

SQA DAYS / 28

/ Как сократить время на регрессию и не сыпать ошибки в прод?



## Марина Куликова

IT BDD — BSL

Директор по развитию бизнеса  
в IT, компания BSL.

Место работы:  
BSL — разработчик веб-сервисов  
и мобильных приложений,  
высоконагруженных систем  
и интеграции

« Более 10 лет работаю в отрасли.  
Имею опыт в мобильном,  
десктопном и веб-тестировании,  
занимаюсь оптимизацией  
процессов производства  
и тестирования.

Активно развиваю QA-сообщество  
в России

Основала QA School в Redmadrobot »

цель доклада

# Оптимизация— регресс— опыт

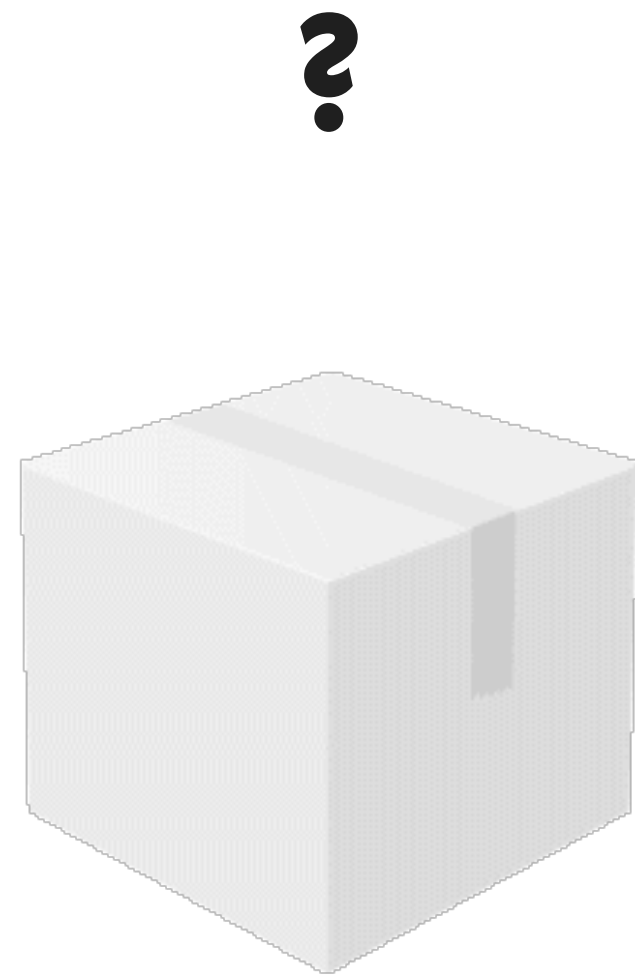


## план доклада

1. Регресс — как много в этом слове
2. Основные принципы оптимизации регрессивных тестов
3. Альтернативные принципы оптимизации
4. Лайфхаки о которых стоит всегда помнить



# регресс

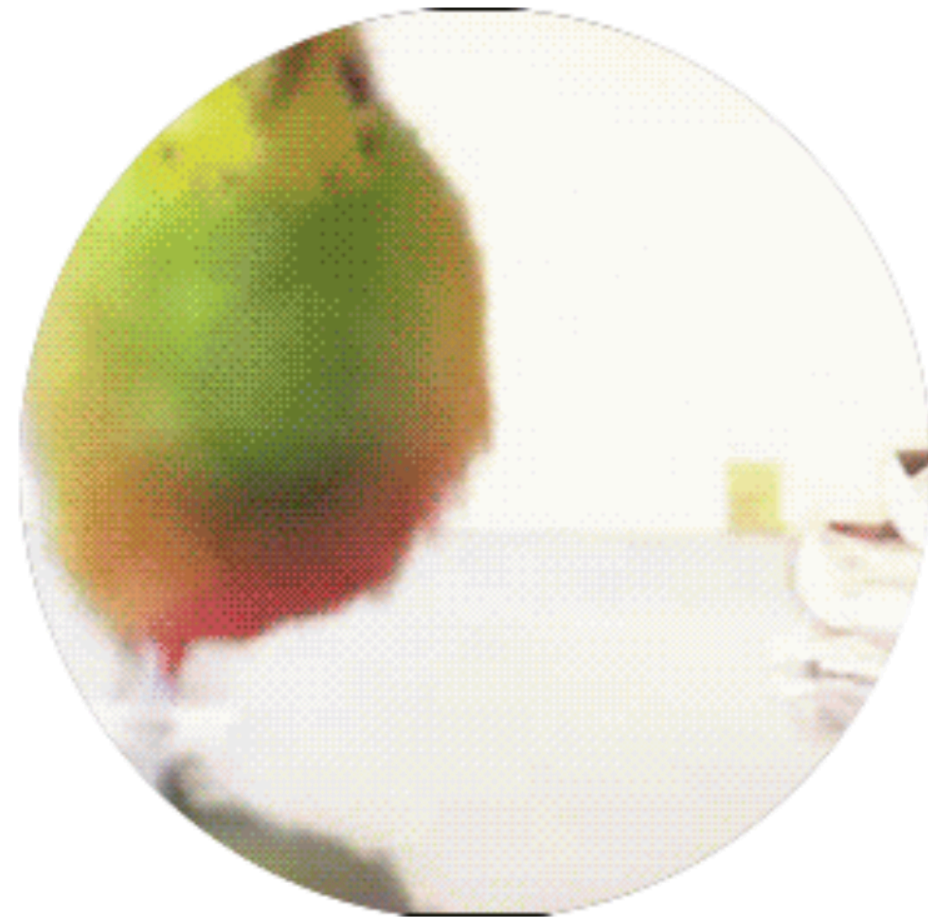


# регресс

**страшно!!!**

**НЕ ПОНЯТНО**

**а-а-а-а-а-а-а....!**



# Что же такое регрессионное тестирование?

Регрессионное тестирование — это набор тестов, направленных на обнаружение дефектов в уже протестированных участках приложения. Делается это совсем не для того, чтобы окончательно убедиться в отсутствии багов, а для поиска и исправления регрессионных ошибок.



# Основная проблема — большое количество тестов

## причины

1. Много ручных тестов
2. Очень много нестабильных автотестов
3. Огромная сложная интеграционная система требующая большого количества тестов
4. Сильная профдеформация тестировщиков в выделенных командах регресс
5. Выгорание тестировщиков
6. Нехватка времени и ресурсов

# Основные виды оптимизации тестирования

01

**Двухуровневый подход**  
(two-level approach)

02

**Коллективный подход**  
(collaboration approach)

03

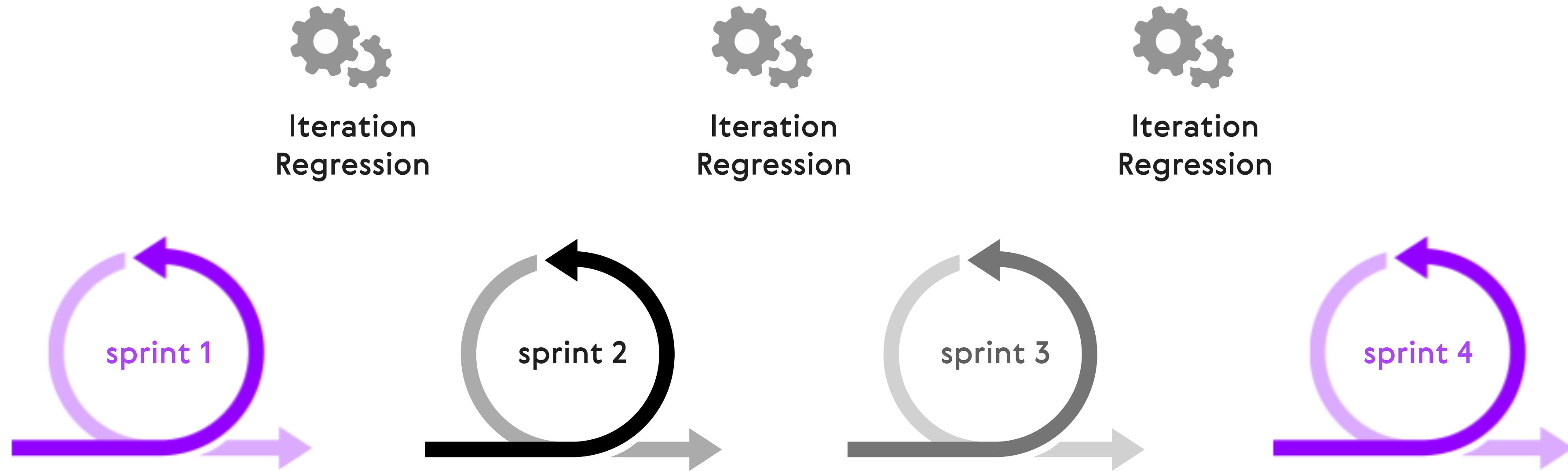
**Оценка рисков**  
(risk-based approach)

04

**Автоматизированный подход**  
(Automation Regression)

# Двухуровневый подход

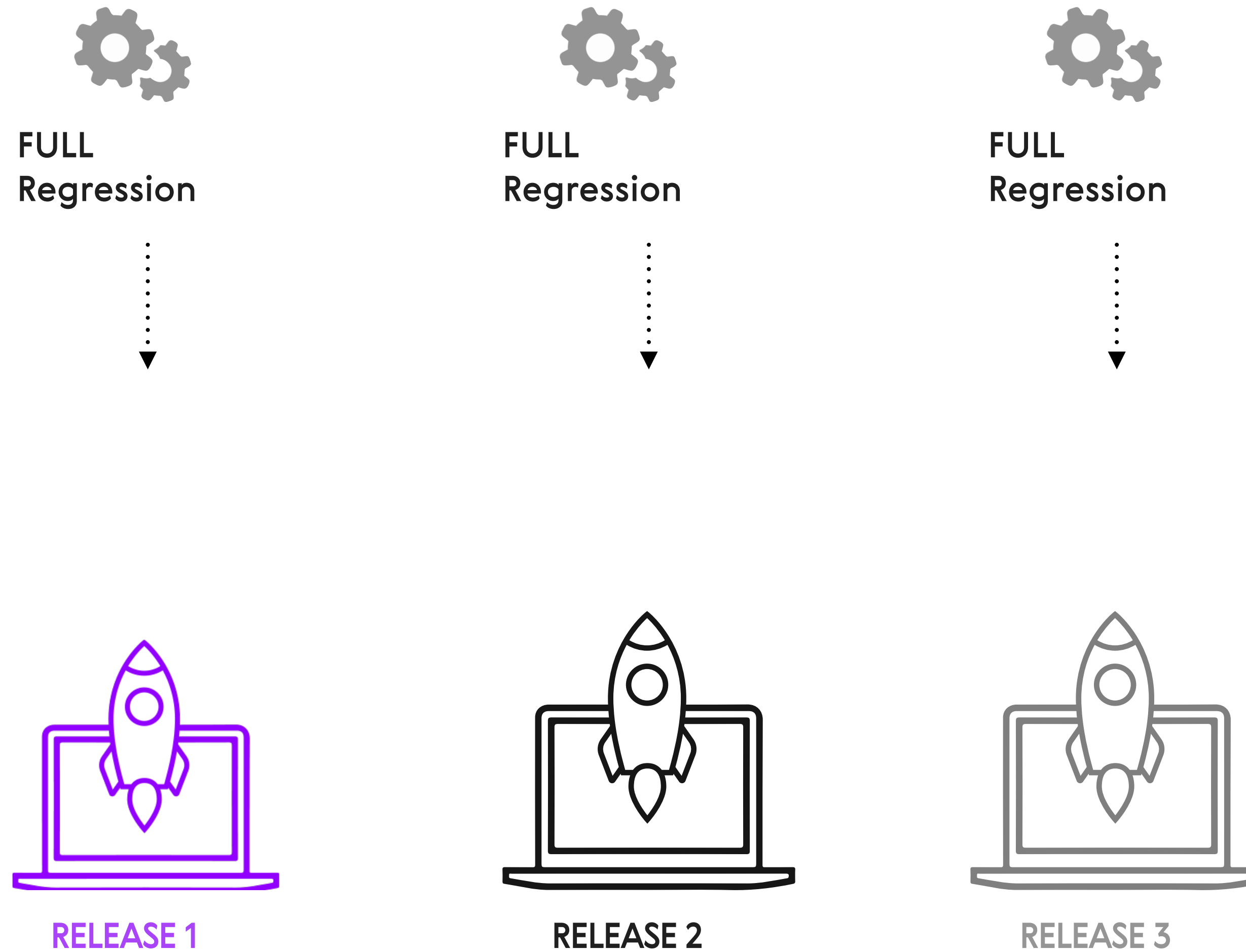
## LEVEL 1



Marina Kulikova

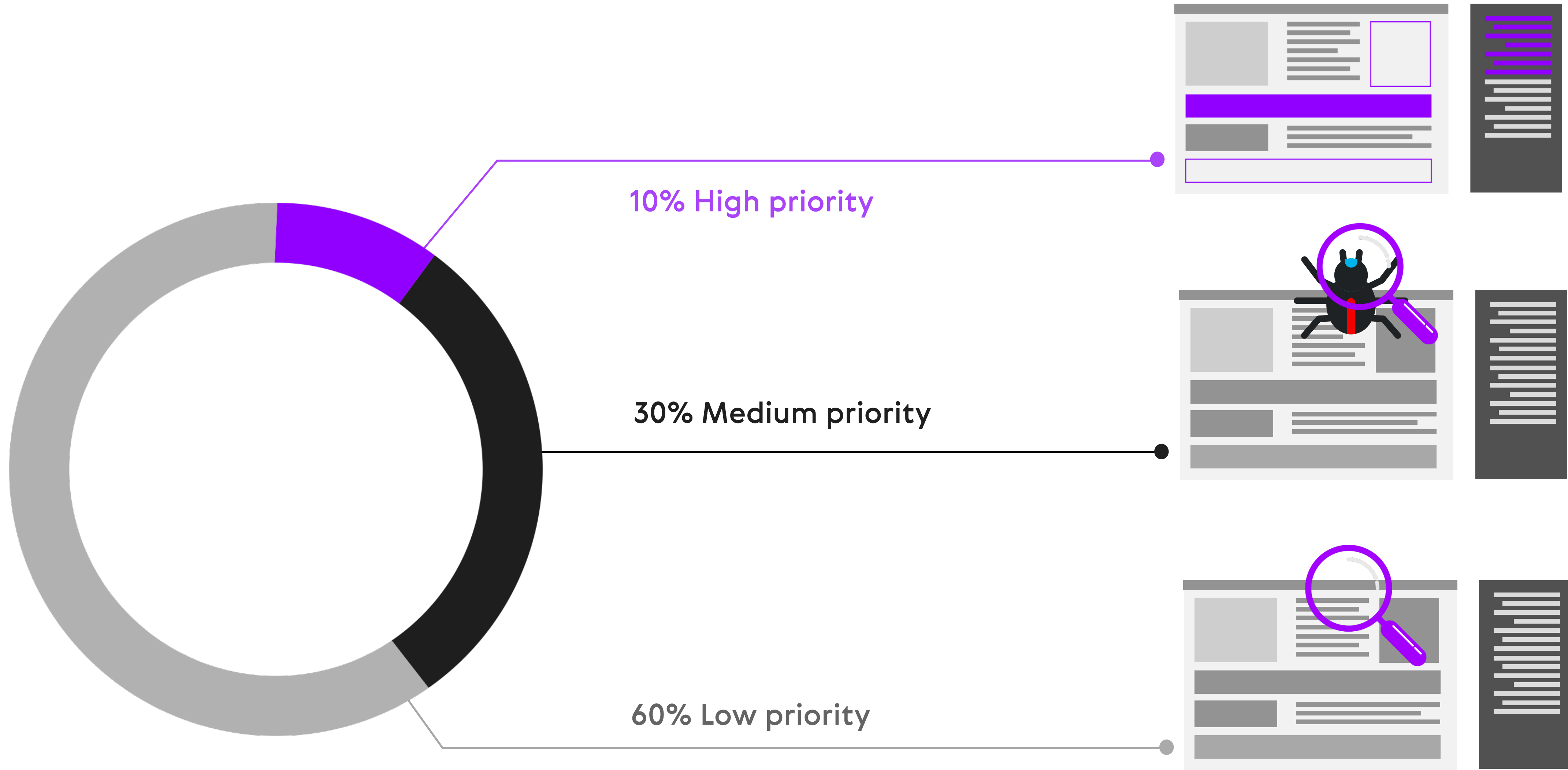
Оптимизация регрессивных тестов

# Двухуровневый подход LEVEL 2



# ПОДХОД НА ОСНОВЕ РИСКОВ

Marina Kulikova

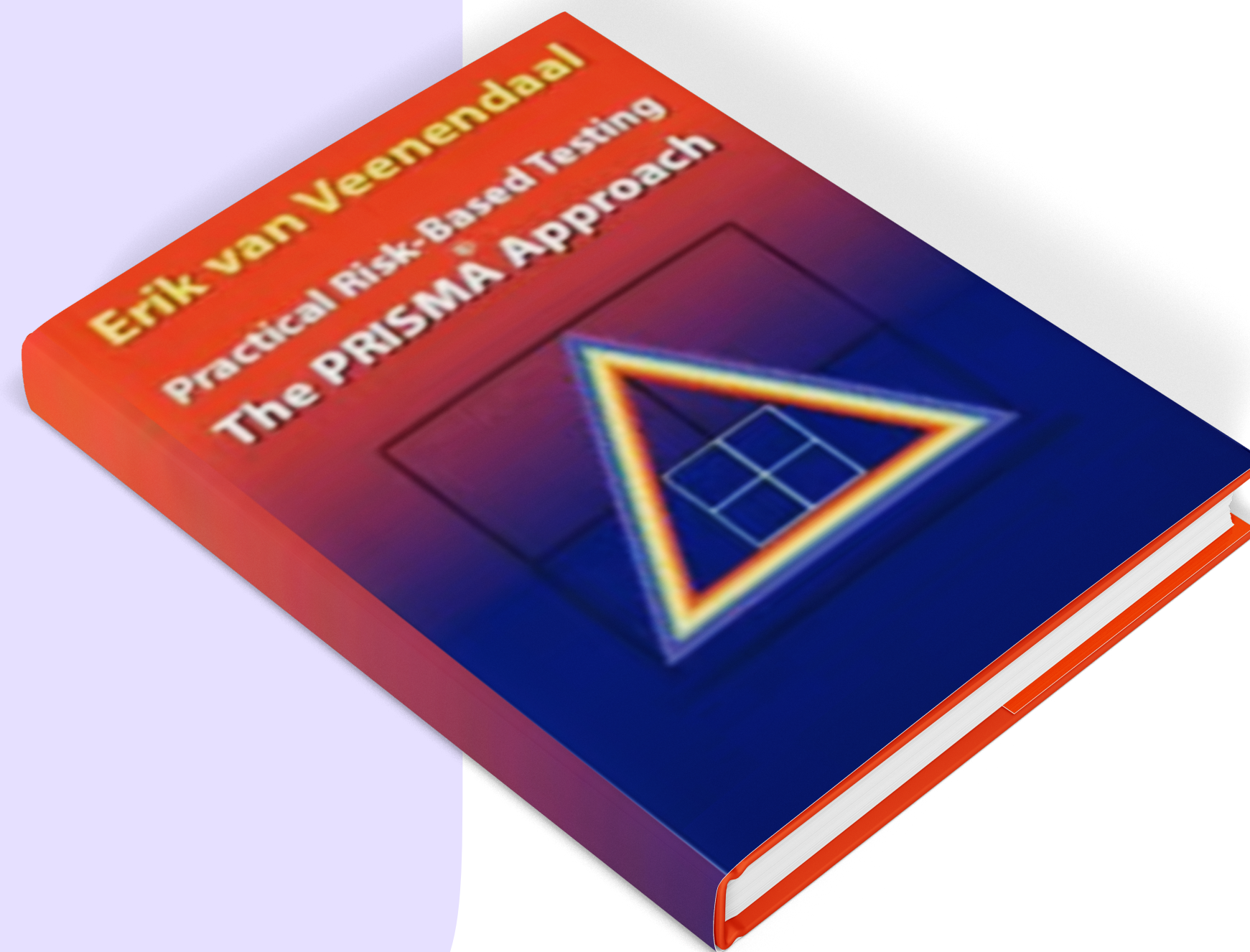


Оптимизация регрессивных тестов

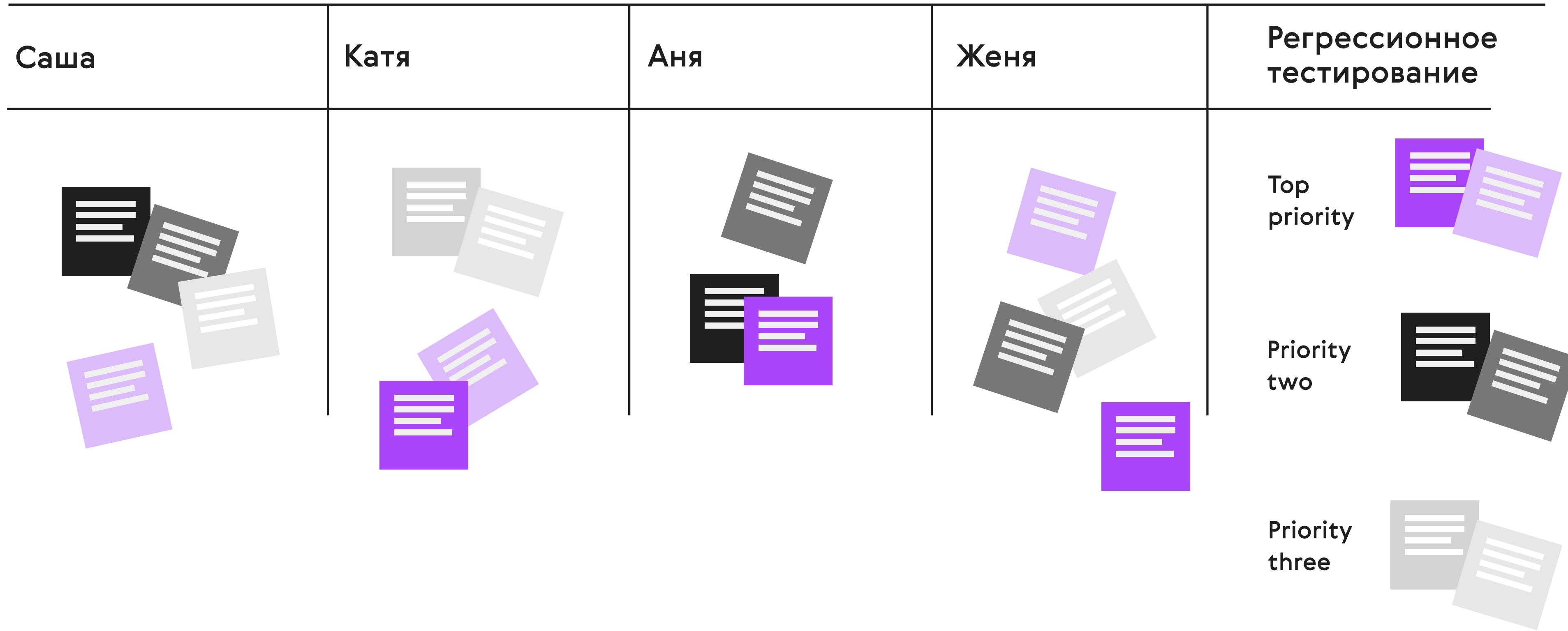
## ПОДХОД НА ОСНОВЕ РИСКОВ

Хочешь узнать  
больше про риски?

— читай!



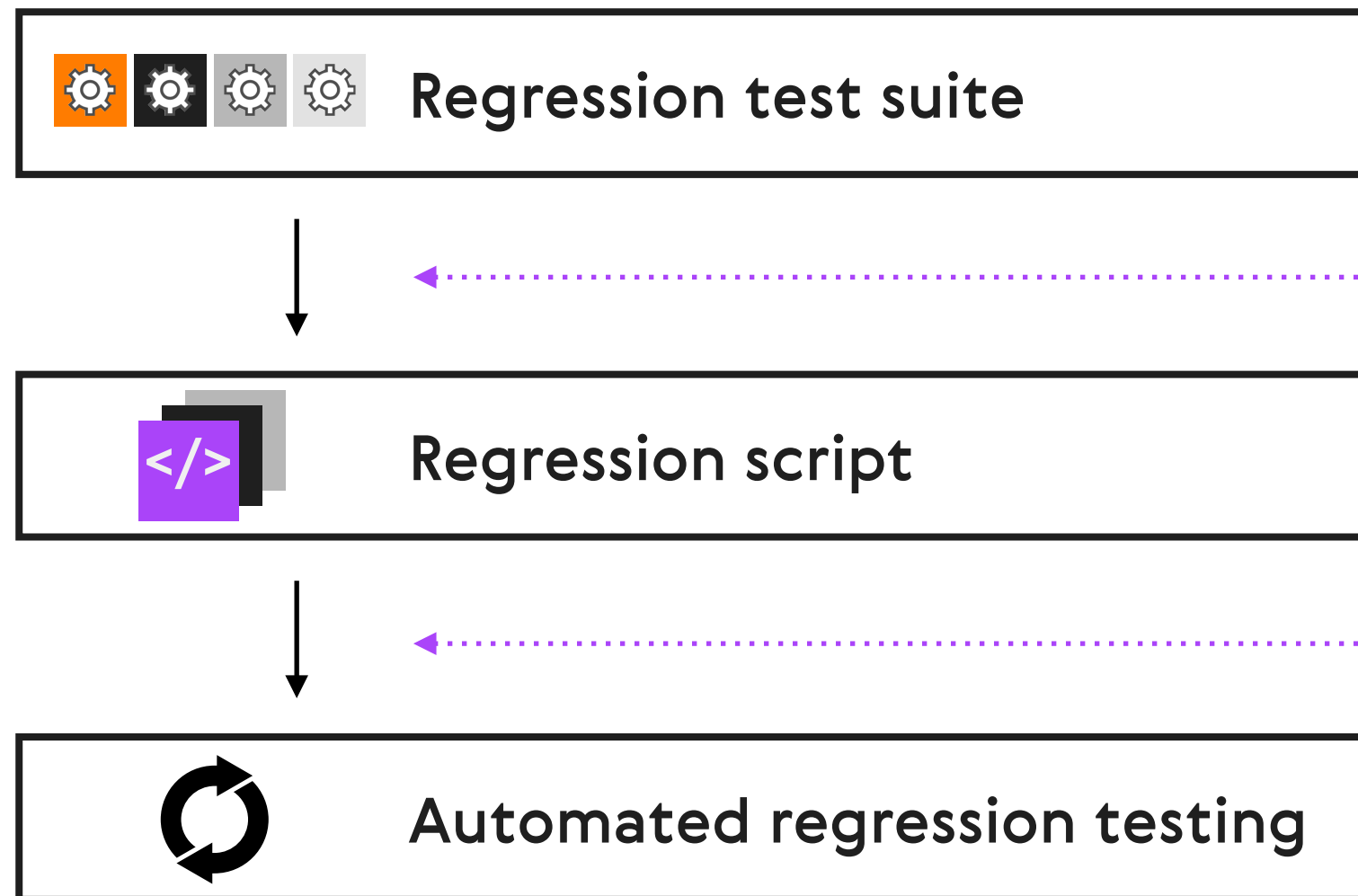
# КОЛЛЕКТИВНЫЙ ПОДХОД



Marina Kulikova

Оптимизация регрессивных тестов

# АВТОМАТИЗАЦИЯ



# альтернативные способы ОПТИМИЗАЦИИ

01

**Impact Analysis**

02

**People management**

03

**Pairwise Testing**

# ИМПАКТ АНАЛИЗ

## Основная боль

Много ручных  
регрессивных тестов



## Как так вышло

- Функционал добавляется. Добавляются тест-кейсы, тестовая модель растёт, следовательно, и регрессионный съют тоже.
- Старое никто почти не выпиливает или не актуализирует.
- Тесты составлены плохо, много лишних проверок.
- «Релиз-трейн» набрал обороты, никто не занимается оптимизацией тестовой модели — все регрессят.

# ИМПАКТ АНАЛИЗ

## Пример

- Есть люди которые постоянно занимаются регрессом
- Люди выгорают от монотонной работы. Баги таки все чаще на проде , ибо глаз замылен
- Отдали регресс в команды — не помогает, объем большой плюс стопорится работа над релизами
- Привлечение больше ресурсов даже в виде разработчиков — не помогло. Багов больше новые фичи стоят



# ИМПАКТ АНАЛИЗ

## Решение

### impact analysis

чтобы сократить время на тестирование, определяй, что было затронуто изменениями и тестируй только эти области системы.

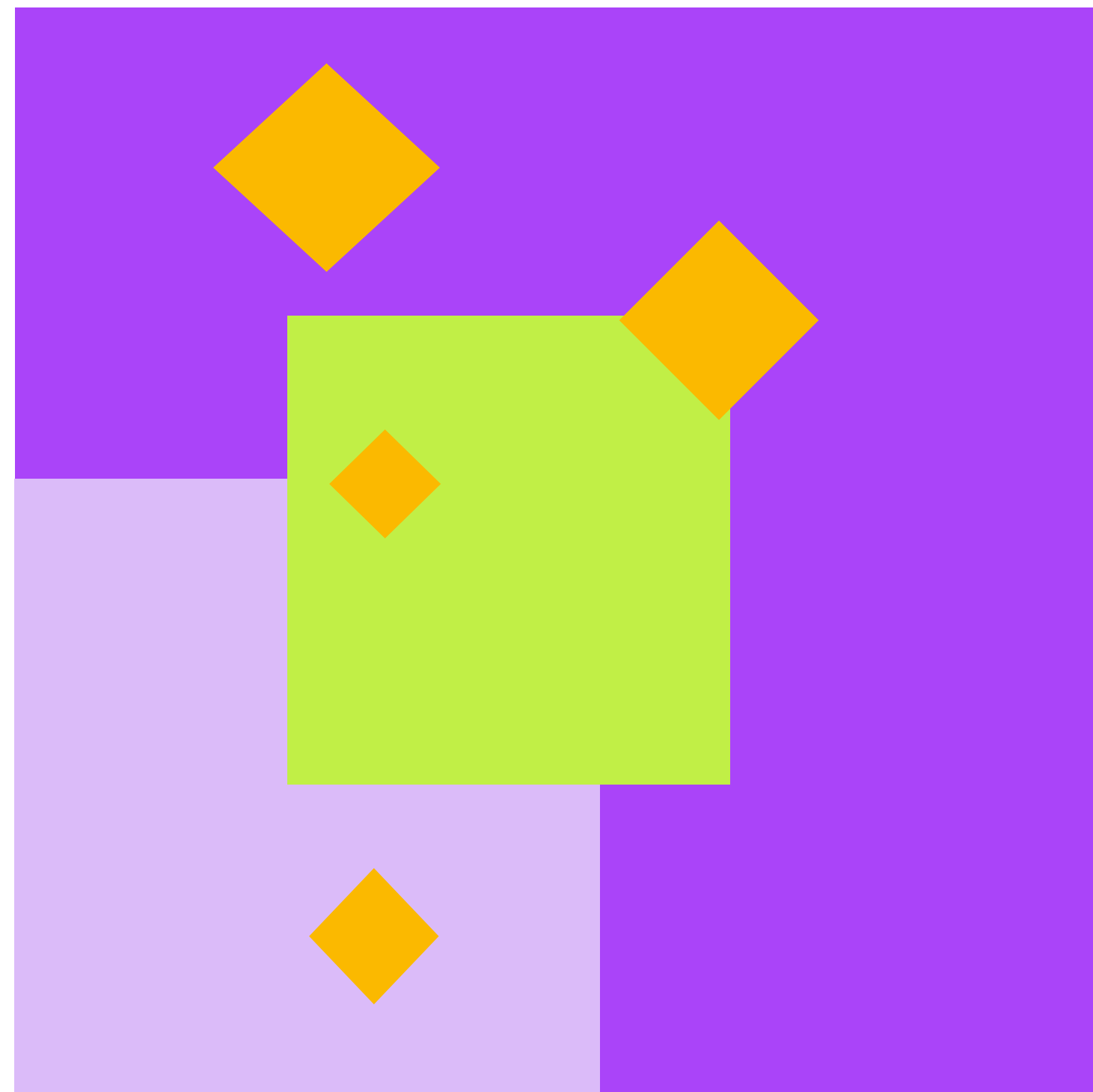
## Когда стоит применять импакт-анализ:

- когда вопрос «что тестировать?» генерирует ответ «всё»;
- при изменении сложно-связанной логики в продукте;
- когда хочется тестировать быстро и эффективно.

Применение импакт-анализа позволяет ответить на вопрос «что нужно, а что не нужно тестировать?».



# ИМПАКТ АНАЛИЗ



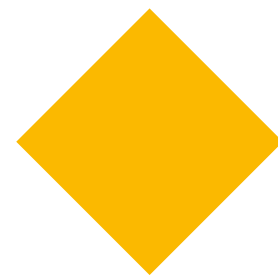
Весь скопу тестов — вся система



Покрытие регрессом



Внесенное изменение в систему



Баги и что реально нужно было тестить

# ИМПАКТ АНАЛИЗ

## Как реализовать?

- В середине спринта берем девелопера смотрим на его изменения в текущем релизе
- Пытаем его до тех пор пока он не осознается во всех места что затронул
- Собираем тестовый набор который тестирует измененный функционал, вместо того чтоб гонять всю регрессию

P.S. Альтернатива Матрицы зависимостей смотреть тут



# ИМПАКТ АНАЛИЗ

## Вывод:

- + быстрый результат
- + легкое внедрение
- + входит привычку

- тяжело внедрить, если полностью новая команда или достался чужой код
- +/- надо постоянно практиковаться и частично документировать

# People management

## Основная боль

- Много ручных регрессивных тестов
- Автоматизация только внедряется
- Команда из нескольких тестеров
- Релизы около месяца, регресс занимает около 3-4 дней



## Как так вышло

- На проекте поменялась команда — к нам пришли новые сотрудники
- На момент смены команды было 500-600 кейсов (уже максимально оптимизированных) на его прогон выделялось 4-5 дней.
- Естественно, нам не всегда хватало этого времени, ведь новые сотрудники проходили кейсы не так быстро.

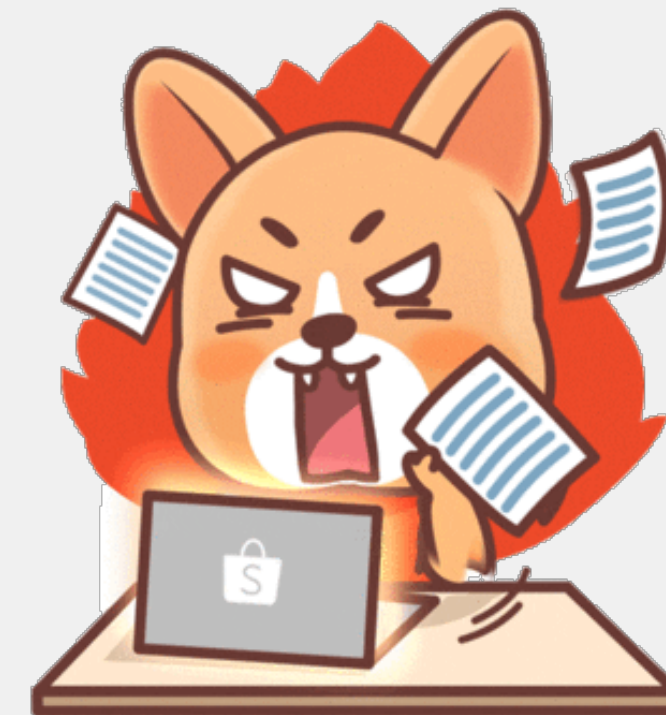
## Цель

Сократить время регресса до 1-2 дней

# People management

## Решение

Включить максимум лайфхаков менеджмента и основ тестирования и сделать невозможное — возможным



# People management

## Что делать?

1.

**Распределить задачи между ребятами по смарт**

- выявить что любит сотрудник что не любит
- договорится что от регресса к регрессу может меняться ответственный
- каждый дополнительно оптимизирует свой скоуп

2.

**Проводить еженедельные 15-ти минутки по обучению или парному тестированию (вести документик с темами)**

3.

**Автоматизировать рутинную работу и создание тестовой даты:**

- перед регрессом сразу набор данных необходимых создаем док и кто то один все готовит.
- Использовали на практике моки( если сервак упадет) создавали данные и Soap UI, коллекции в Postman и запросы для Charles, дампы платежей и Т.Д

4.

**Автоматизировали кейсы с первым приоритетом отдавали на автоматизацию**

5.

**Тестирование Бизнес сценариев( пишем понятную аксептанс критерию и частично около регресс к ней, передаем на продактов)**

## Результат

1. сократили регресс до 1,5 дней

2. протестировали бизнес флоу

3. обучили всю команду

4. расширили знание продукта

# Pairwise testing

огромная сложная  
интеграционная система  
требующая большого  
количества тестов

## Как так вышло

- Много комбинаторики из-за большого количества параметров, которую нельзя унести на нижние слои пирамиды тестирования.
- Есть функционал, который нужно обязательно проверять e2e-тестами, потому что он очень критичный.
- Случаи, когда участвуют сторонние системы и автоматизация очень сложна (делать моки дорого, например).
- Автоматизация не настроена, все гоняем вручную.

# Pairwise testing

## Решение

Это техника тест-дизайна для формирования наборов текстовых данных. Считается, что большинство ошибок в программах вызвано взаимодействием не более двух факторов, поэтому достаточно проверить только **все парные комбинации значений**.

У нас есть система, которая принимает на вход значения. Нужно составить датасеты для тестирования.

params	Значение 1	Значение 2	Значение 3	Значение 4
Enabled	ИСТИНА	ЛОЖЬ	-	-
Choice type	1	2	3	-
Catagory	A	B	C	D

# Pairwise testing

## Решение

Проверка всех возможных комбинаций была бы равна перемножению всех диапазонов значений ( $2 \cdot 3 \cdot 4$ ) = 24 датасета.

Но если применить Pairwise, то при перемножении наибольших значений (3 и 4) получаем 12 датасетов.

Мы получили все возможные пары между значениями. Последние строки отмечены "~", т.к в текущих строках не важно, какие будут значения, - пары уже были проверены в строках 9,10 со значением "3" и в строках 3,4 со значением "с" и "d" соответственно.

№	Enabled	Choice type	Catagory
1	ИСТИНА	1	A
2	ЛОЖЬ	1	B
3	ИСТИНА	1	C
4	ЛОЖЬ	1	D
5	ЛОЖЬ	2	A
6	ЛОЖЬ	2	B
7	ЛОЖЬ	2	C
8	ИСТИНА	2	D
9	ИСТИНА	3	A
10	ЛОЖЬ	3	B
11	-TRUE	3	C
12	-FALSE	3	D

# Pairwise testing

## Пример

	До сокращения	После применения pairwise	Profit
Количество кейсов	45 кейсов	15 кейсов	<b>33,3 %</b>
Время на регресс	14 часов	2,5 часа	<b>82 %</b>

# Pairwise testing

## Инструменты



Pict от Microsoft



Онлайн-инструмент

# Useful tips

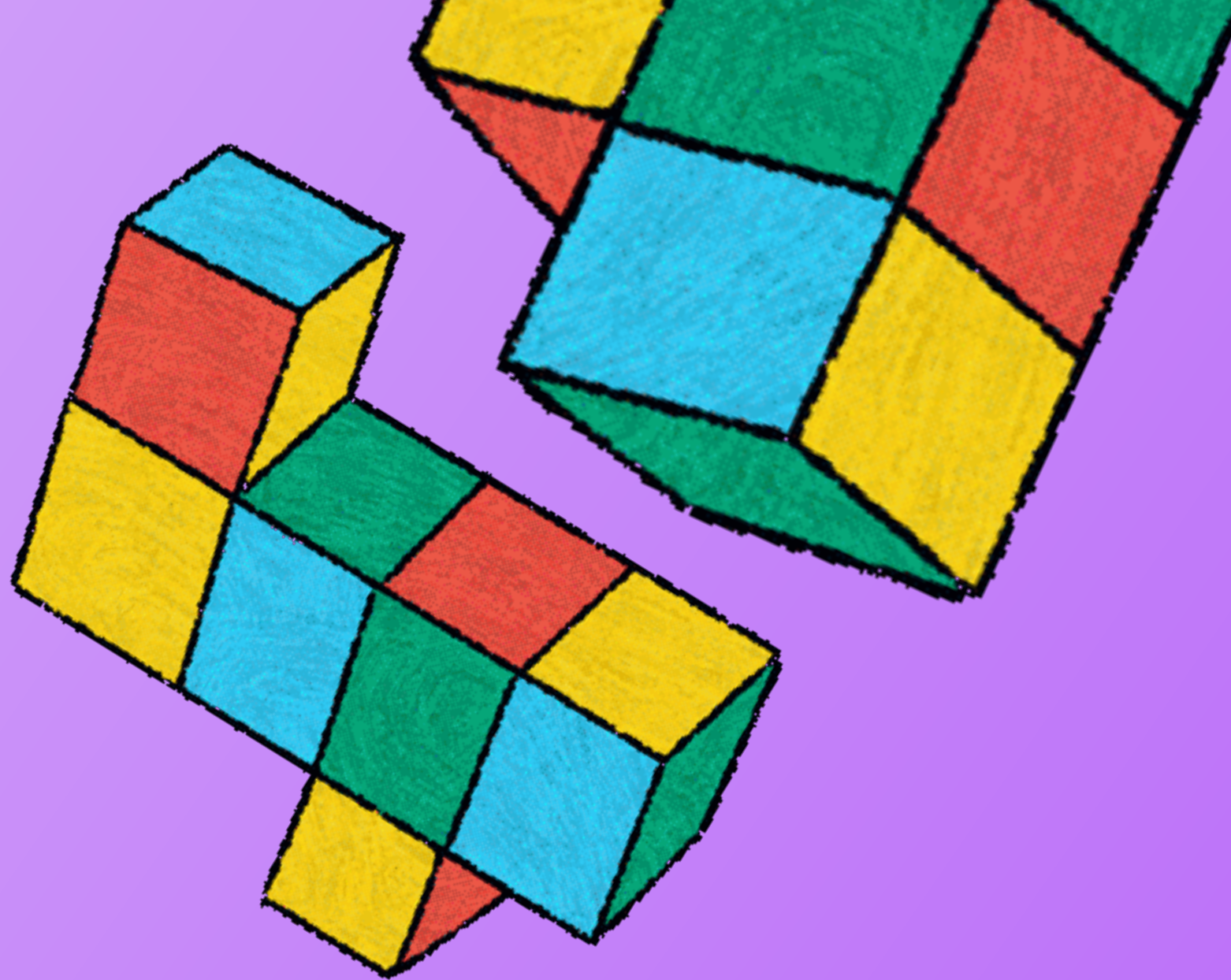
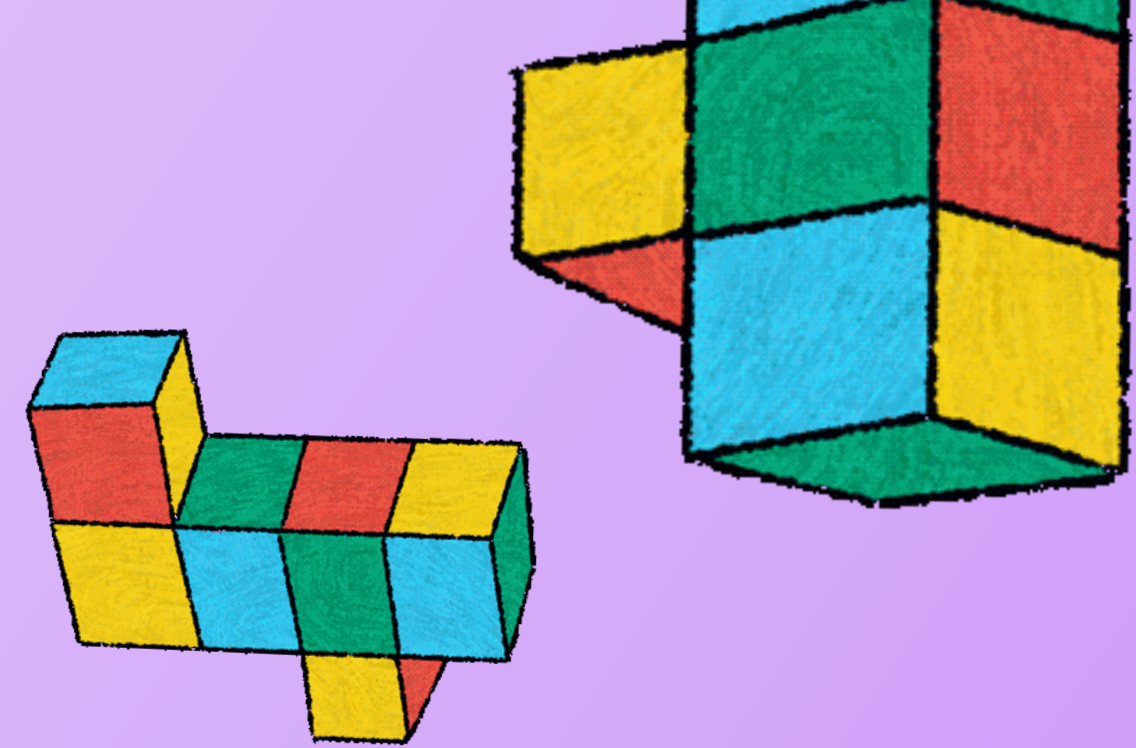
**1.** Основные принципы тестирования  
Особенно принцип раннего тестирования, кластеризации и исчерпывающего тестирования

**2.** Пирамида тестирования  
Помни e2e тесты не так стабильны, переключай на нижние уровни  
Согласуй пирамиду со всеми

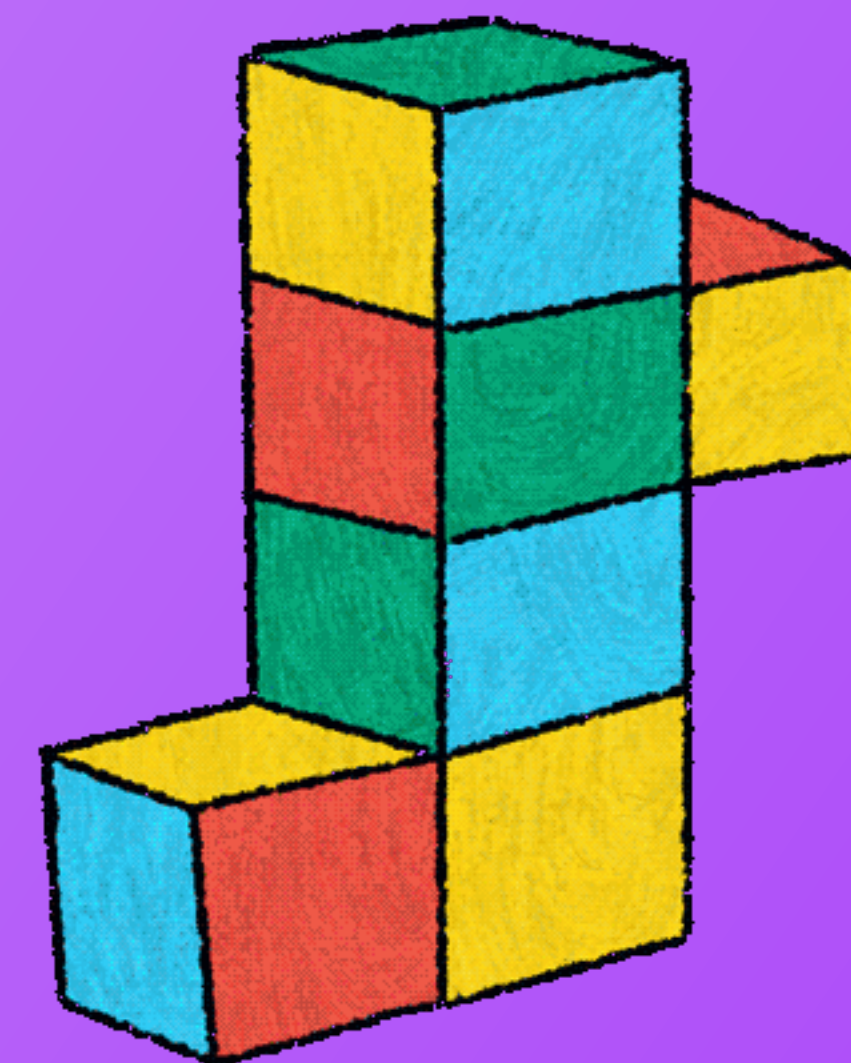
**3.** Парное тестирование с разработчиками

**4.** Применения оптимизации на каждом этапе тестирования

**5.** Техники тест дизайна  
Помни это процесс направленный на составление оптимального набора тестов



Думайте,  
творите  
и комбинируйте!



Есть вопросы по качеству?  
Обращайтесь!



Marina Kulikova

Оптимизация регрессивных тестов

**Куликова Марина**  
IT BDD, BSL

+7 (917) 555-32-55

m.kulikova@bsl.dev  
kulikova.qa@gmail.com

facebook.com/maryna.kulykova

kulikovaqa.com  
bsl.dev

**/ спасибо за внимание!**