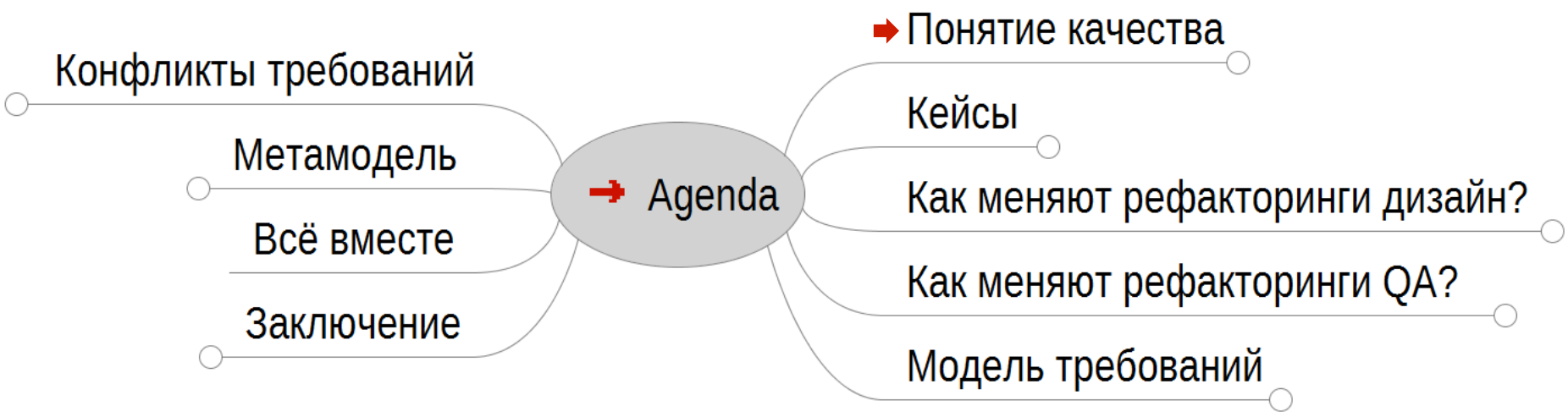
Осознанность рефакторинга

Евгений Кривошеев,

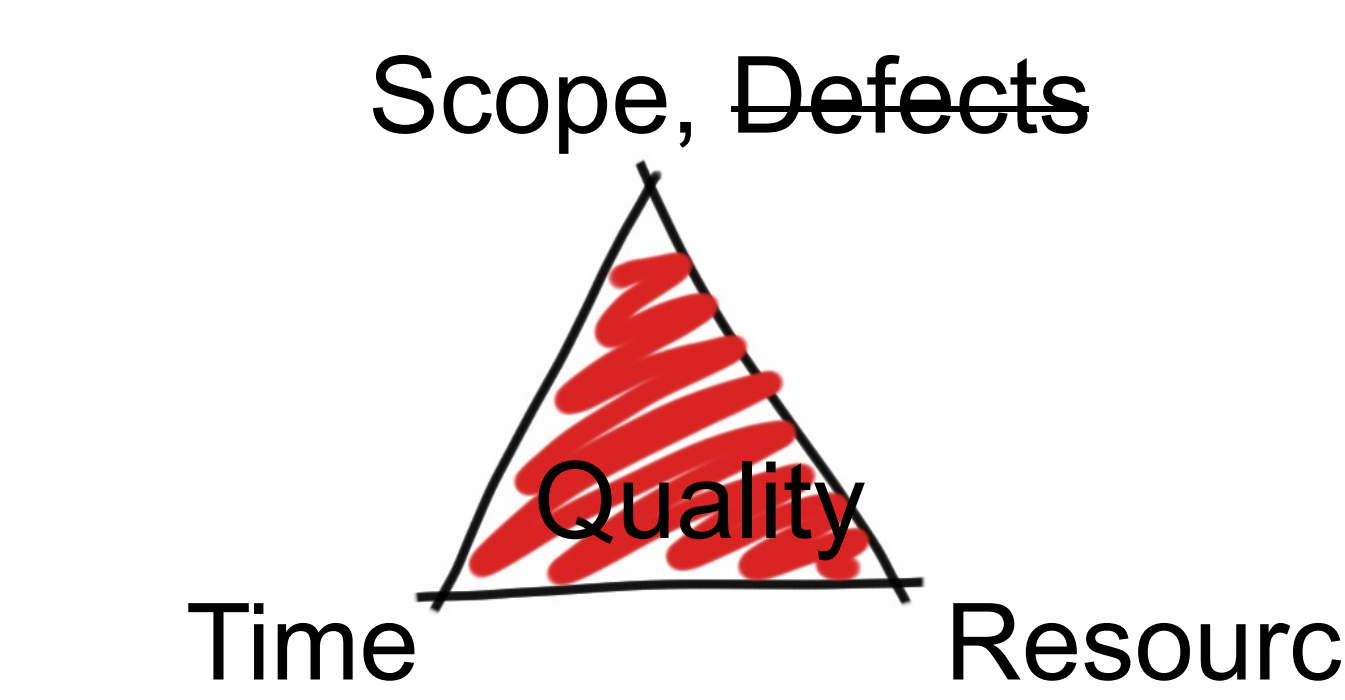
Цели доклада

- Участники смогут принимать осознанные инженерные решения
- Участники смогут обеспечить высокое качество дизайна
- Участники решат как минимум одну нерешенную практическую проблему

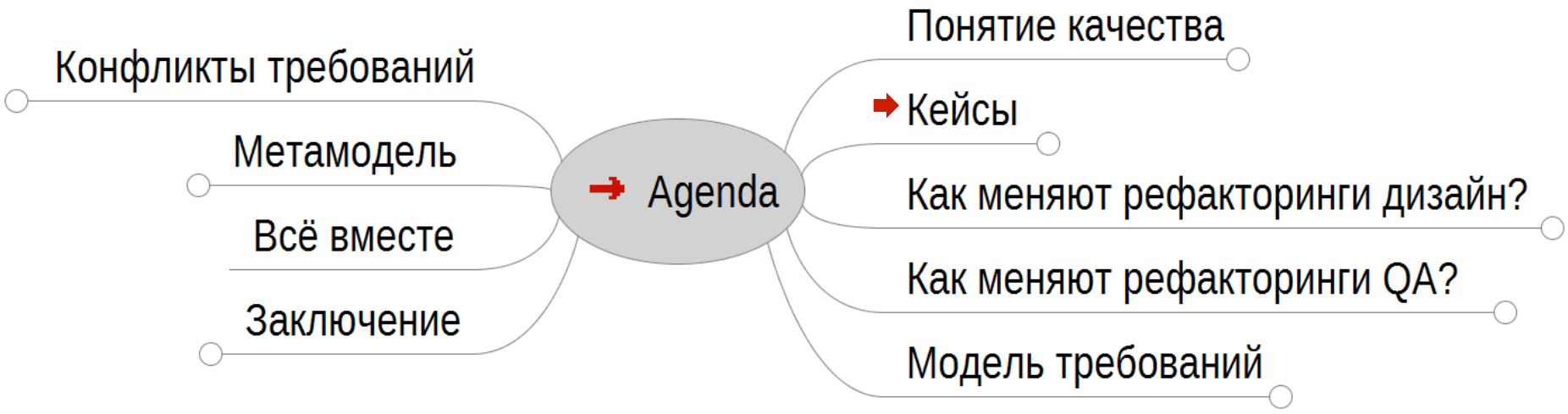
Сценарий


Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований
Конфликты требований
Метамодель
Всё вместе
Заключение

Понятие качества



Сценарий


Конфликты требований
Метамодель
Всё вместе
Заключение
Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований

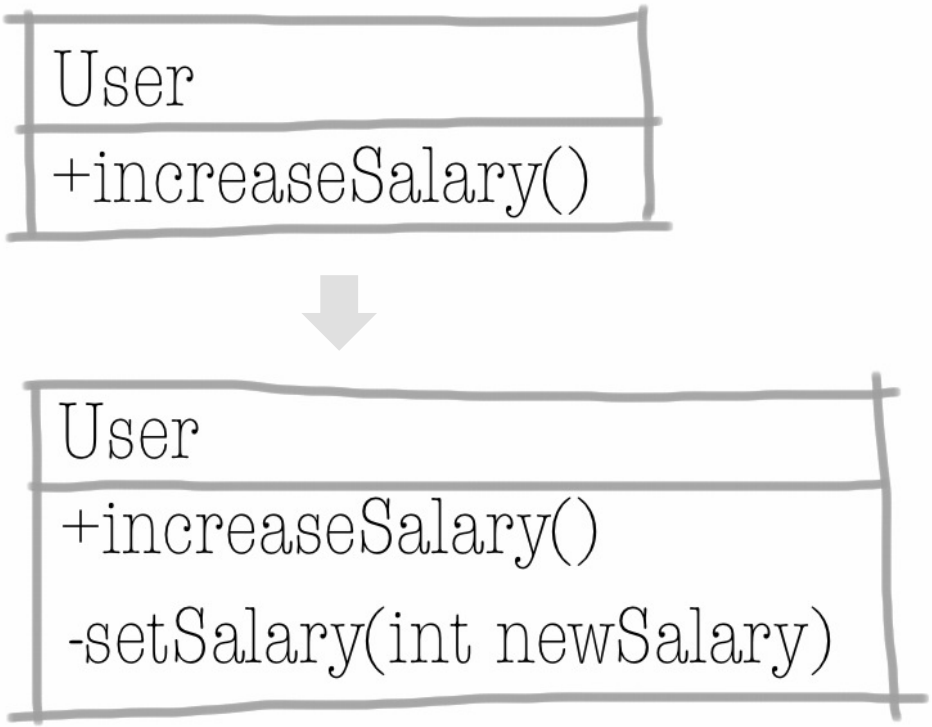
Обоснование через качество

Не уверен, что делаю в коде
так, потому что надо…

…или потому что так
написано в книжке

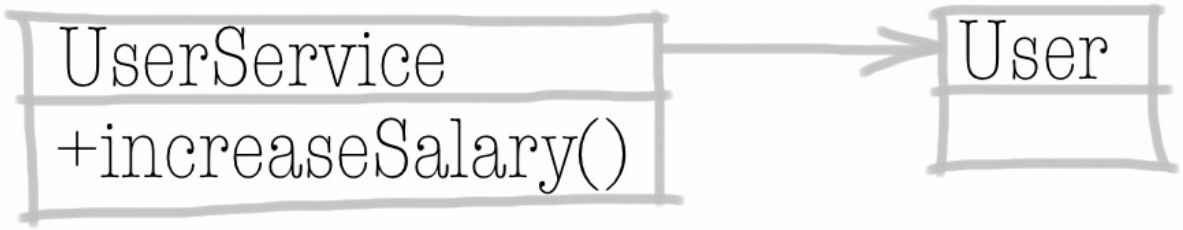
Кейсы рефакторингов

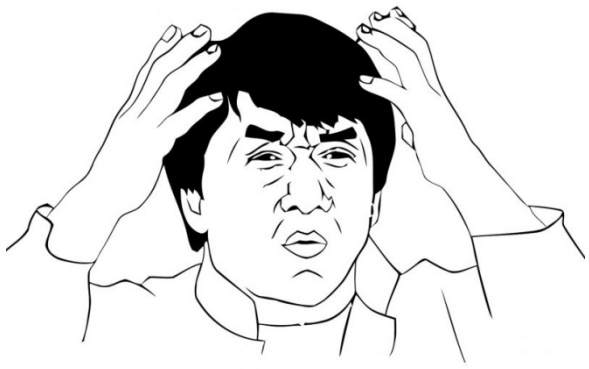
// extract method



Кейсы рефакторингов

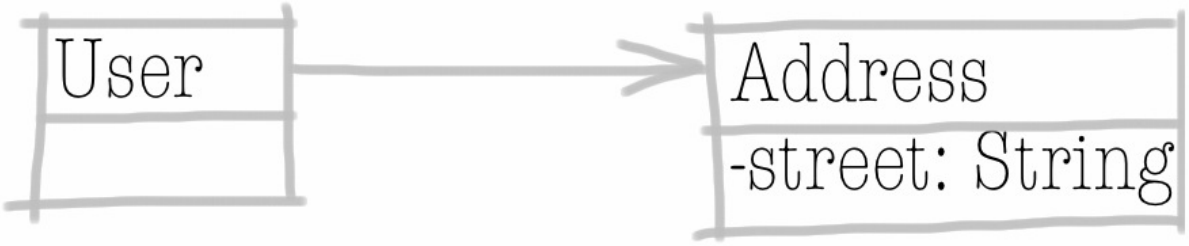
// move method





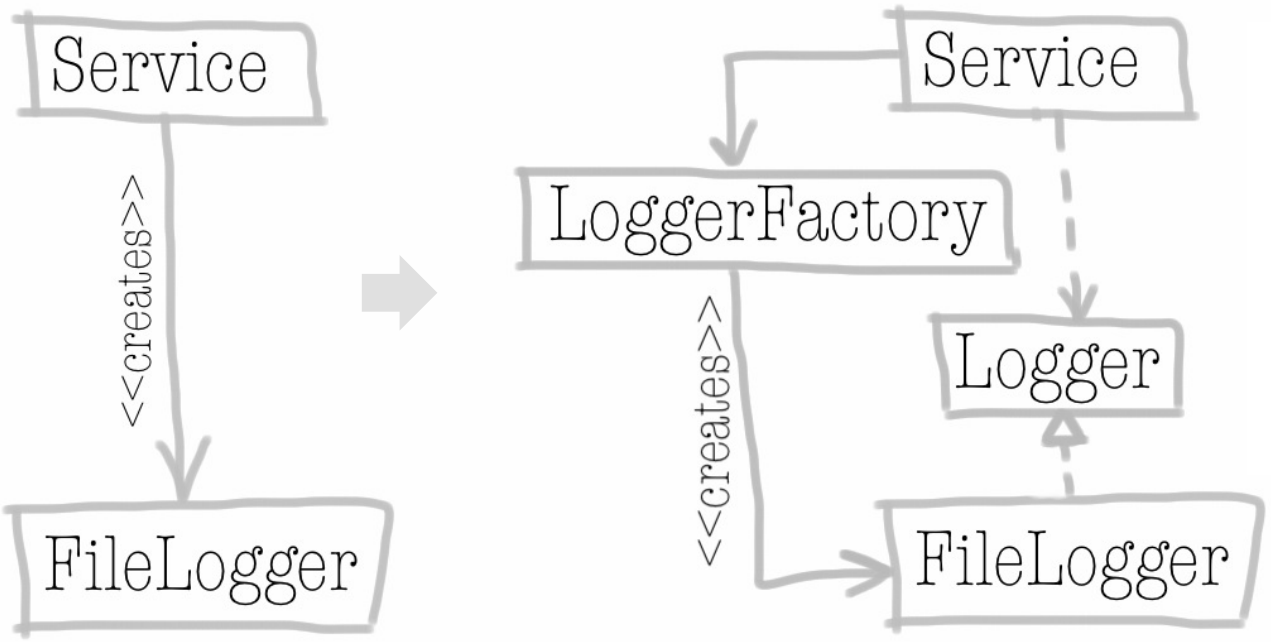
Кейсы рефакторингов

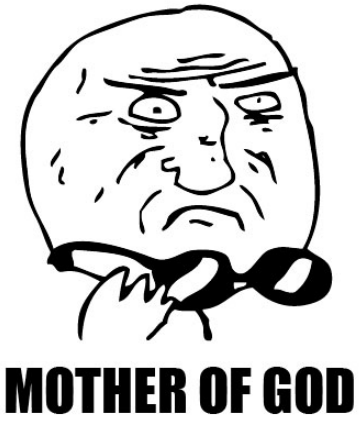
// extract class



Кейсы рефакторингов

// creator pattern → factory pattern

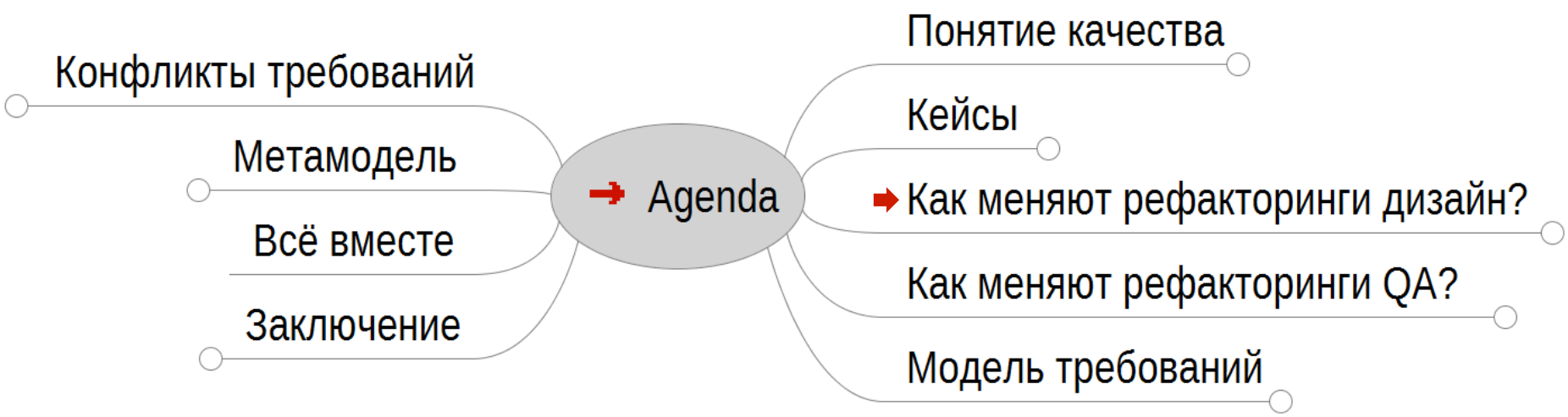




Кейсы участников

- Напишите на листочке последнюю сложную или нерешенную проблему
- Последний holy war
- Проблема любая, но желательно в контексте дизайна / рефакторинга
- В конце Вы попробуем решить её

Сценарий


Конфликты требований
Метамодель
Всё вместе
Заключение
Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований

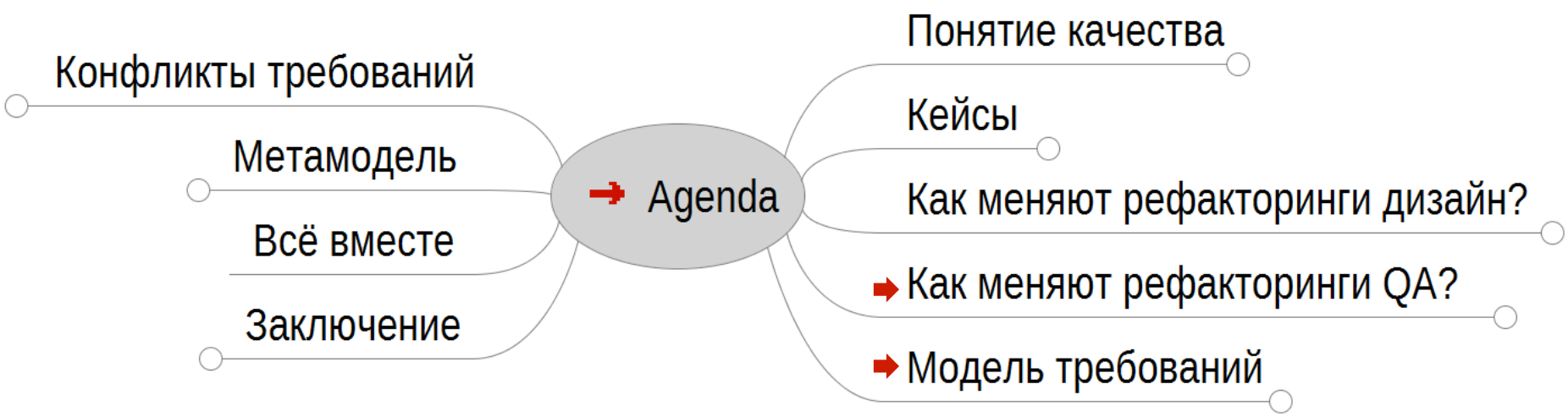
Задача рефакторинга

Привести дизайн
к желаемым характеристикам

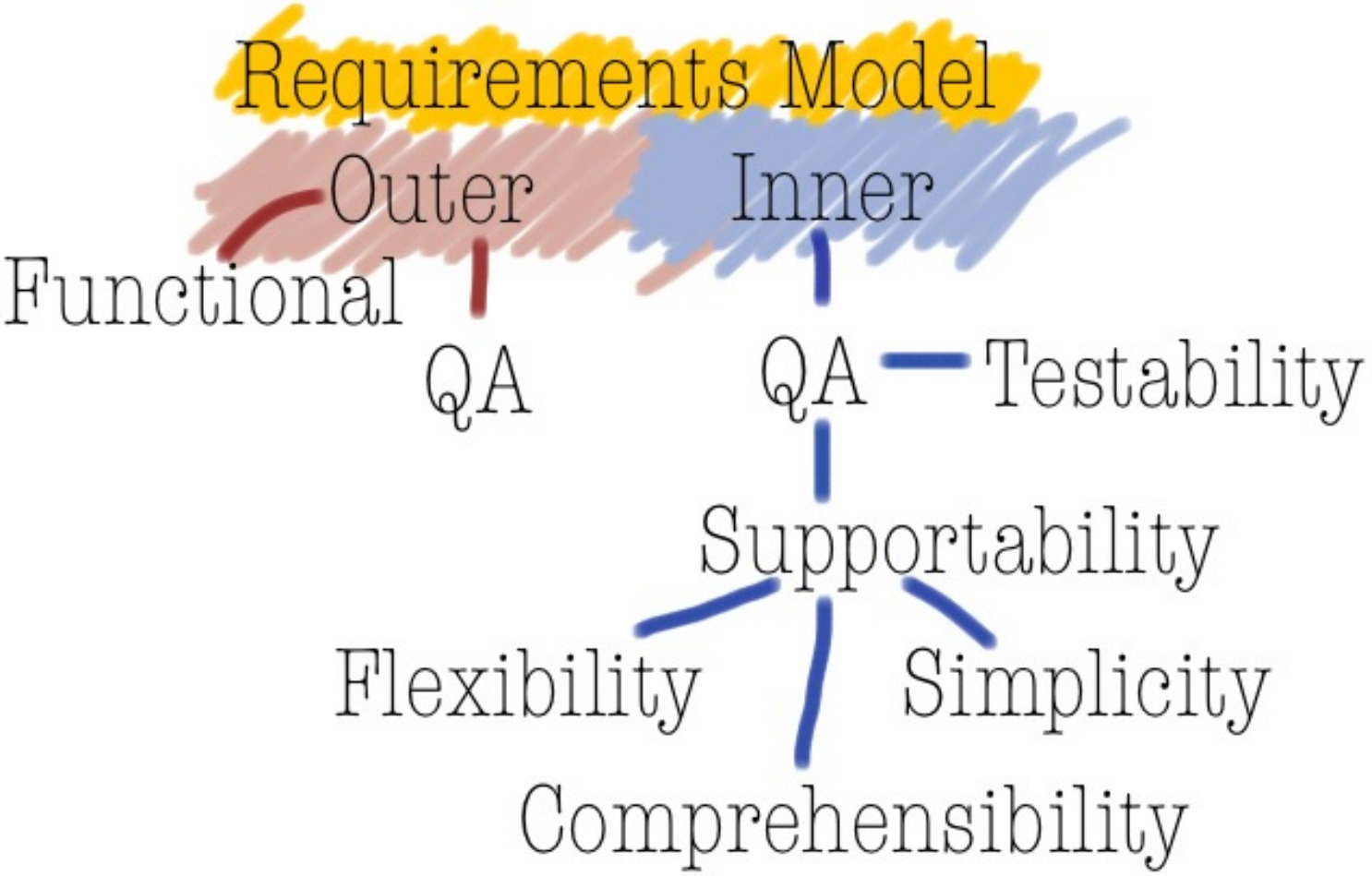
Какие характеристики
желаемые?*

*→быть фабрике или
создателю?

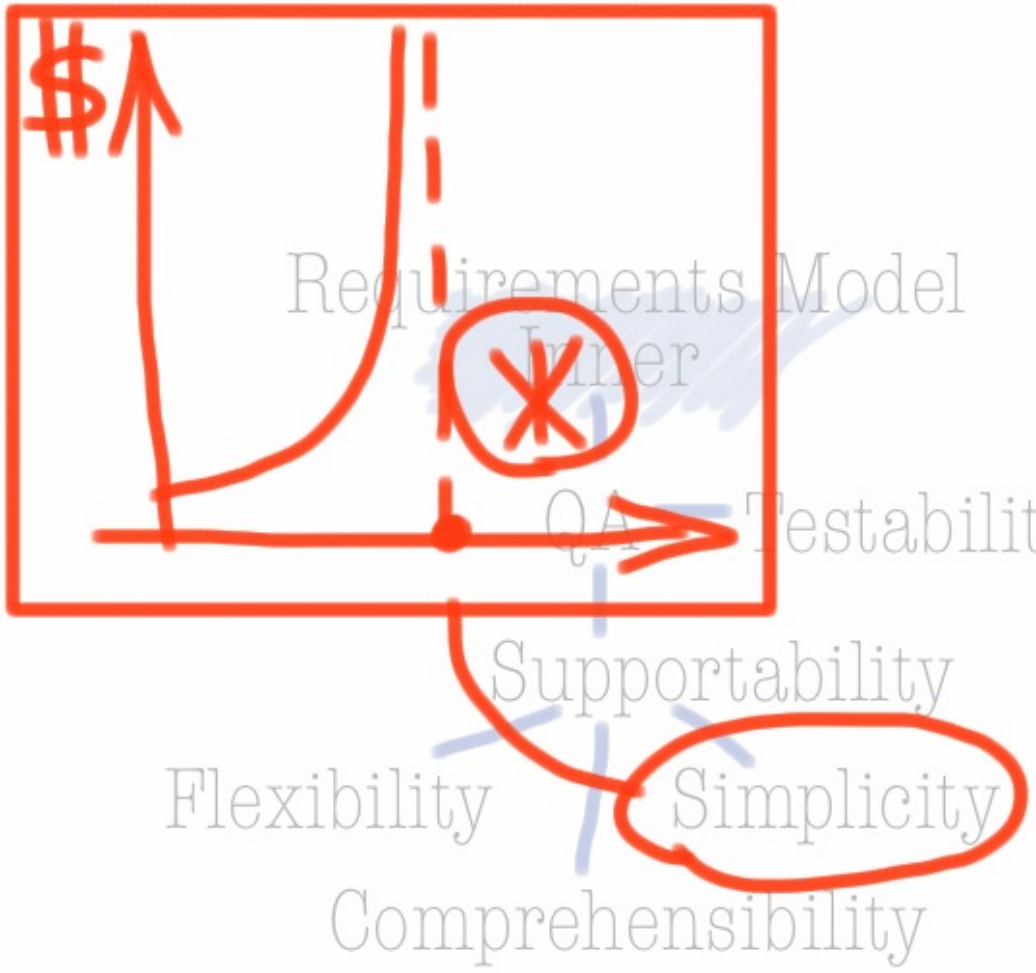
Сценарий


Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований
Конфликты требований
Метамодель
Всё вместе
Заключение

Место QA в требованиях

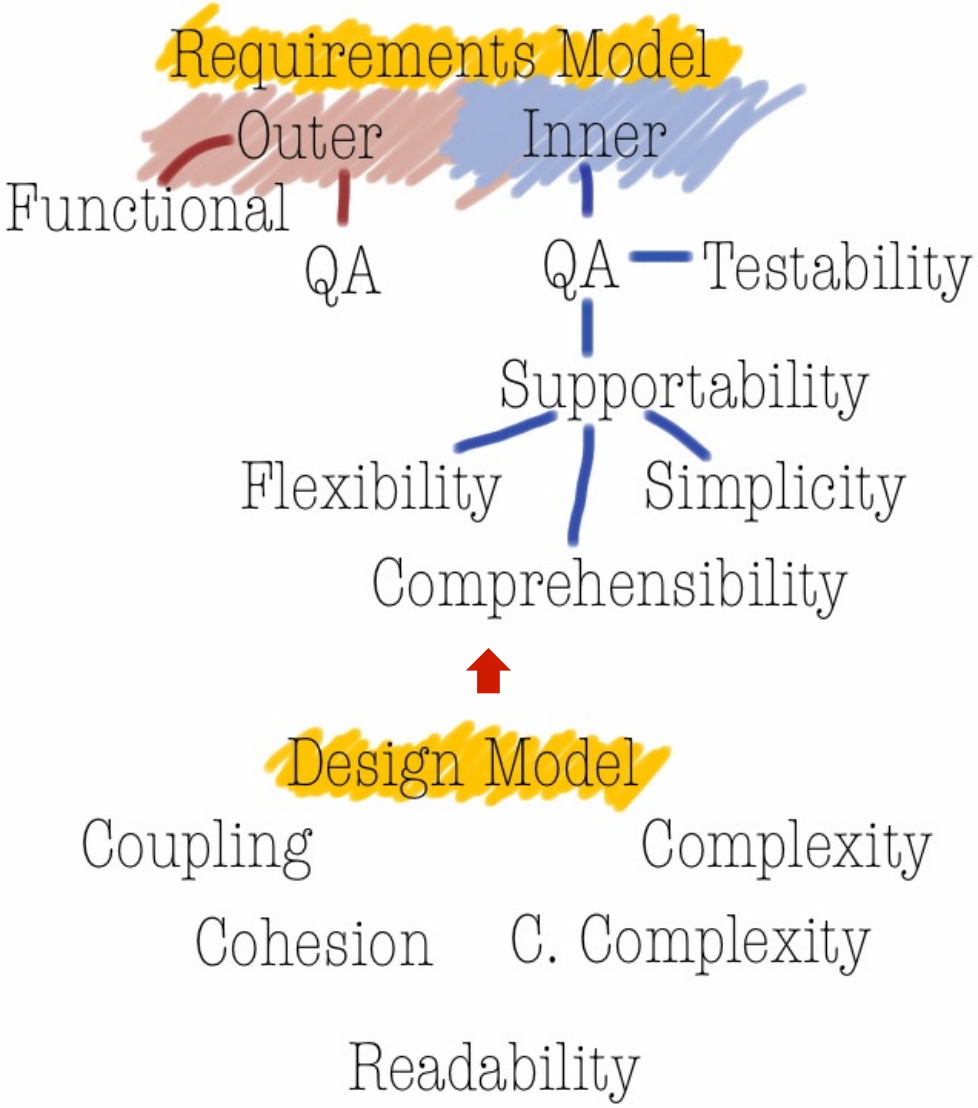


Важность простоты



Простота vs Понятность*

Влияние D.M. на QA



Дизайн должен реализовать требования

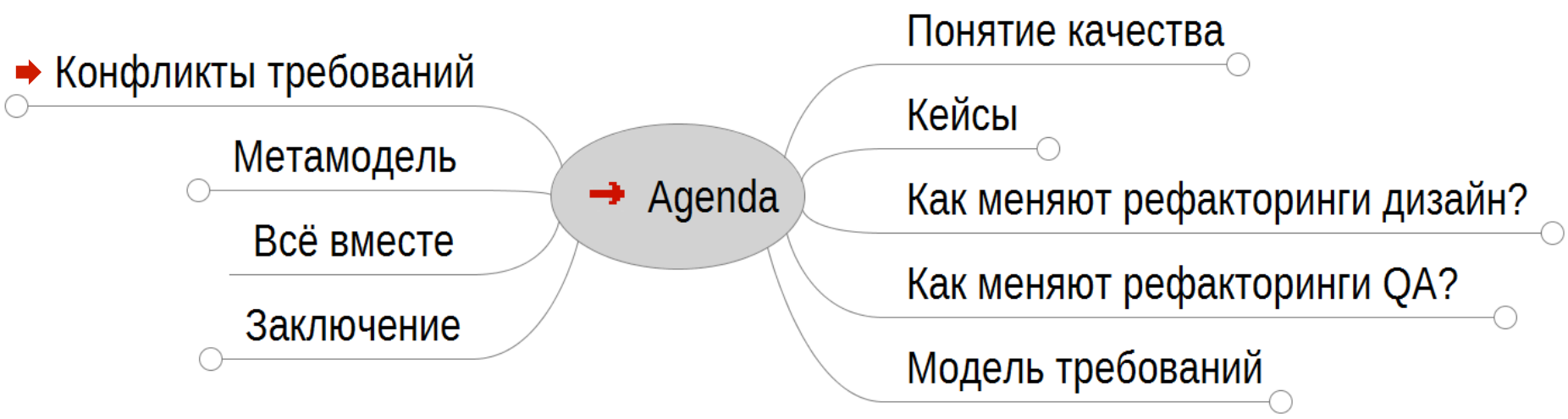
Рефакторинги и QA

Меняя дизайн, меняем и QA

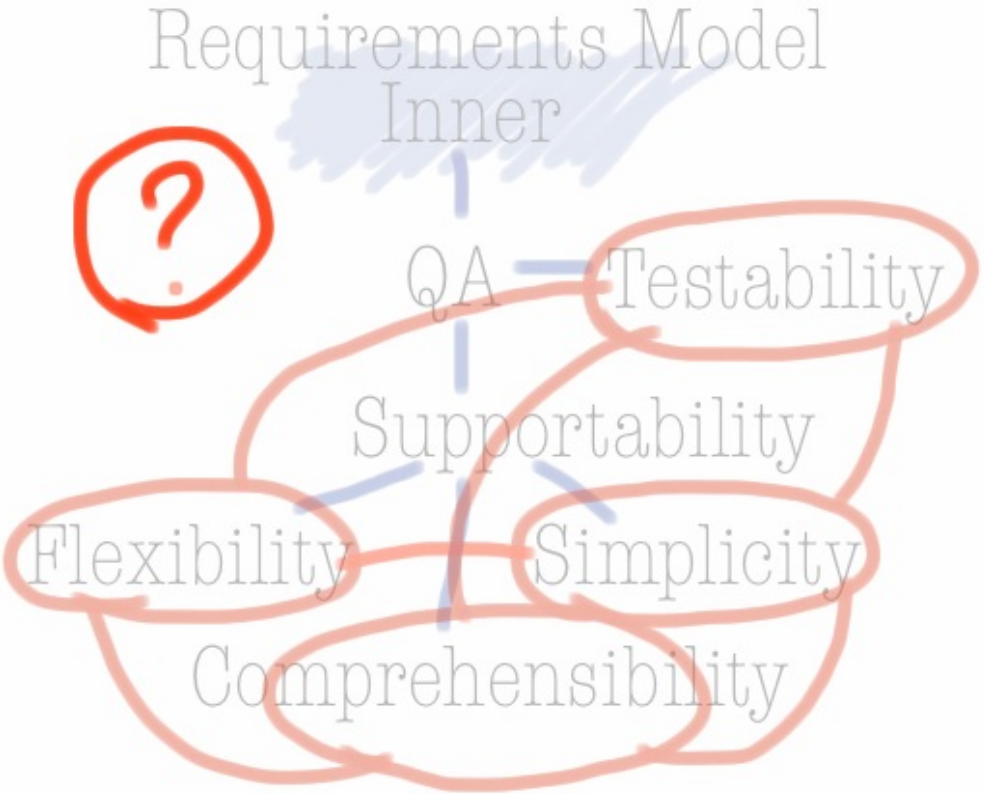
Рефакторинг	Гибкость	Простота	Понимаемость	Тестабельность
extract method				
extract class				
move method				
introduce factory				

↑ или ↓ или ?

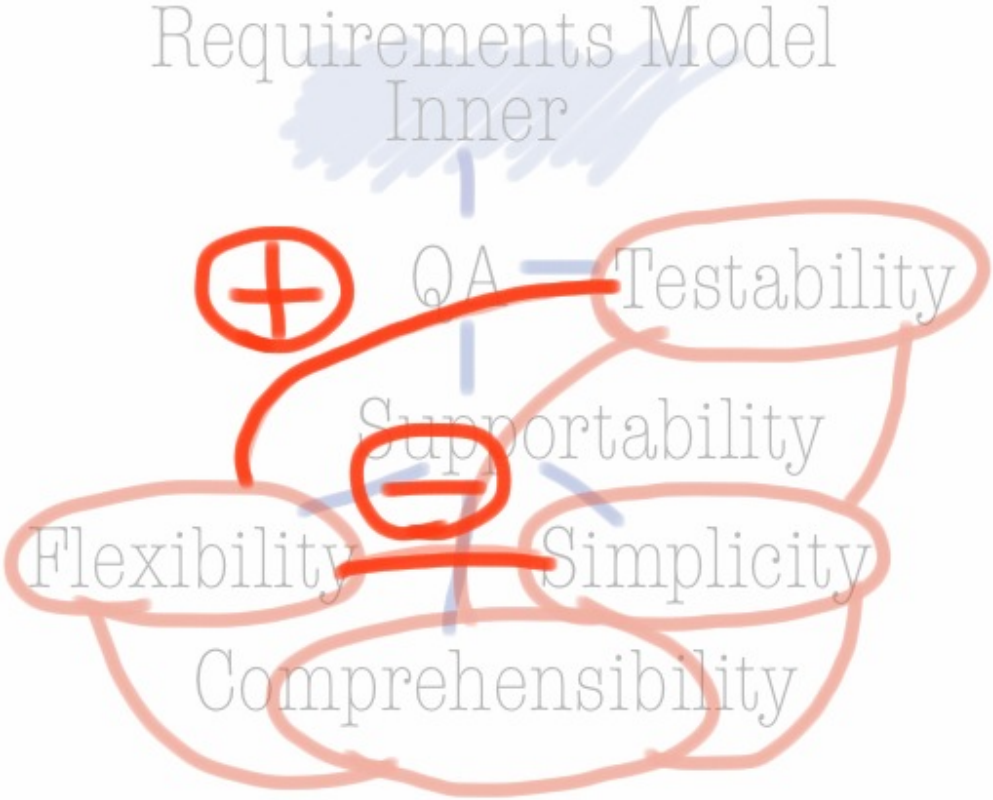
Сценарий


Конфликты требований
Метамодель
Всё вместе
Заключение
Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований

Корреляции требований



Корреляции требований

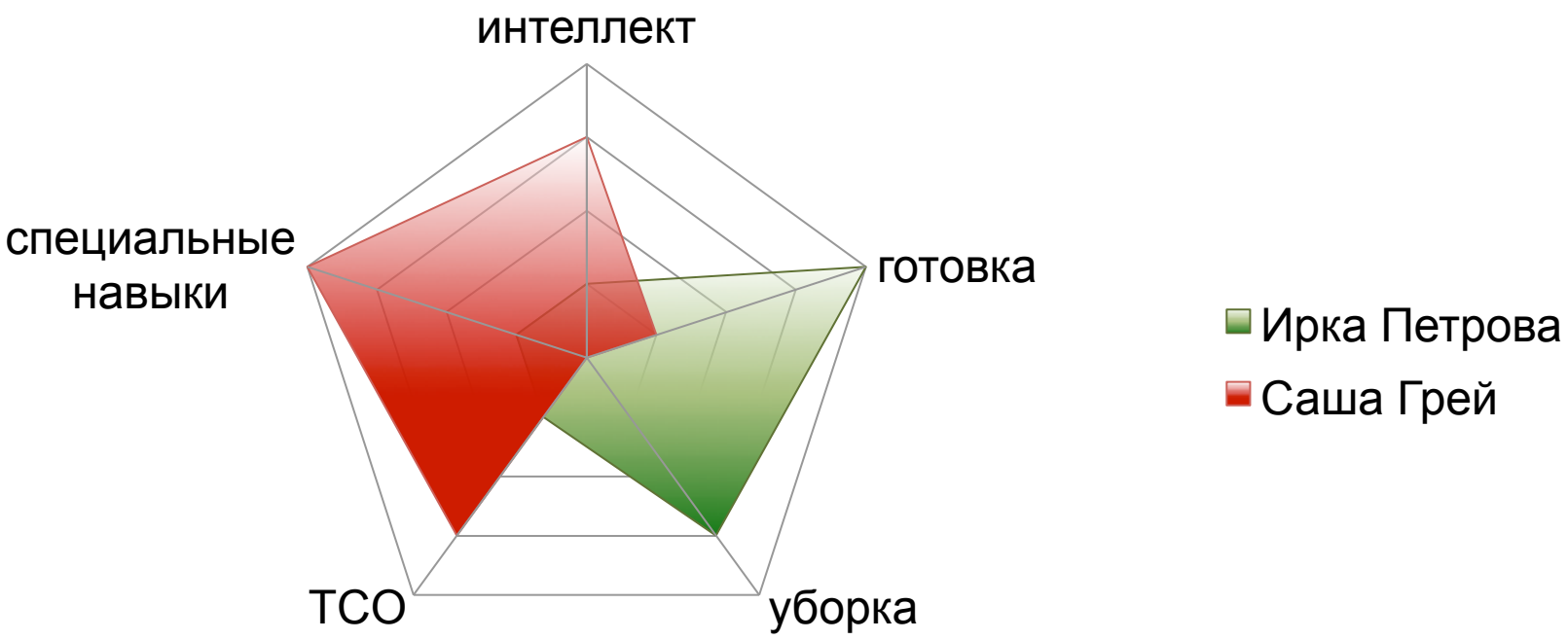


Корреляции требований

Требования зачастую конфликтуют.
Дизайн – это компромисс.*

*За все приходится платить

Конфликты требований



Design is a tradeoff

Factory Pattern	
+	−

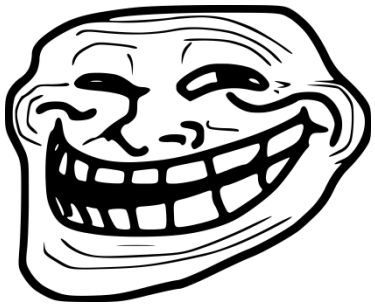
Cache Pattern	
+	−

Design is a tradeoff

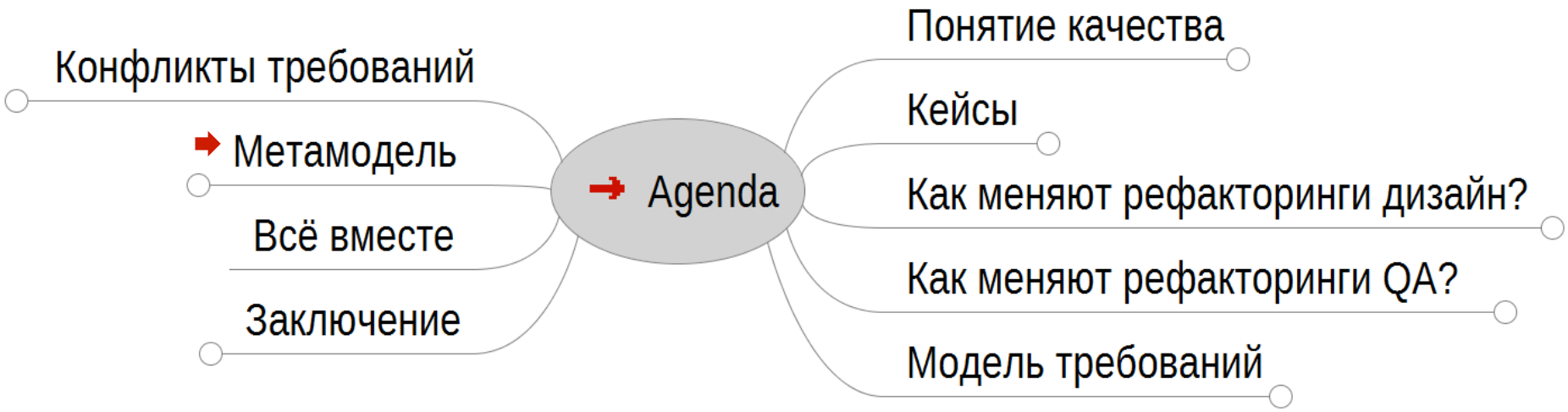
Нет «хорошего» и «плохого» дизайна.
Есть подходящий и неподходящий.

Так быть или не быть?

Factory
Pattern



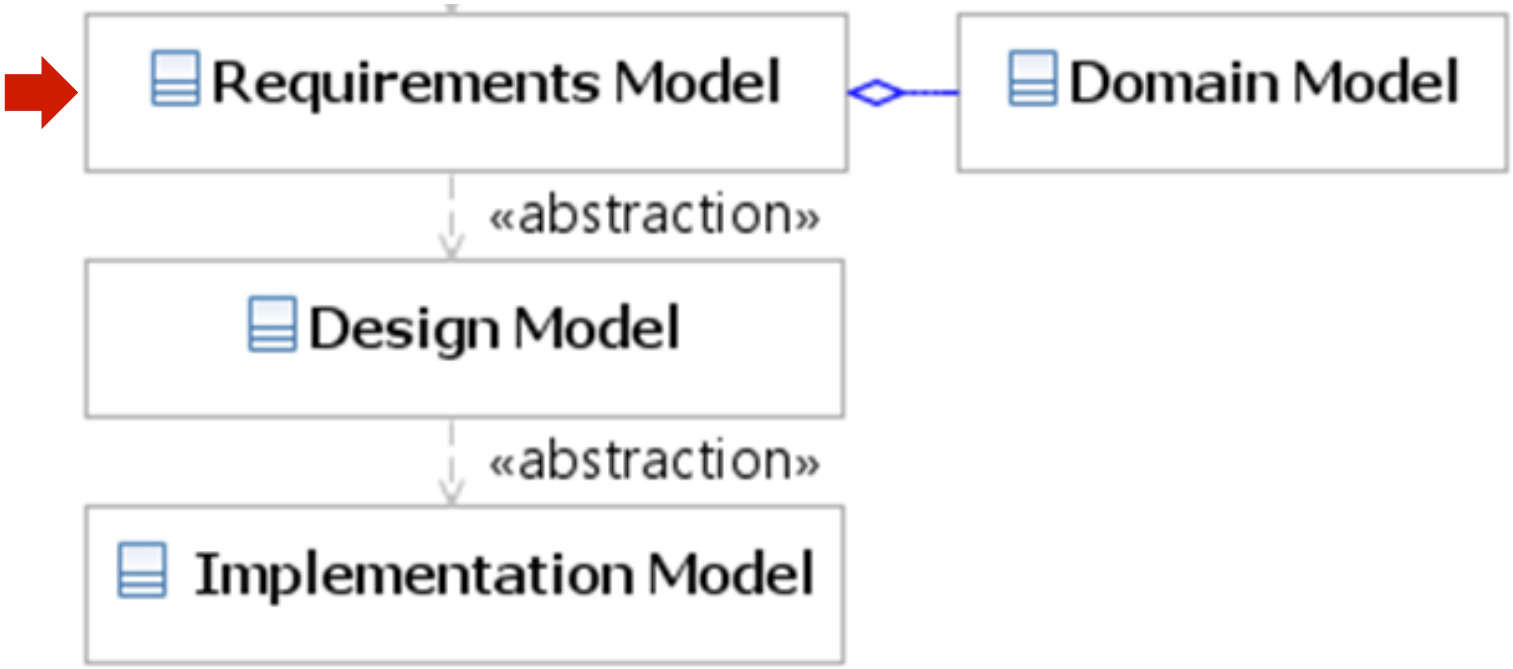
Сценарий


Конфликты требований
Метамодель
Всё вместе
Заключение
Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований

Минутка матана

<матан>
    <a>Вторая теорема Гёделя о неполноте
</матан>

Метамодель дизайна



Дизайн через требования

Функционал вариабелен (ось вариативности)

Функционал стабилен

Гибкость

Простота

Инкапсулируем ответственность в класс

Инлайнируем ответственность

Полнота требований

Функционал вариабелен (ось вариативности)

?

Функционал стабилен

Метамодель требований

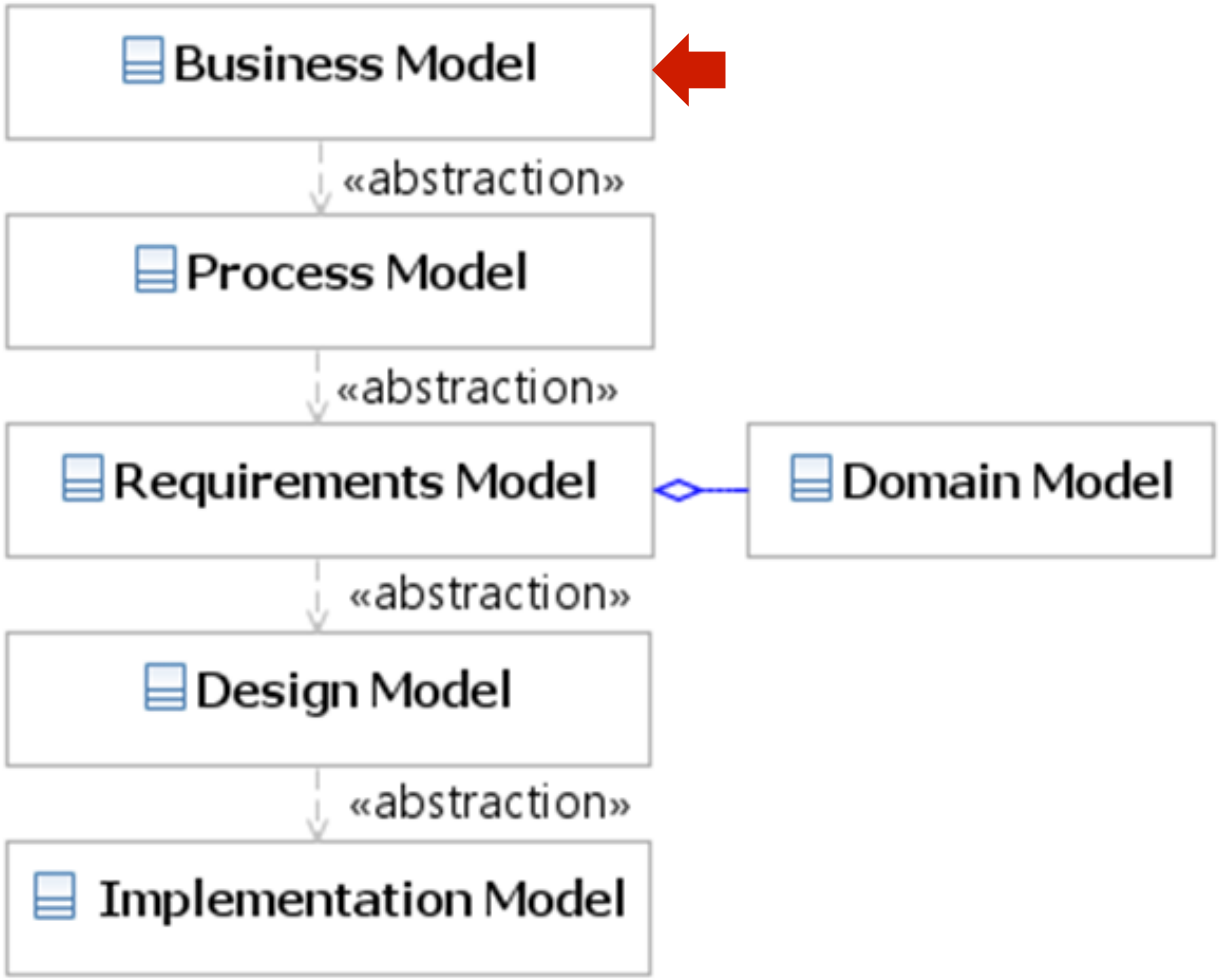
BDUF

YAGNI

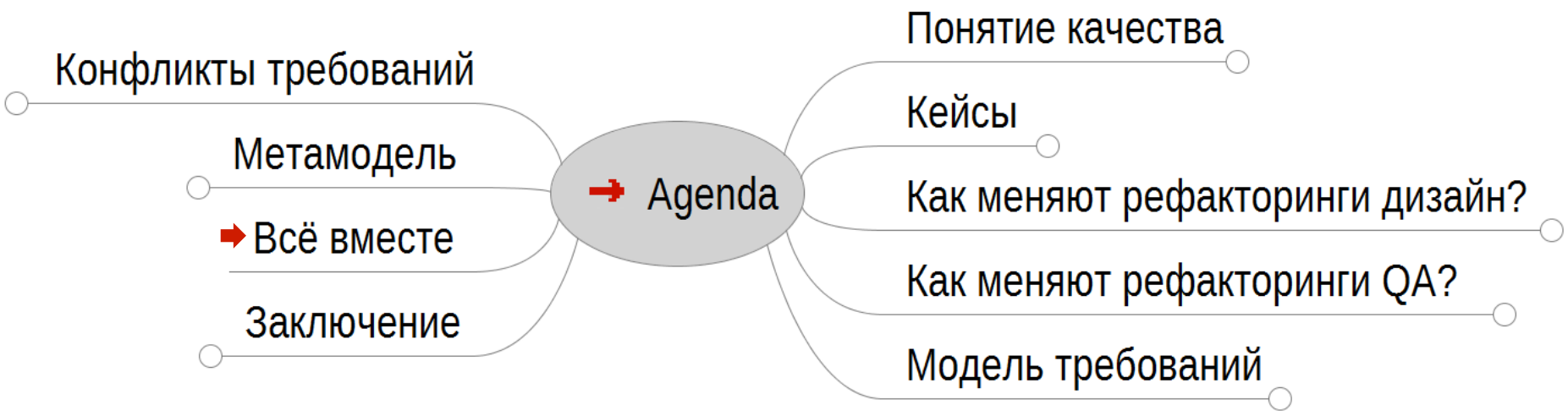
Метамодель требований

BDUF ? YAGNI

Метамодель процесса



Сценарий


Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований
Конфликты требований
Метамодель
Всё вместе
Заключение

Обоснованный дизайн

?

BDUF ? YAGNI

Функционал вариабелен (ось вариативности)

? Функционал стабилен

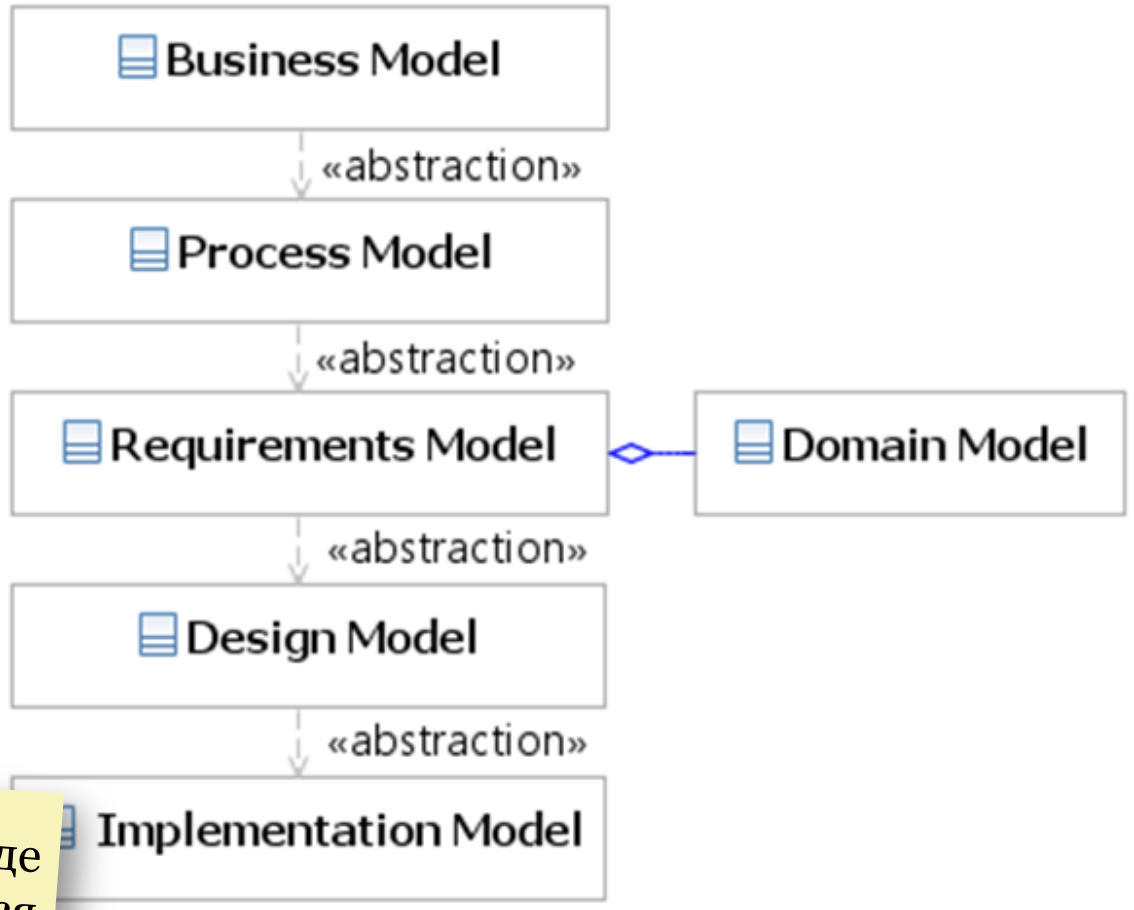
Гибкость

Простота

Инкапсулируем ответственность в класс

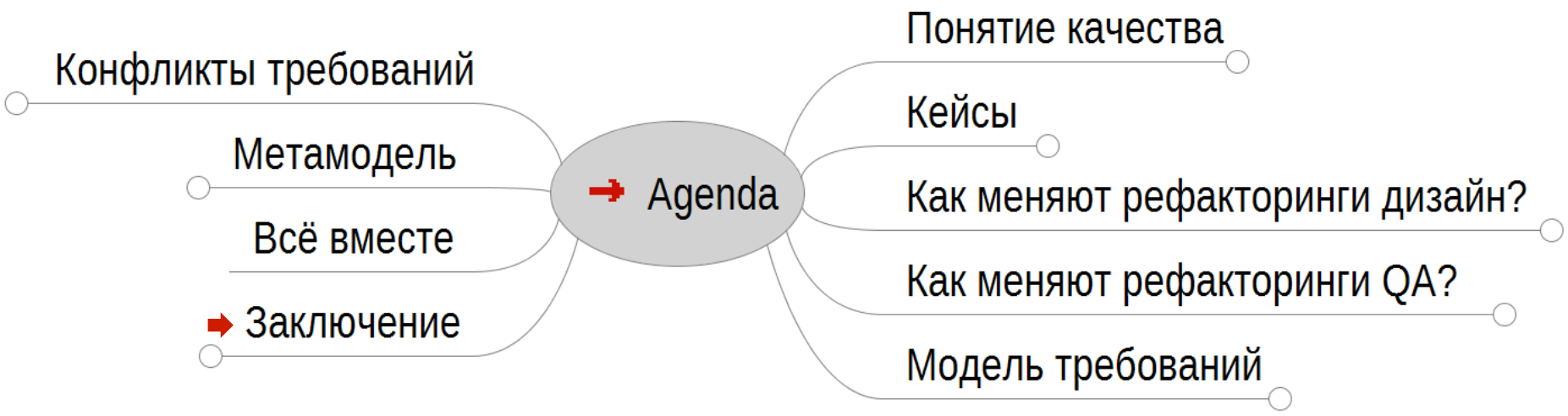
Инлайнируем ответственность

Обоснованный дизайн



Наличие `java interface` в коде
в итоге обосновывается
бизнес-моделью компании

Сценарий


Конфликты требований
Метамодель
Всё вместе
Заключение
Agenda
Понятие качества
Кейсы
Как меняют рефакторинги дизайн?
Как меняют рефакторинги QA?
Модель требований

Отлить в граните

- Рефакторинг – направленное обоснованное изменение дизайна.
- Дизайн – это компромисс. За все нужно платить.
- Необходимо выявить конфликт ожиданий.
- Для принятия решения следует подняться выше на уровень абстракции.
- Решения локальны и специфичны.

Персональный кейс

Получилось ли решить Ваш кейс?*

*Если нет, продолжим за

Контакты

Евгений Кривошеев, ekrivosheyev@scrumtrek.ru
Никита Филиппов, nfilippov@scrumtrek.ru
Асхат Уразбаев, askhat@scrumtrek.ru

«Тяжело в учении – легко в бою»

SkillTrek – это дистанционный центр компетенций, где специалисты получают востребованные на рынке знания и навыки в условиях реальных проектов с выбором удобной им загрузки