Driven Development by Example by Kent Beck, 2000

124

TEST-DRIVEN DEVELOPMENT PATTERNS

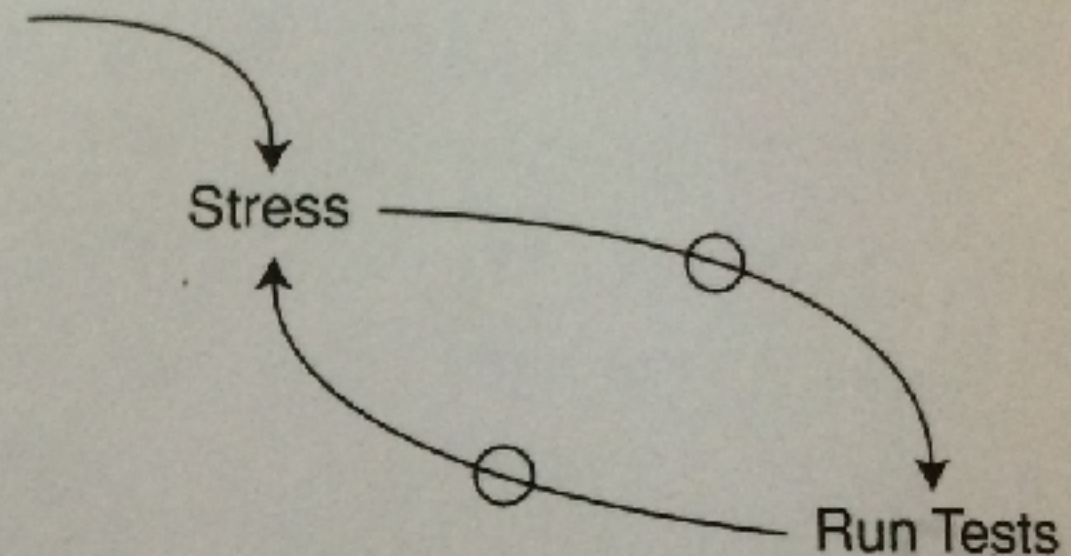


Figure 25.1 *The “no time for testing” death spiral*

This is a positive feedback loop. The more stress you feel, the less you will do. The less testing you do, the more errors you will make. The more errors you make, the more stress you feel. Rinse and repeat.

How do you get out of such a loop? Either introduce a new element or change one of the elements, or change the arrows. In this case we'll replace

The most simple
engineering practices
are the most powerful

Pair Programming

Just a few rules

- 2 roles: Driver and Observer
- Switch roles every 10-15 mins
- Rotate pairs at least every day

Lightweight

Simple to understand

Difficult to master

Single team open space







Pair Programming: HW

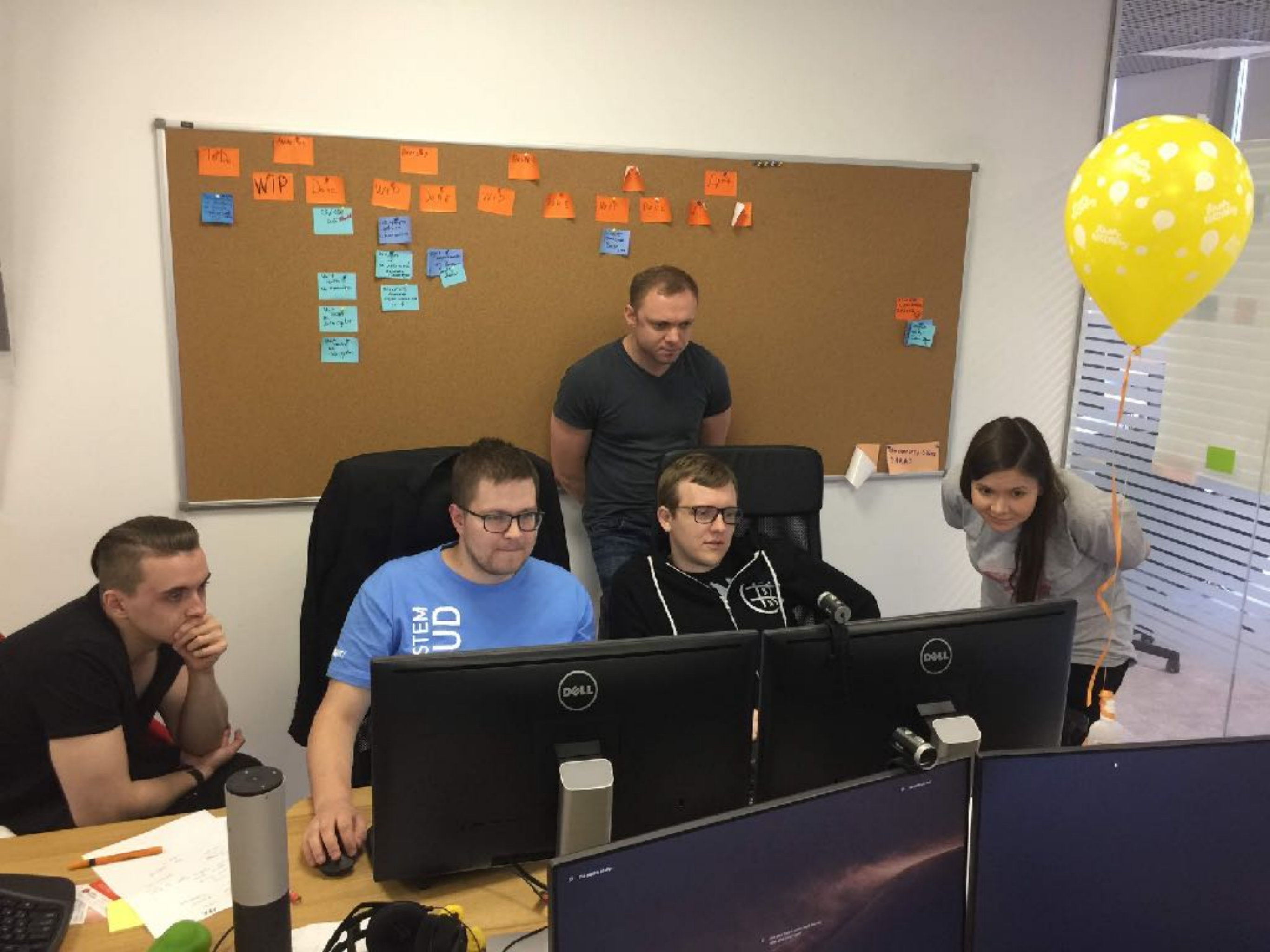
Do's	Dont's
Open space for each team	Open space for multiple teams Cubicles
Flat wide tables	Corner desk Narrow table
Powerful pair stations	Crappy hardware
2 wide 24"-30" monitors Duplicate screen	Single crappy monitor
Consistent settings, hotkeys, etc.	Personalize

Pair Programming: People

Do's	Dont's
Demonstrate	Push
DevSchool	
Mentor	
Let them choose	

Pair Programming: Pairs

Do's	Dont's
Experienced + Newbie	Newbie + Newbie
Experienced + Experienced	



Pair Programming: Roles

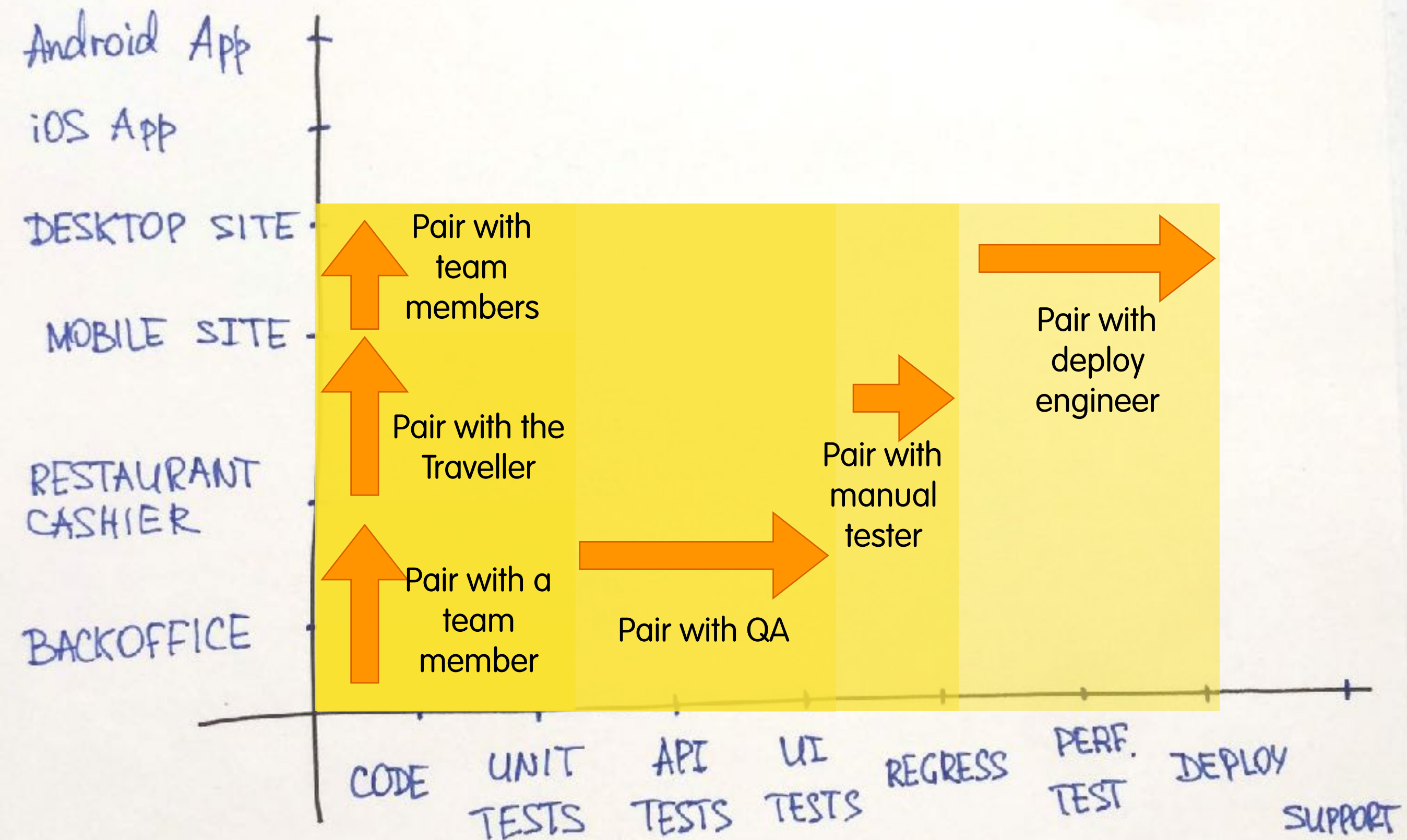
Do's	Dont's
Developer + Developer	Limit yourself
Developer + QA	
Developer + Analyst	
Developer + Lawyer	
Developer + ...	

Pair Programming is not just about programming ...

- Code structure
- Coding standards
- New technologies
- Business domain
- Decomposition
- Focus
- Internal processes
- Other practices (TDD etc.)
- Deploy
- DoD
- Tips & tricks
- Help each other

Pair Programming builds the Team

Pair Programming in LeSS



Test Automation



Unit Test example 1

```
[Test]
public async Task SimpleProductsOnNextStage_ShouldOrderByChangeDate()
{
    var firstOrder = Given.Order()
        .WithSandwich(stage: ProductionStage.OnHotAssemblyTable)
        .ChangedAt(1.October(2017).At(13, 40)).Please;
    var secondOrder = Given.Order()
        .WithSnack()
        .CreatedAt(1.October(2017).At(13, 35)).Please;

    var result = await TrackerService.GetColdTableAsync(RequestContext, CURRENT_STATE_COUNT,

result.ProductionItems.Select(x => x.UUID).Should()
    .HaveCount(2)
    .And.StartWith(secondOrder.Line(1).UUID)
    .And.EndWith(firstOrder.Line(1).UUID);
}
```


Unit Test example 2

```
[Test]
public void OrderWillBeOnDelivery()
{
    var currentDate = 18.April(2018).At(10,00);
    var order = Create.Order(unitId, version: 2, orderType: OrderType.Pickup).Please();

    var result = order.CarryOut(
        new CarryOutOrderCommand(unitId, order.Id, pizzeriaShiftId, cashboxSessionId, cu

    result.Should().Be(CarryOutResultCode.Ok);
    order.State.Should().Be(OrderState.OnDelivery);
    order.Version.Should().Be(3);
}
```

Unit BDD Test example

```
[TestFixture]
[FeatureDescription("when extend courier shift")]
public class WhenExtendCourierShift : FeatureFixture
{
    [Scenario]
    public void CourierReturnsToQueue()
    {
        var shiftOpened = 11.July(2018).At(8,00);
        var shiftClosed = 11.July(2018).At(10,00);
        var currentDate = 11.July(2018).At(11,00);
        Runner
            .WithContext<CourierContext>()
            .RunScenario(
                _ => _.given_courier_NAME("Andrey"),
                _ => _.given_COURIER_is_on_shift_from_FROM("Andrey", shiftOpened),
                _ => _.given_COURIER_shift_is_closed_at_AT("Andrey", shiftClosed),
                _ => _.when_you_extend_COURIER_shift_at_AT("Andrey", currentDate),
                _ => _.then_extending_is_Ok(),
                _ => _.then_COURIER_is_STATUS("Andrey", ShiftStatus.Queued)
            );
    }
}
```


API Test example

```
[Test]
[Category("ru")]
public void CanCookOrderWithPizzaAndSnacks()
{
    var order = Create.Order()
        .WithProduct(Product.Margarita())
        .WithProduct(Product.Dodster())
        .Please();

    var margaritaName = ResourceManager.GetString("pizzas.margarita");
    TrackerApi.Cook(order.ProductionItems.First(_ => _.Product.Name.Equals(margaritaName)));

    var dodsterName = ResourceManager.GetString("snacks.dodster");
    TrackerApi.Cook(order.ProductionItems.First(_ => _.Product.Name.Equals(dodsterName)));

    TrackerApi.Complete(order);

    var completedOrder = LegacyFacadeApi.GetOrderOnTheShelf(order);
    Assert.That(completedOrder.OrderState.AsEnum(), Is.EqualTo(OrderState.OnTheShelf), $"{ord
}
```


UI Test example

Функция: Глобальный сайт → Создание заказа на самовывоз

Контекст:

Пусть Я открыл глобальный сайт

Когда Перехожу на страницу конкретной пиццерии

Сценарий: Создание заказа на самовывоз с оплатой наличными

Пусть Добавляю пиццу "pizzas.margarita" с размером "pizzas.sizes.30" в корзину

Когда Я раскрываю корзину

И Нажимаю кнопку Корзина

Когда В корзине нажимаю кнопку Заказать

Тогда Вижу окно выбора типа доставки

Когда В окне доставки выбираю вкладку "site.deliveryType.carryout"

И В окне доставки выбираю текущую пиццерию

И В окне доставки нажимаю кнопку Выбрать

И Авторизуюсь со случайным телефоном, если появилось окно авторизации

Тогда Открывается окно контактов и оплаты

Когда В окне контактов и оплаты ввожу произвольное имя

И В окне контактов и оплаты произвольный телефон, если он не заполнен

И В окне контактов и оплаты выбираю способ оплаты "payment.type.cash"

И В окне контактов и оплаты нажимаю Оформить заказ

Тогда Открывается окно детализации заказа

Когда Запоминаю номер заказа

Когда Провожу заказ по трекингу

И Разлогиниваюсь

Когда Я вошел в бэкофис как Менеджер смены

И Выполнен переход в раздел Заказы

Тогда Я нашел заказ в статусе Доставка

Test Automation

Do's	Dont's
Automate critical path	Automate everything
Invest in containers infrastructure	
Unit tests API tests Component tests	E2E tests
DSL Unit tests readable by biz	Mocks of mocks of mocks

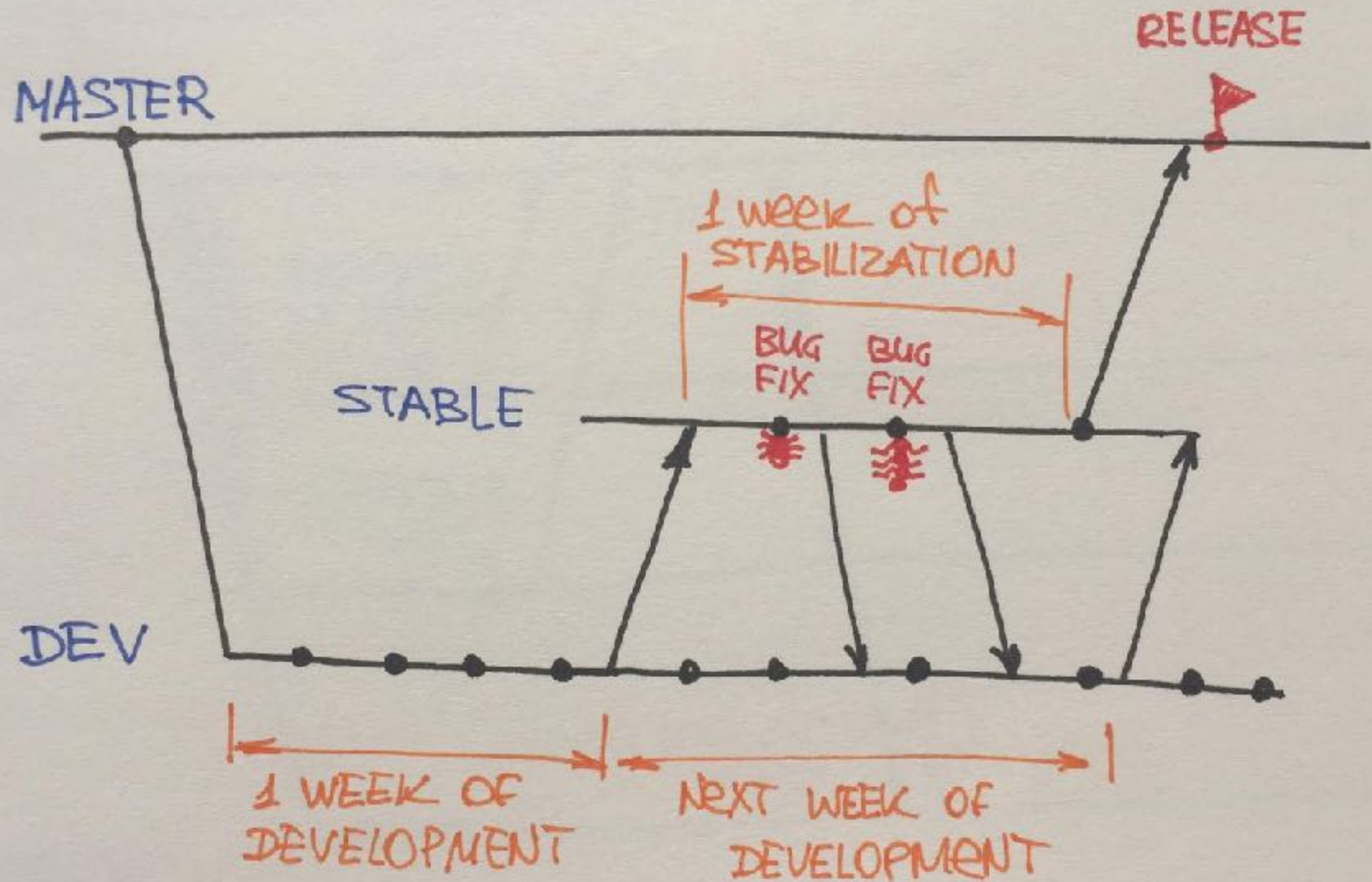
Test Automation

Do's	Dont's
Developers write tests	Separate team writes tests
QA team as a quick fix	Permanent QA team
Keep high test quality	
Delete useless tests	

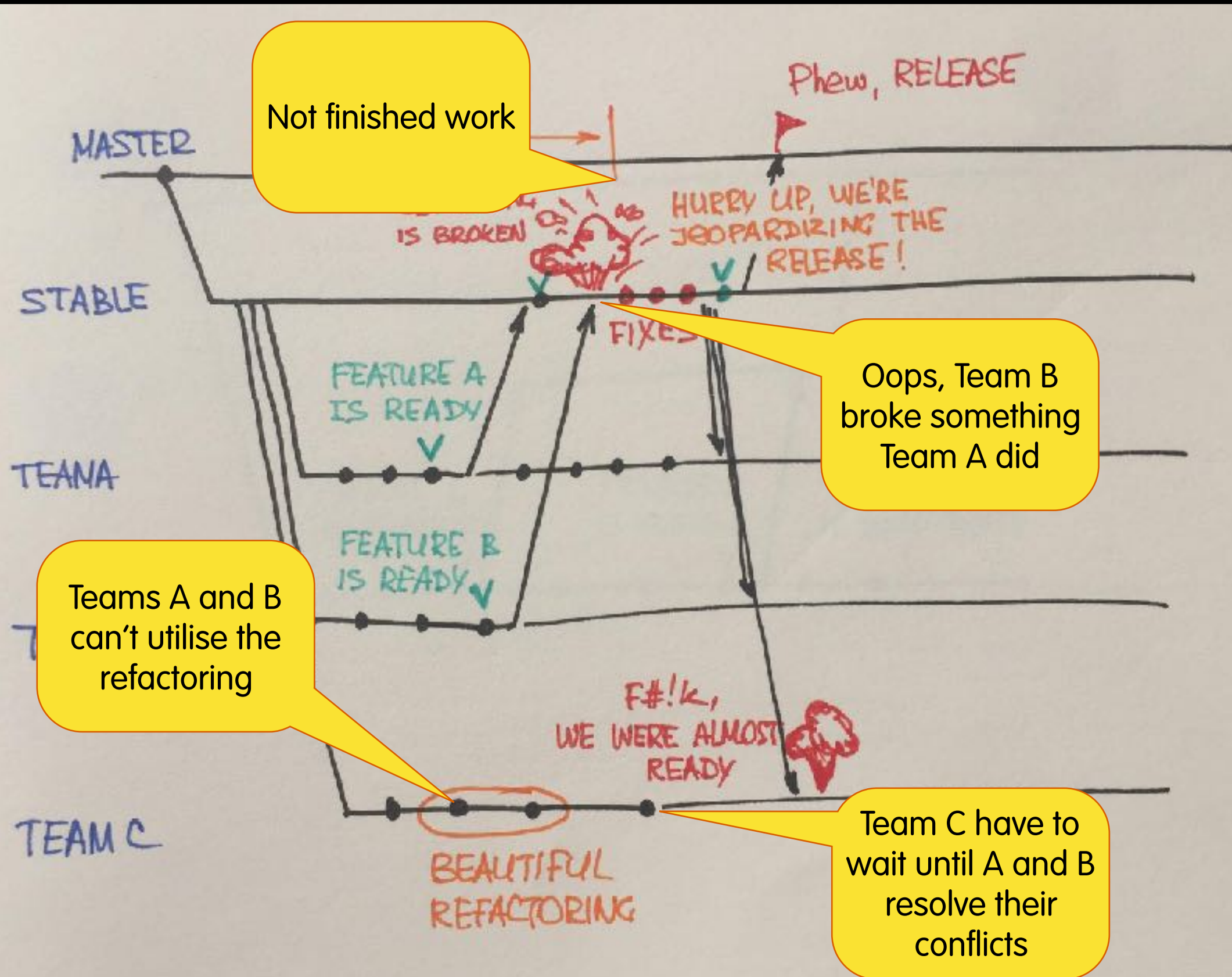
Test Automation
is a prerequisite
for ...

Continuous Integration

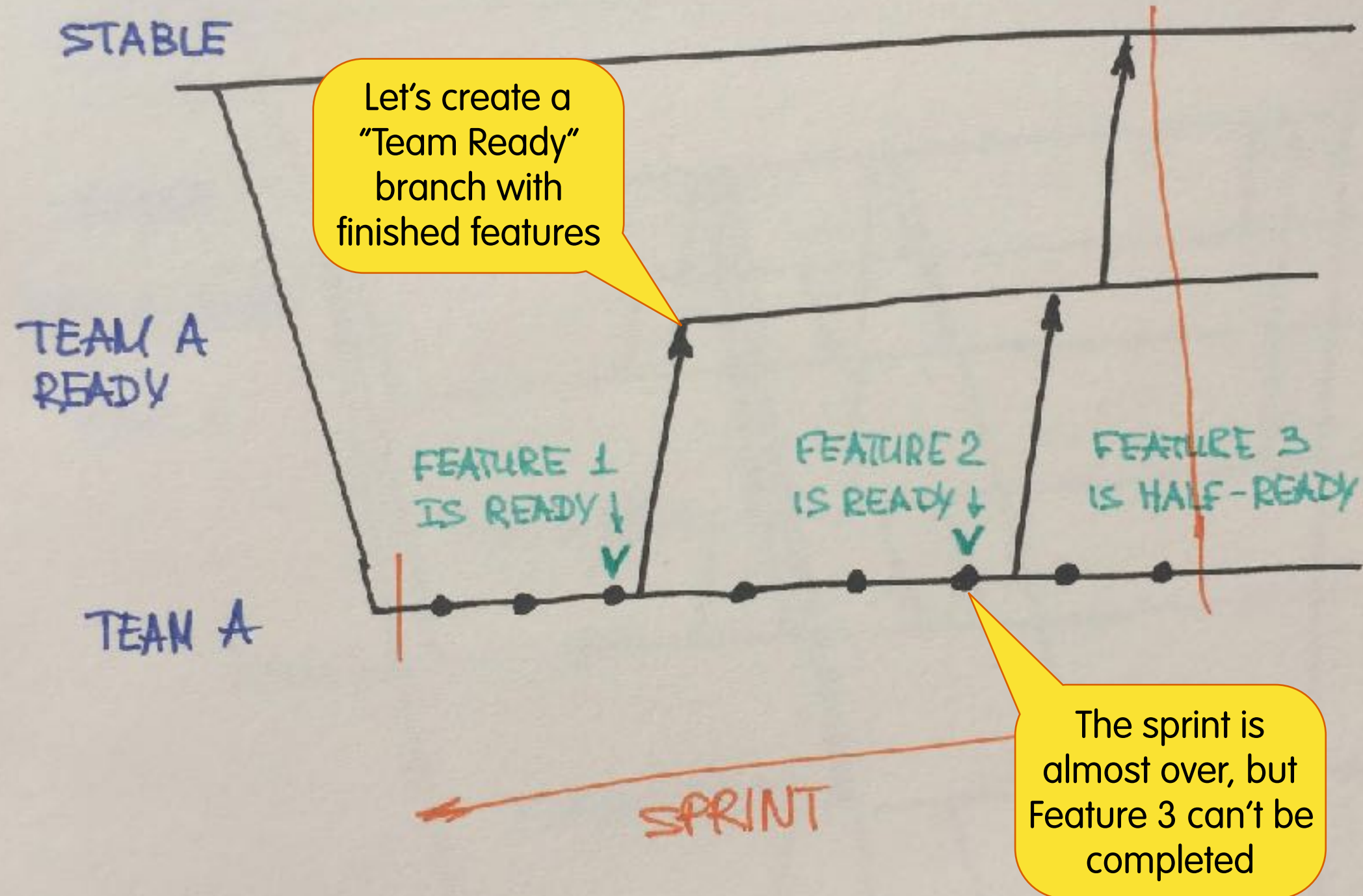
Configuration Management 2017



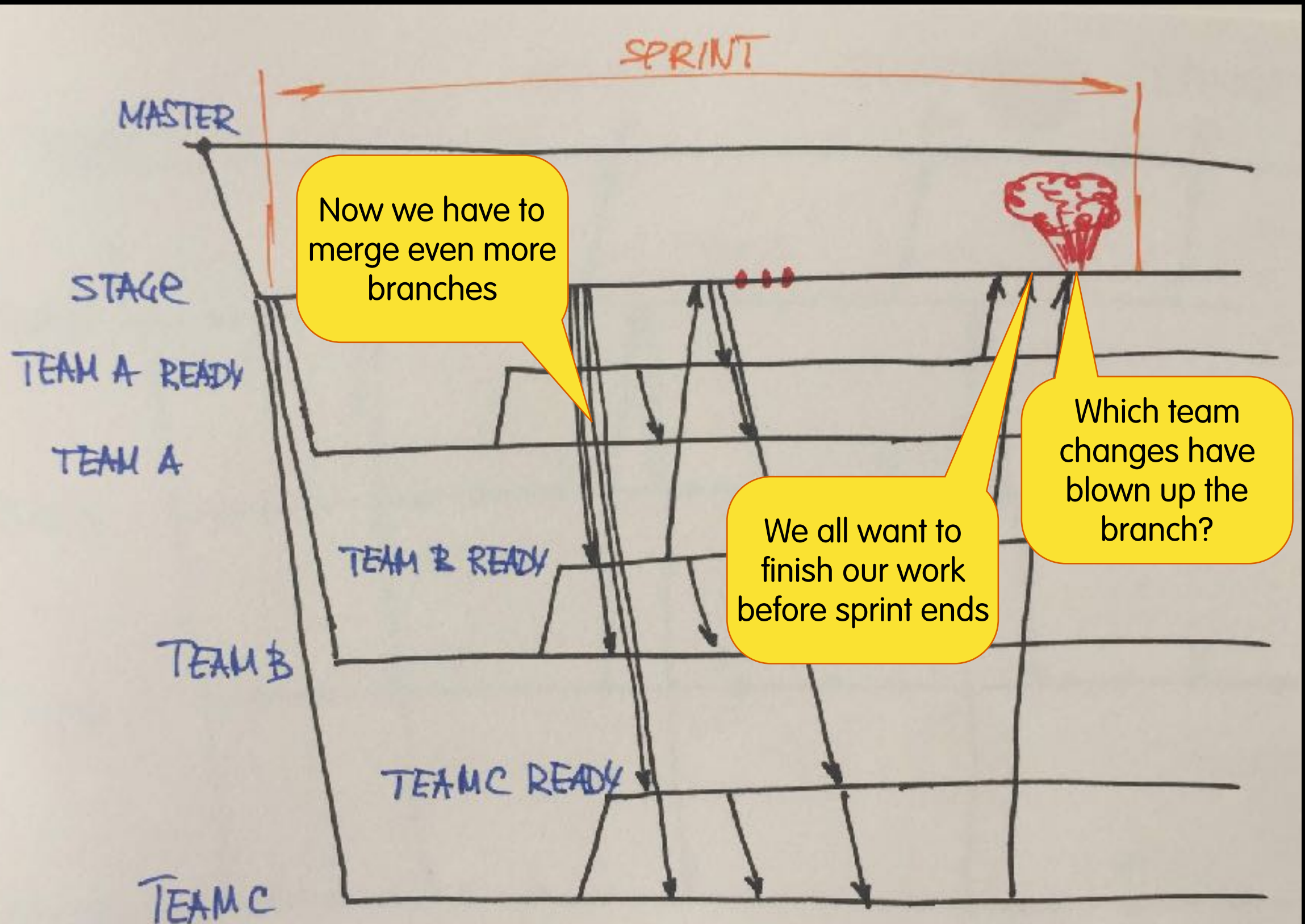
Teams' branches



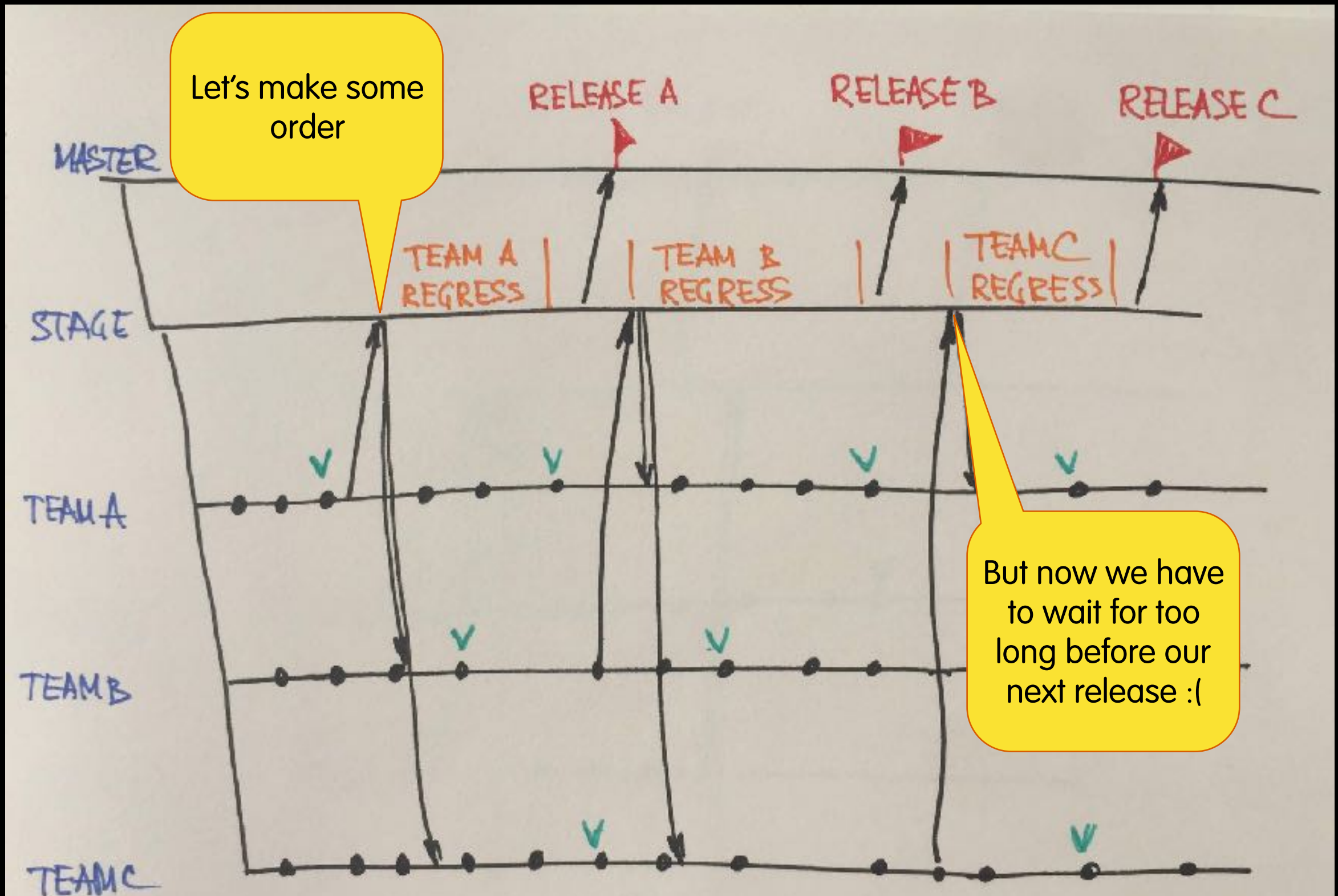
Unfinished work



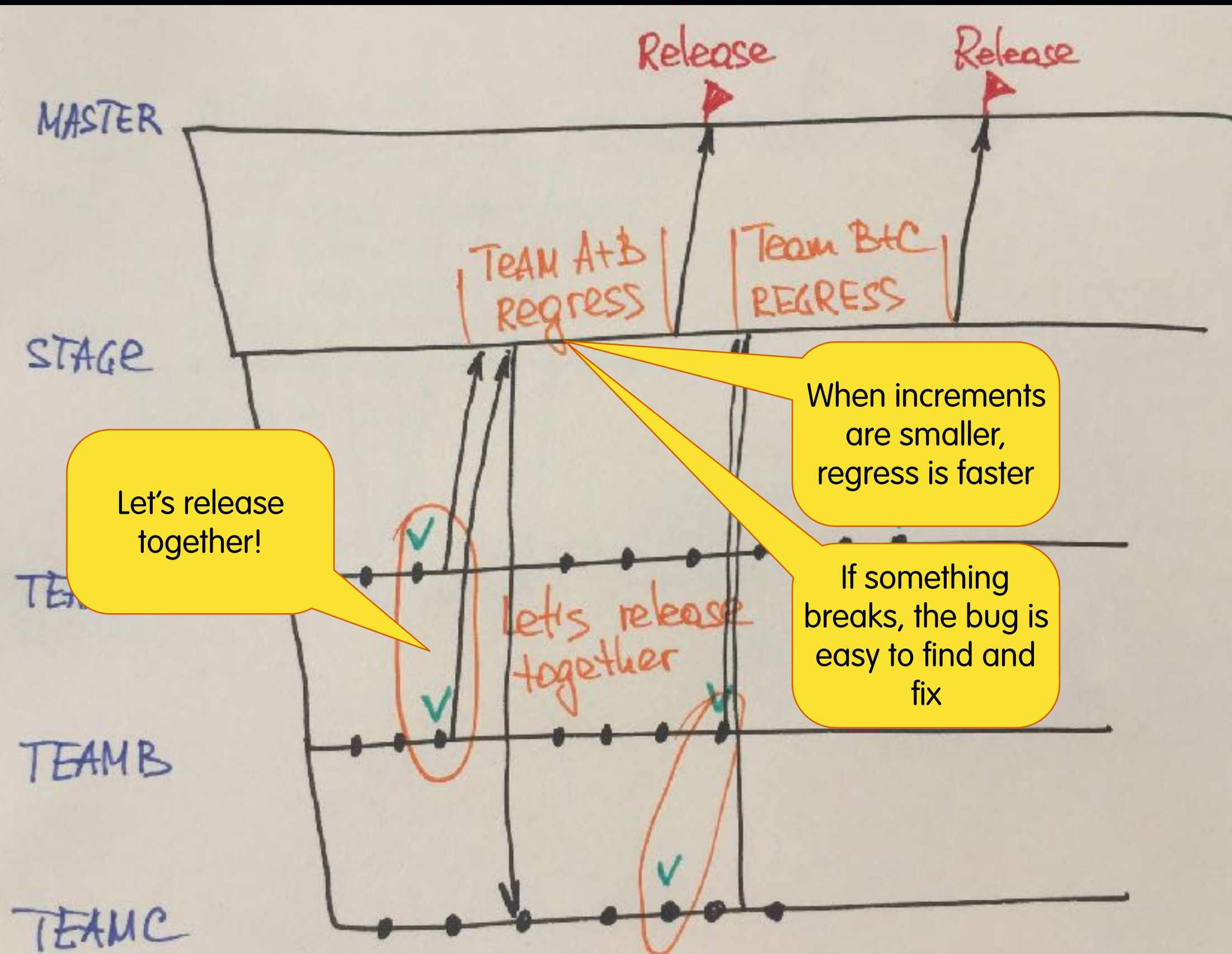
Complexity



Sequential team releases

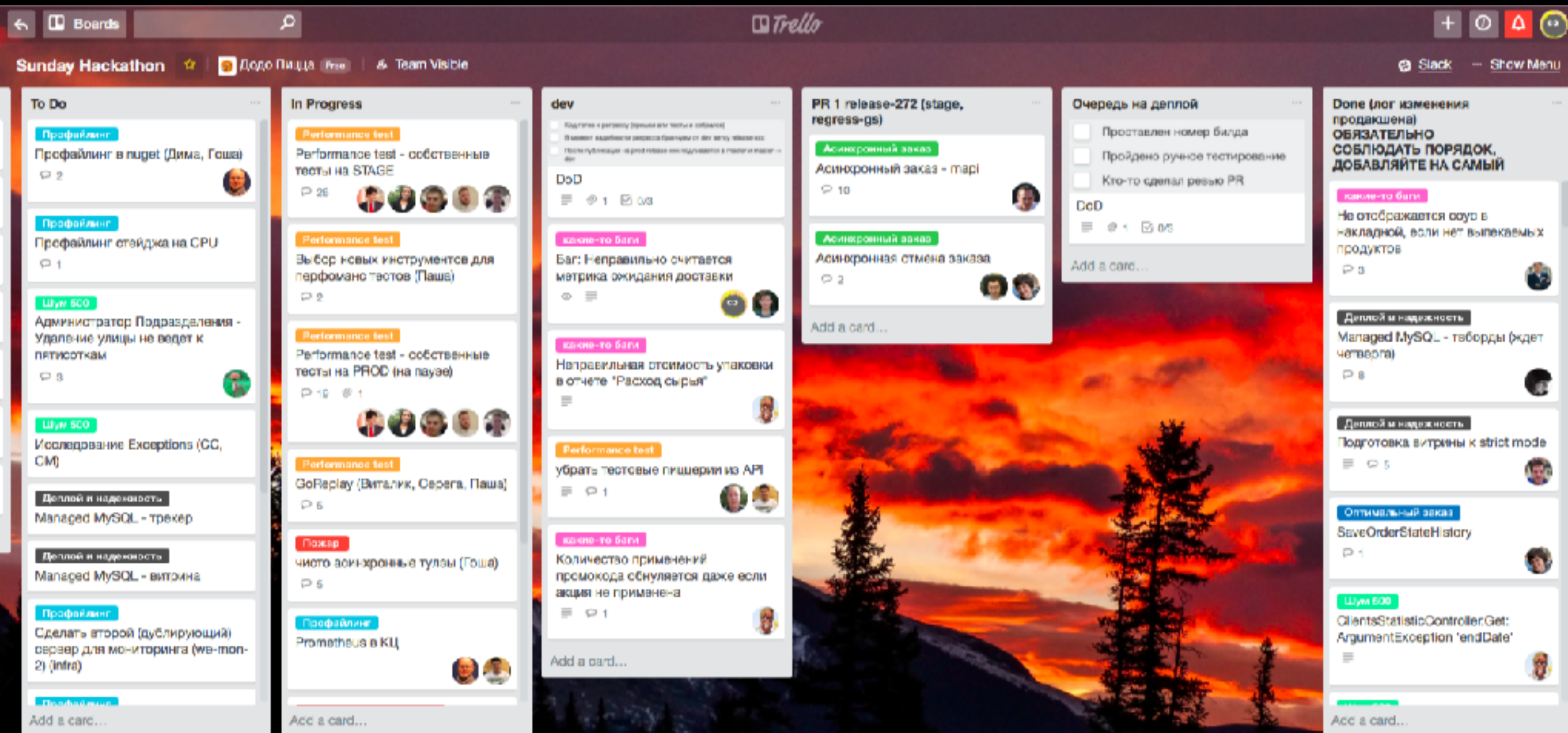


Joined teams releases



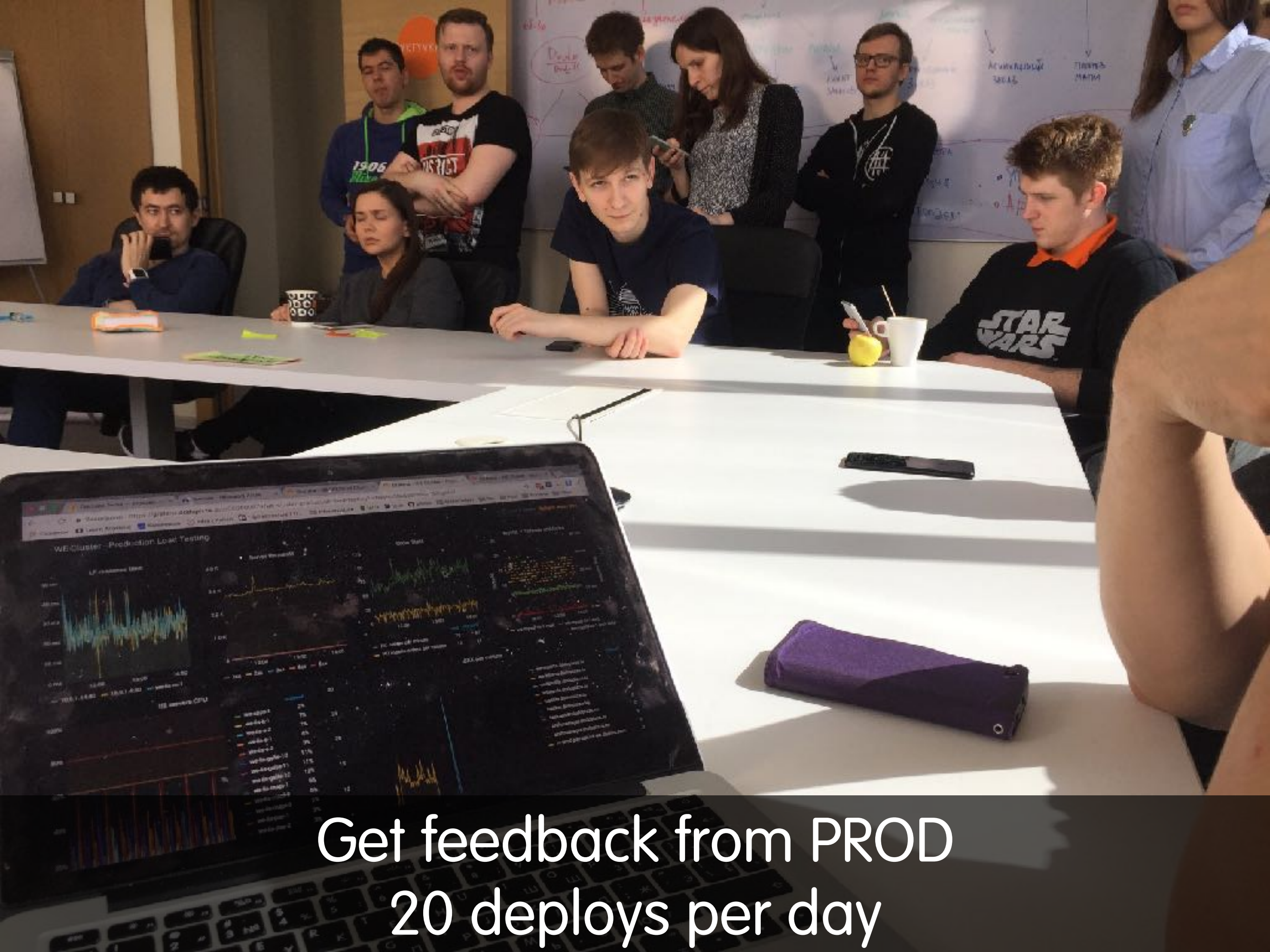
16 April 2018

Focus on ONE goal



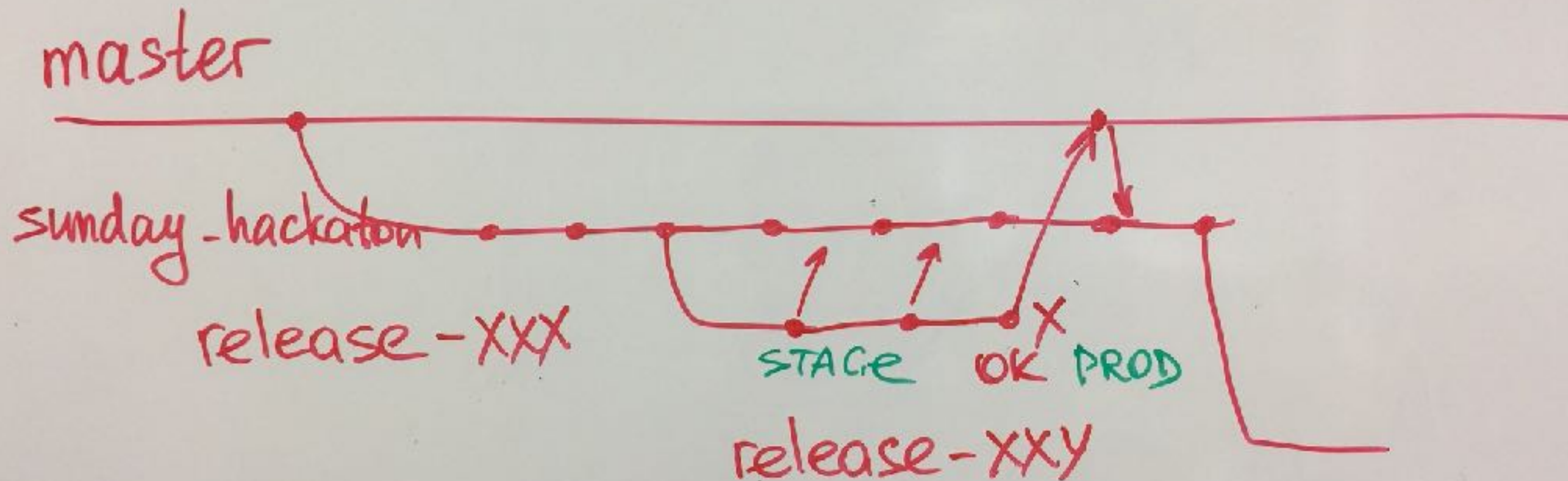


All teams in a single room
synching twice daily

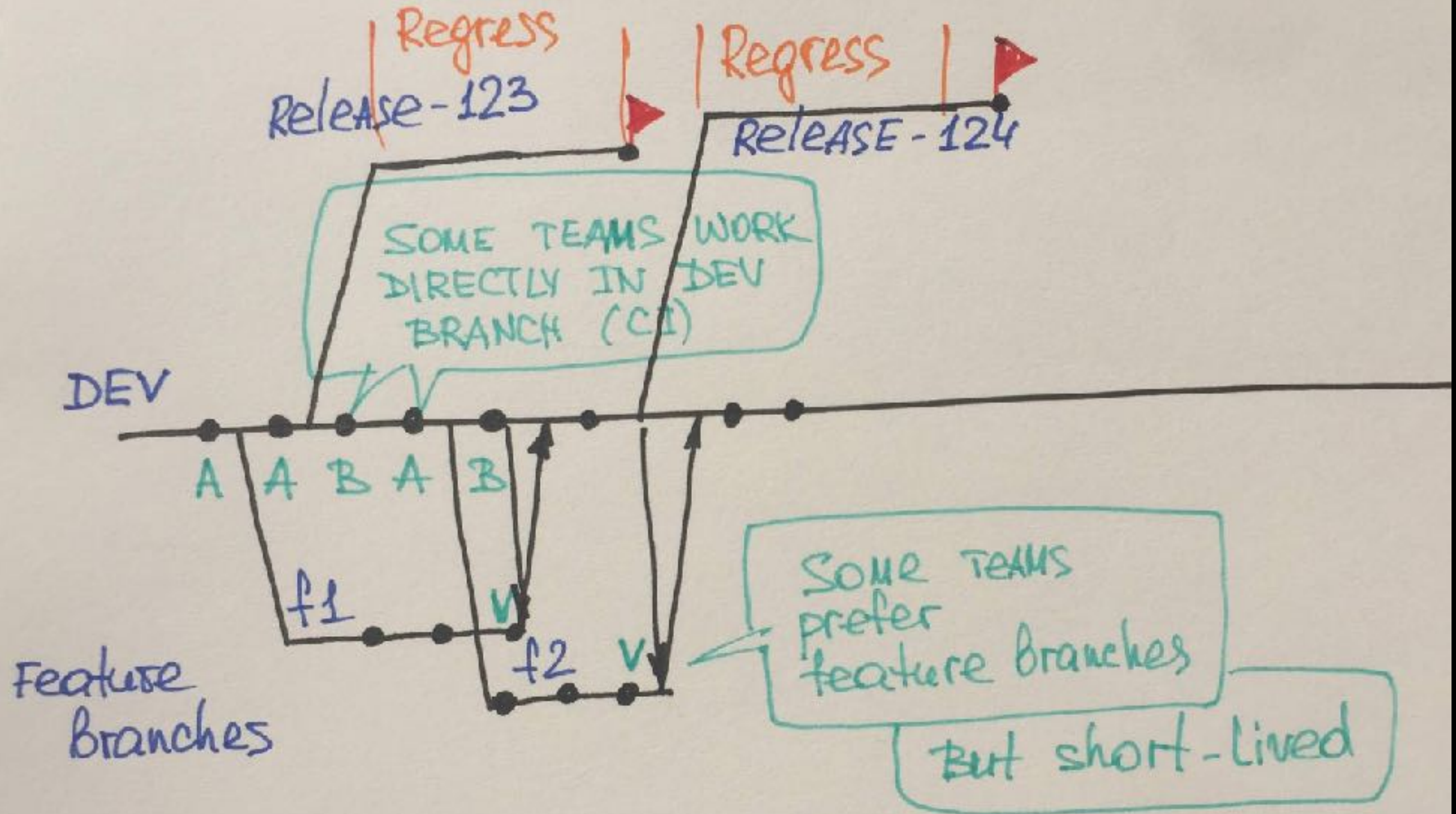


Get feedback from PROD
20 deploys per day

Ongoing releases



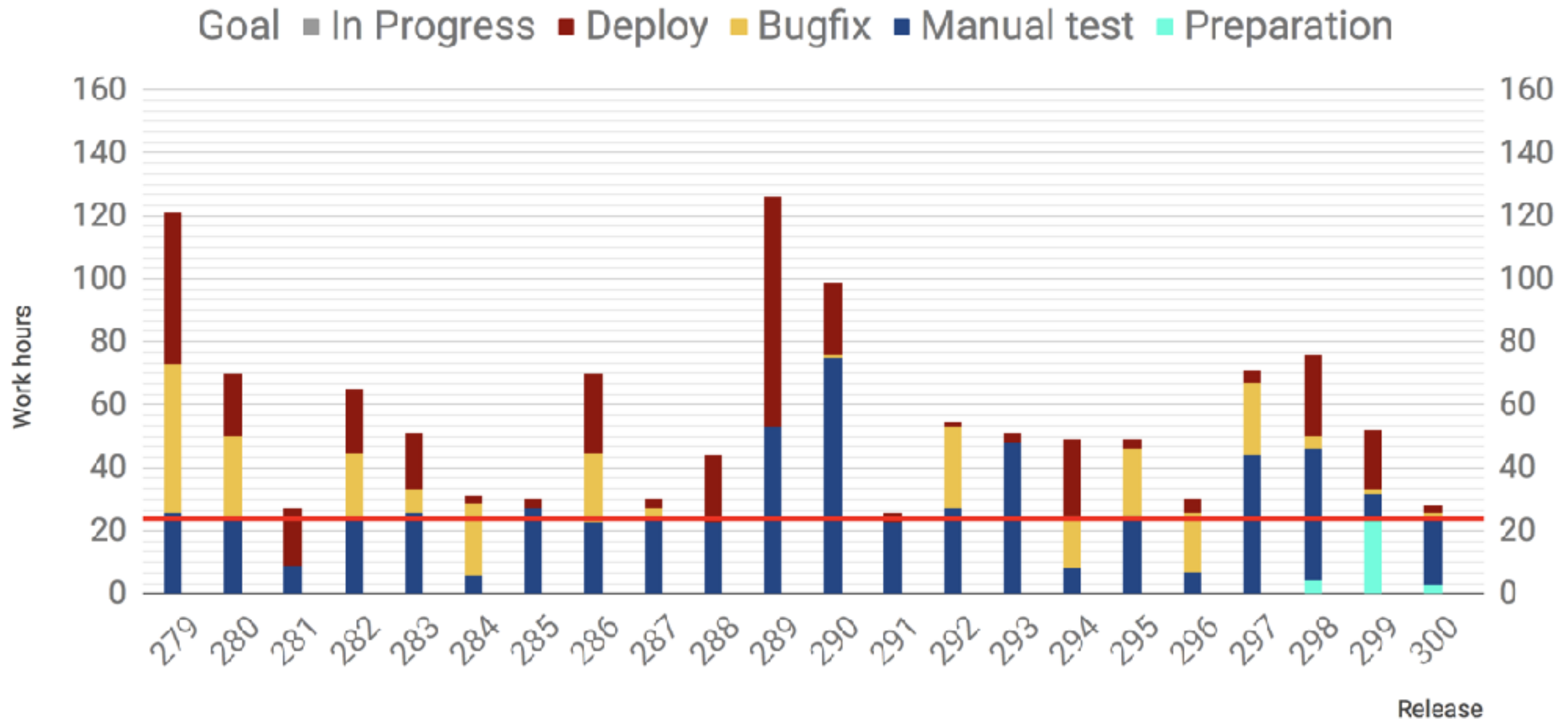
Ongoing releases

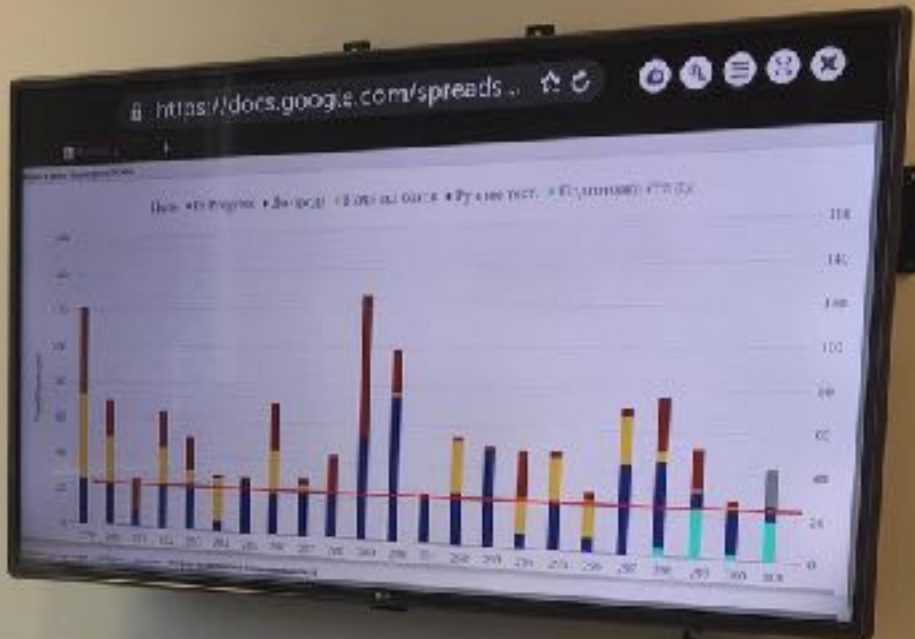


“Merge conflicts are gone.
I can't remember when it was the last
time I resolved complex merge conflict to
DEV in the last 6 months”.

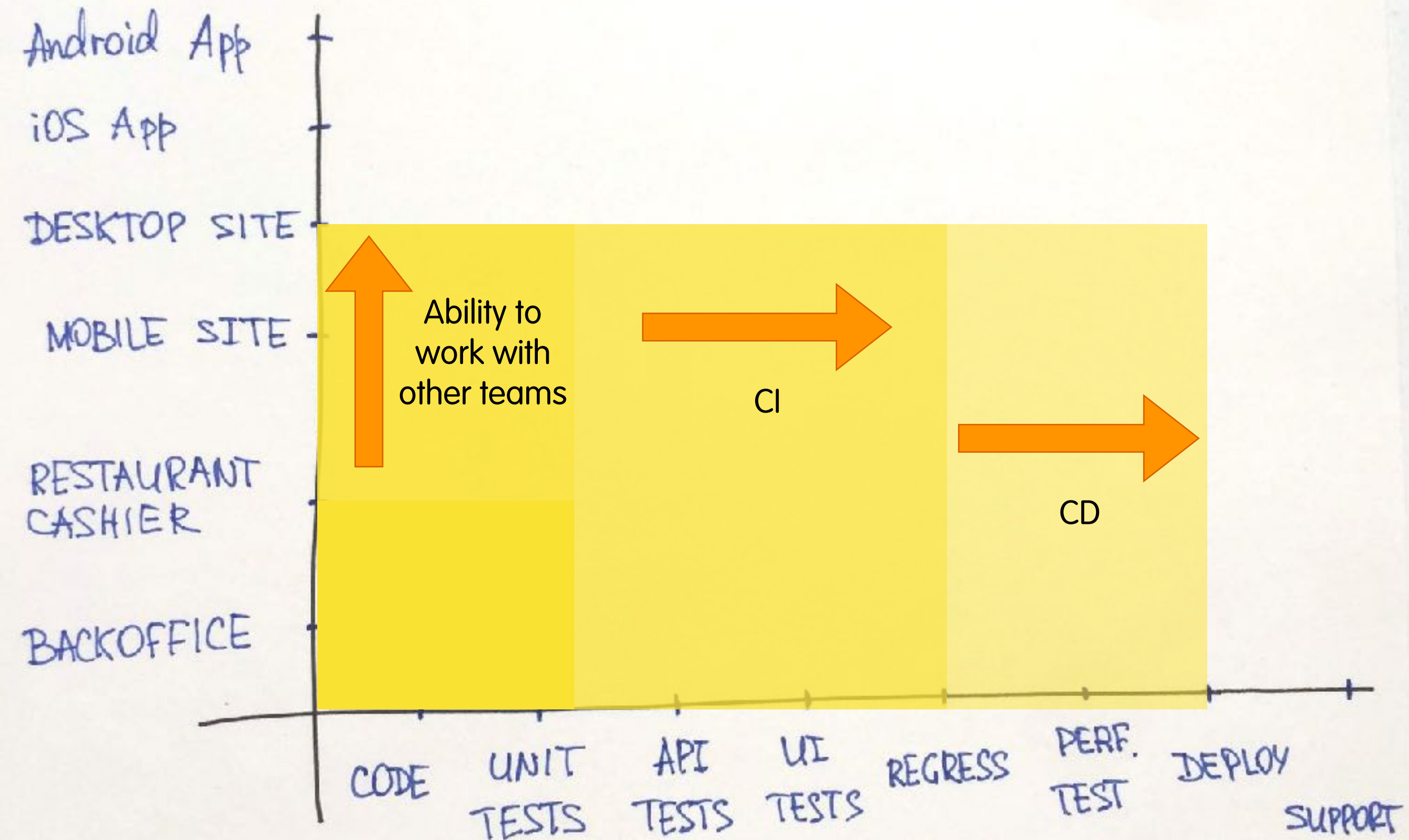
–Vitaly Pomozov, DodolS developer

Releases stat





Continuous Integration in LeSS



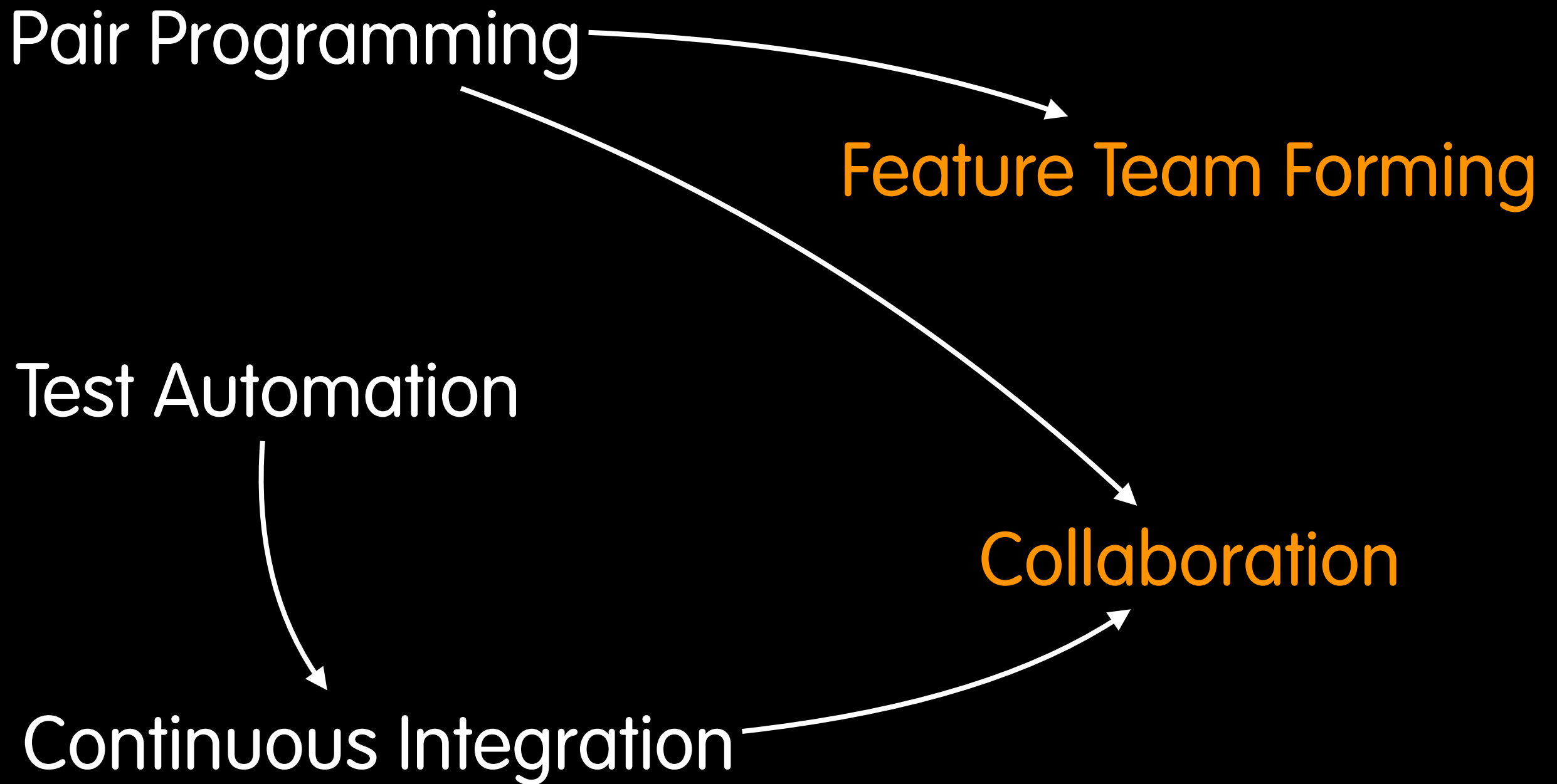
Pair Programming

Feature Team Forming

Test Automation

Collaboration

Continuous Integration



Thanks!



Anton Bevzuk
a.bevzuk@dodopizza.com

- Easy in greenfield, hard in legacy code
- Impediments:
 - Viscosity of the code makes writing unit testing harder
 - Ok, let's write some integration tests -> lack of API makes writing integration tests impossible
 - Ok, let's write some UI tests -> lots of slow, fragile and untrusted tests -> tests are useless
- Solution
 - Split monolith
 - Microservices is not the only option
 - Bounded domain context
 - Start with critical path
 - Refactoring of legacy code book
 - Find the seeds and inject dependencies
 - Extract API where possible
 - Prefer API tests to UI E2E tests

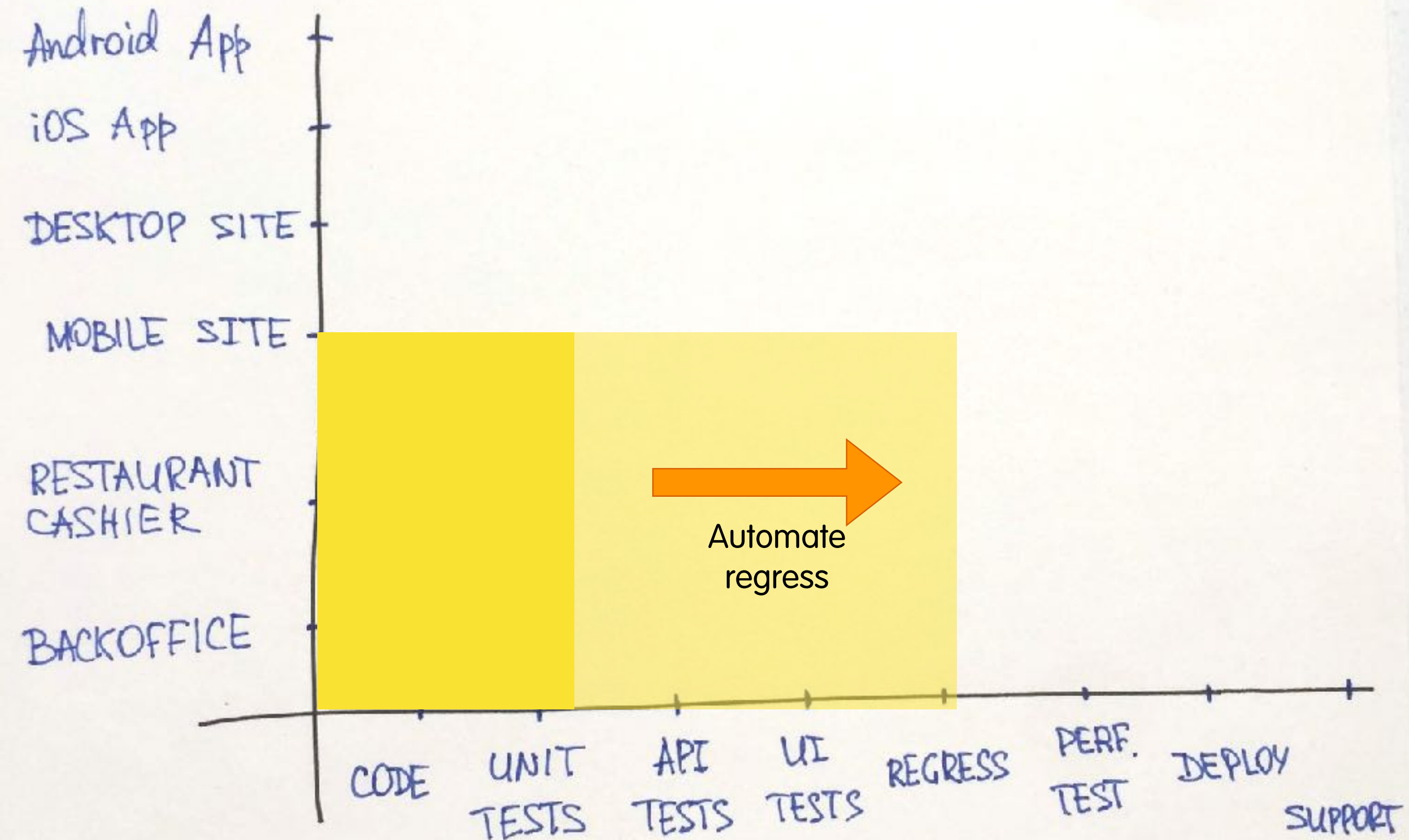
Unit Test example 2

```
[Test]
public void WillChangeProductionStage_IfItIsOnPackingTable()
{
    var trackerOrder = Given
        .Order(state: OrderState.Kitchen)
        .WithVersion(2)
        .WithPizza(version: 1, stage: ProductionStage.OnPackingTable)
        .TrackerOrder();

    trackerOrder.Pack(_localDateTime, new [] {trackerOrder.Line(1).UUid});

    trackerOrder.Product().ProductionStage.Should().Be(ProductionStage.WaitingDelivery);
    trackerOrder.Product().Version.Should().BeIncrementedTo(2);
    trackerOrder.Product().ChangeDate.Should().Be(_localDateTime);
}
```

Test Automation in LeSS



How to start

Hire

Educate

Experiment

Evolve