



Эффективная организация параллельного тестирования нескольких проектов

Александр Калугин. PMARCOR

Здрасьте, это я!



к.ф.-м.н., PMP

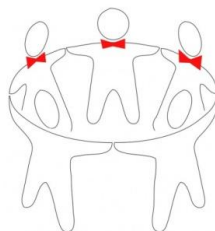


**Менеджер
менеджеров**



Автор

<http://pmarcor.com/>



PMSamara

Соорганизатор

<http://pmsamara.com/>

О чем речь

- Большое количество параллельных проектов
- Проекты с разным процессом
- Широкий спектр технологий
- Короткие или не очень длительные проекты
- Команда тестирования 1-5 тестировщиков на проект
- Сложный софт



А конкретнее?



- Одна но большая или много, но маленьких?
 - сервис или команды?
- Как всё успевать?
 - о параллельных проектах
- Как работать комфортно?
 - о сохранении и переключении контекста

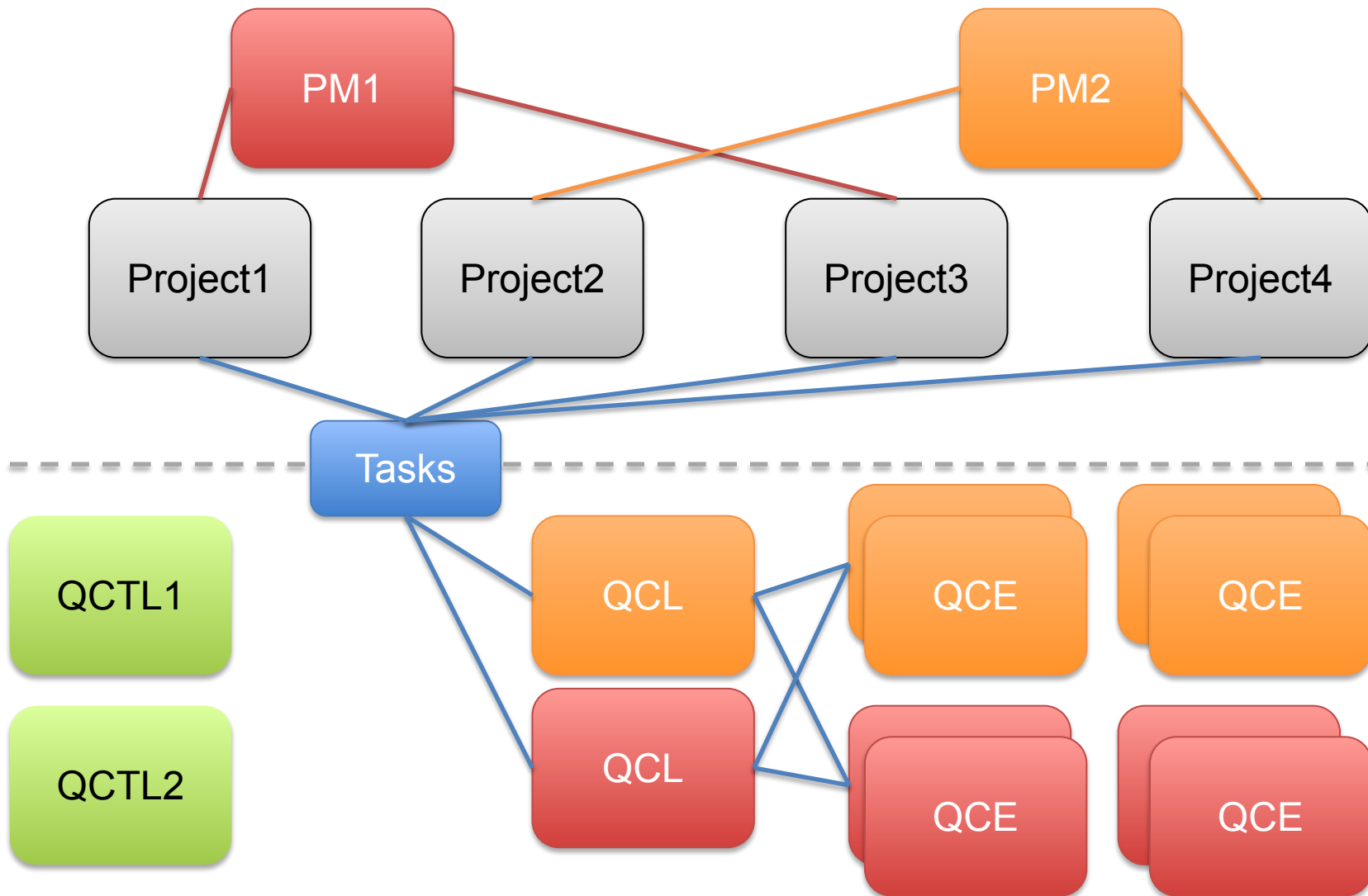
Команда



vs.



Команда: «Чисто» Сервис



Команда: «Чисто» Сервис



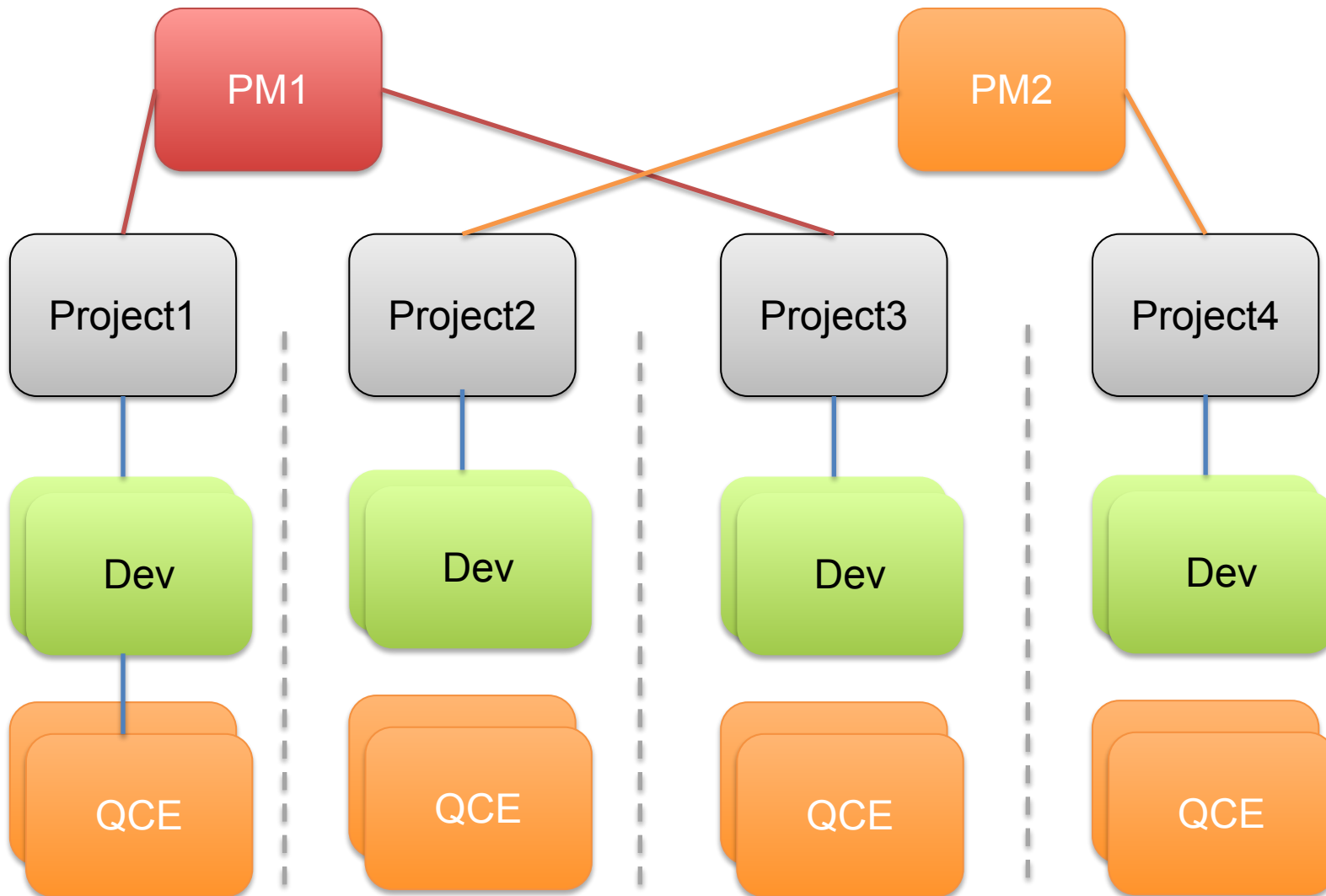
Не работает, так как:

- Никто не понимает, что происходит
- Никто не отвечает за результат
- Неясные приоритеты
- Позднее включение

Работает:

- Компактные задачи вне контекста
- Формализованные процессы

Команда: 1:1 Dev





КТО ВСЕ ЭТИ ЛЮДИ?

Команда: 1:1 с dev



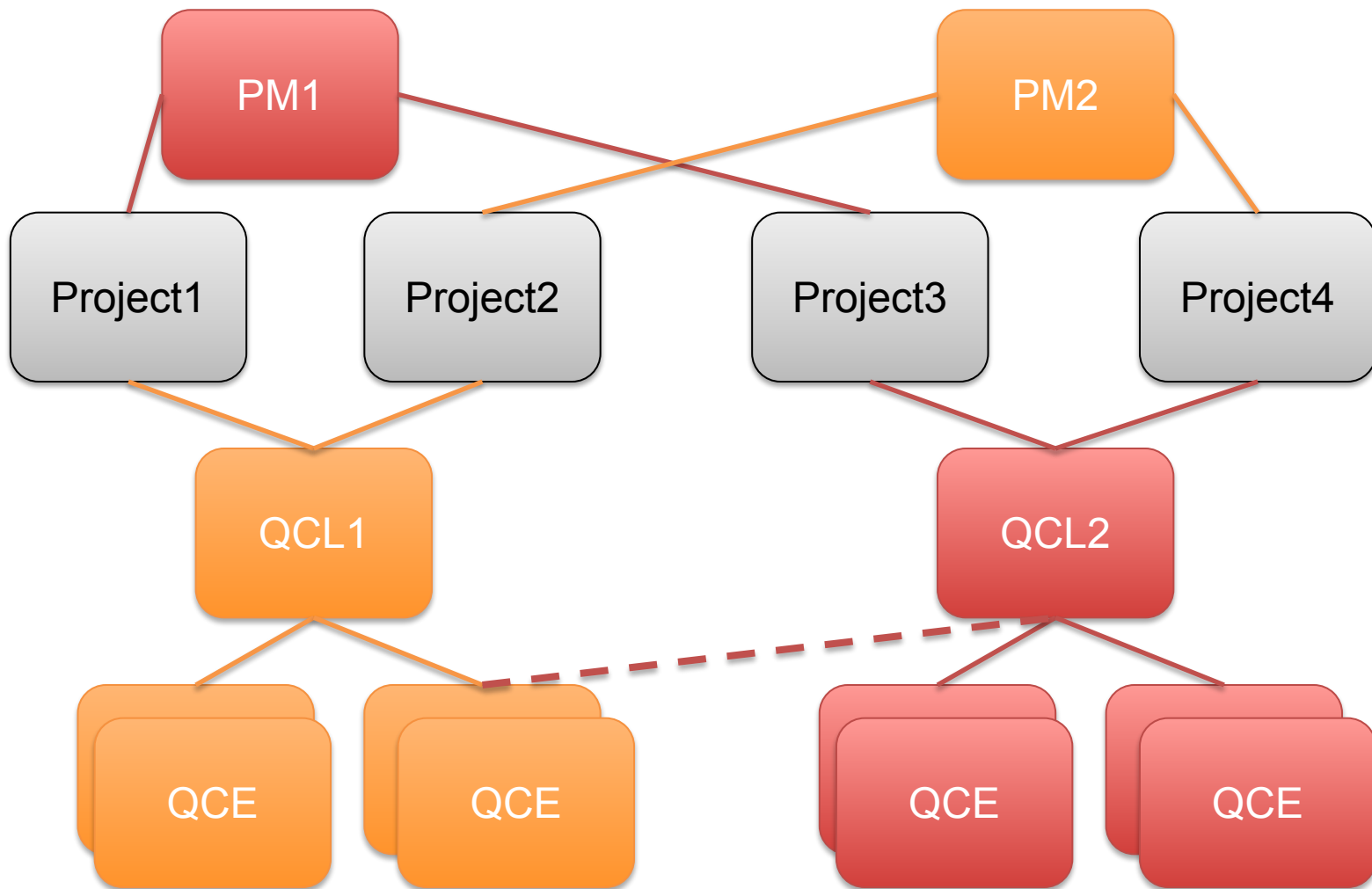
Недостатки:

- Феодальная раздробленность
- Эндемичность
- Нет дома

Работает:

- большие, длительные, итеративные проекты
- четкое соотношение количества участников команды
- не только тестирование

Команда: Баланс



Команда: Что дает?

Достоинства:

- Есть **команда** проекта
Возможно раннее подключение
- Есть отдел тестирования
Переключение между ПМ-ами и проектами
- Синергия проектов

Недостатки:

- Конфликты интересов между проектами



Как всё успевать?



VS.



1. Планирование

- Не совмещать проекты с одинаковой датой выпуска
- Участие в процессе оценивания
- Совмещать проекты со схожим профилем
- Помнить об отпусках
- Промежуточные итерации в разные дни
- Приоритеты определяют ПМ-ы



2. Делать впрок



- Раннее подключение
- Тестирование спецификации, архитектуры
- Анализ рисков и тестирование «от рисков»
- Тестирование ранних билдов, модульное тестирование
- Серый ящик
- Больше информации для дебага
- Экономить итерации

3. Борьба с простоями

- Запасная задача/Plan B, + полдня
- Сделайте мне билд
- Деление full-test-a
- Отложенные недотесты
- Тестирование аналога/ прототипа
- Запасной environment
- Борьба с блокерами
- Проработка чеклиста/ use-case-a



4. Борьба с пробками



- Деление времени
- Сказать как можно раньше
- Уточнение задачи
 - Пропустить билд / часть задач
 - Фокусировка
 - Сужение покрытия
 - Критерии останова
 - Тесты в фоне

4. Борьба с пробками (2)

Тестирование по спирали:

- программисты
- приемка
- smoke
- изменения
- приоритеты
- регресс



5. Déjà vu. Не изобретать велосипед



- Переходы
 - Проекты
 - Команды
 - Функционал
- General Checklist
 - Платформа,
 - Технология,
 - Процесс
- Cross-review
- Обобщения
known-issues и
invalid

Результаты

- более равномерная загрузка, меньше авралов
- меньше трудозатрат
- критичные дефекты раньше
- больше пространства для манёвра



Осталось только...

... не раздвоиться



FAILURE

if you can't blame it on others,
blame it to your other self...

Переключение контекста: Процесс



- Баги проверяют те же, кто нашел.
- По возможности, не делить день.
- Несколько проектов в неделю.
- Схожий профиль.
- По двое.
- Разные роли в разных проектах

Переключение контекста: Инструменты

- Traceability matrix (тесты на билды)
- Нет тесткейсам!
- Чеклисты
- Протоколы сессионных тестов
- Границы разных тестов/четкая стратегия
- Приоритеты
- Анализ wontfix-ов и инвалидов



Результат: удобнее



- больше разнообразия
- выше эффективность
- легче подключение к проекту

Спасибо!

Калугин Александр

info@pmarcor.com

<http://pmarcor.com/>

<http://pmsamara.com>

@pmarcor



Ваши вопросы?

