



Особенности работы на больших проектах

Александр Чаусов, iOS Разработчик

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Полезные инструменты

Ответственность

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Состав команды

Состав команды

- Разработчики
- PM
- QA
- Дизайнеры
- backend
- Заказчик
- Аналитики

Ведение документов

Приготовьтесь, что различной документации будет много

Ведение документов

- Корпоративного уровня (приход/уход)
- Уровня отдела (iOS/Android)
- Уровня проекта (ТЗ, техдок, техдолг, проектирование)

Планирование и оценка

- Умение точно оценить задачу - важный навык
- Тщательное проектирование
- Представление задачи в виде совокупности подзадач
- Самое главное: привести мысли в своей голове в порядок
- Большая роль опыта

Системы управления проектами

- Ведение систем управления проектами (Jira, Redmine)
- Синхронизация команды
- Слежение за ходом проекта

Команда делает тебя лучше

Работая в команде, ты
перенимаешь опыт каждого из ее
участников

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Поддержка инструментов

Работа с зависимостями

Хороший разработчик - ленивый
разработчик

Менеджер зависимостей

- Упрощают и стандартизируют процесс выборки стороннего кода и включения его в ваш проект
- Облегчают обновление сторонних библиотек
- Выбирают соответствующие и совместимые версии зависимостей

Менеджер зависимостей

- Cocoapods
- Carthage

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Поддержка инструментов

Взаимодействие в команде

Работа с зависимостями

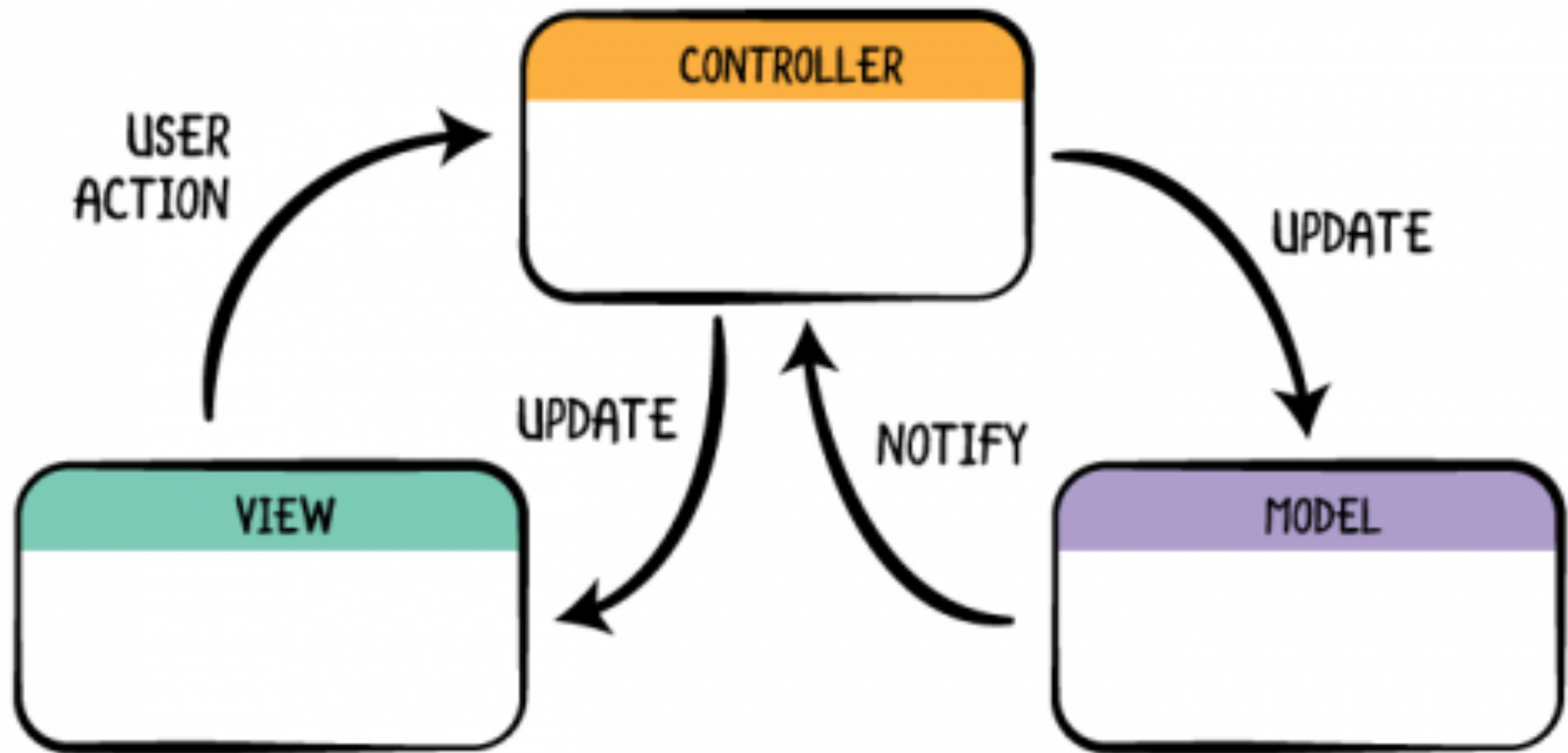
Архитектура

Чистый код

Полезные инструменты

Ответственность

MVC

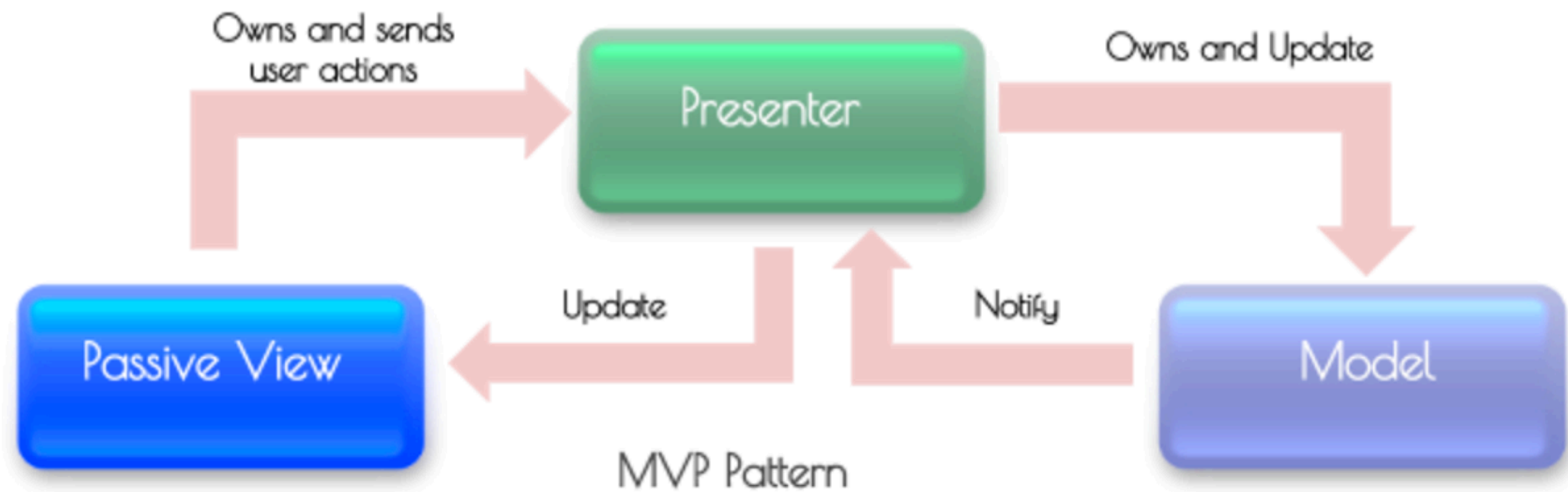


Минусы MVC

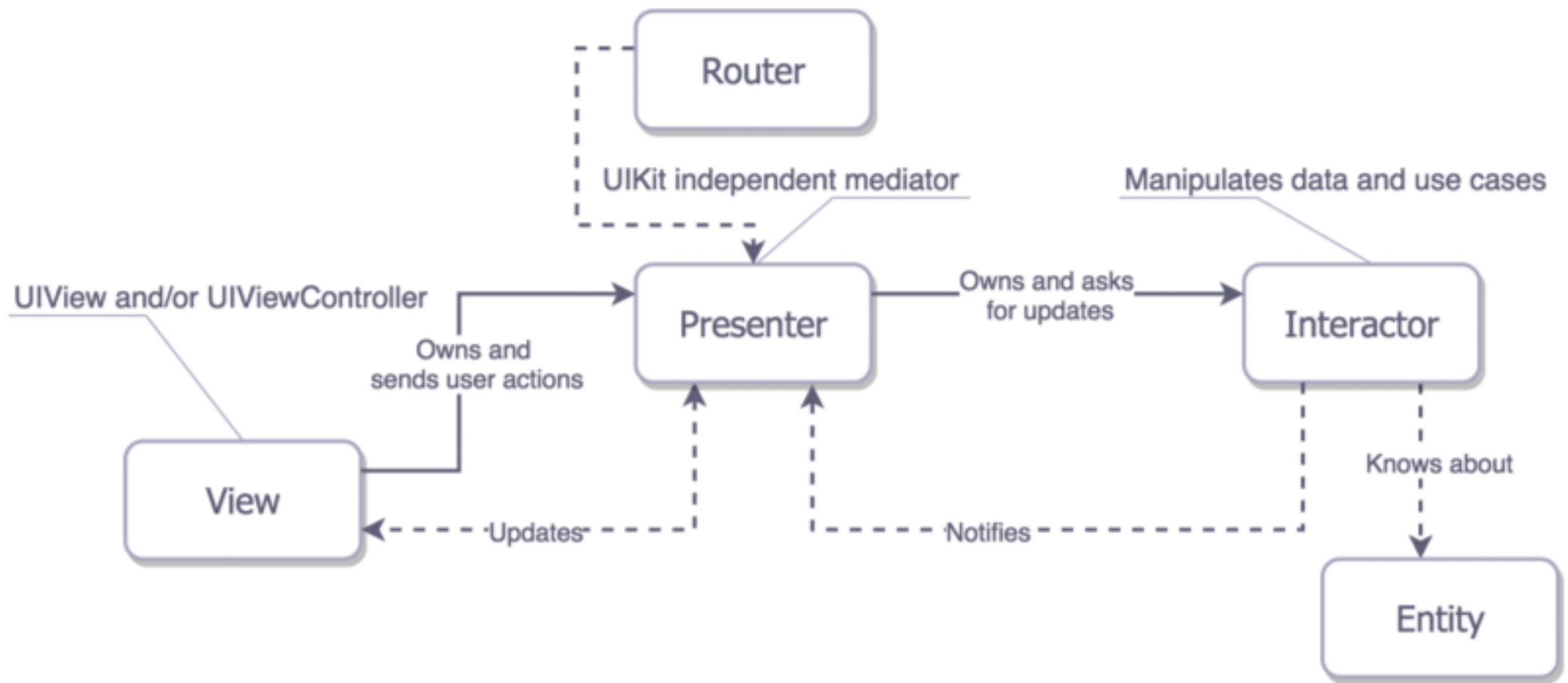
Минусы MVC

- Massive-View-Controller
- Переплетение между собой сущностей View и Controller - ViewController
- Крайне трудоемкая поддержка кода
- Controller несет множество ответственностей

MVP

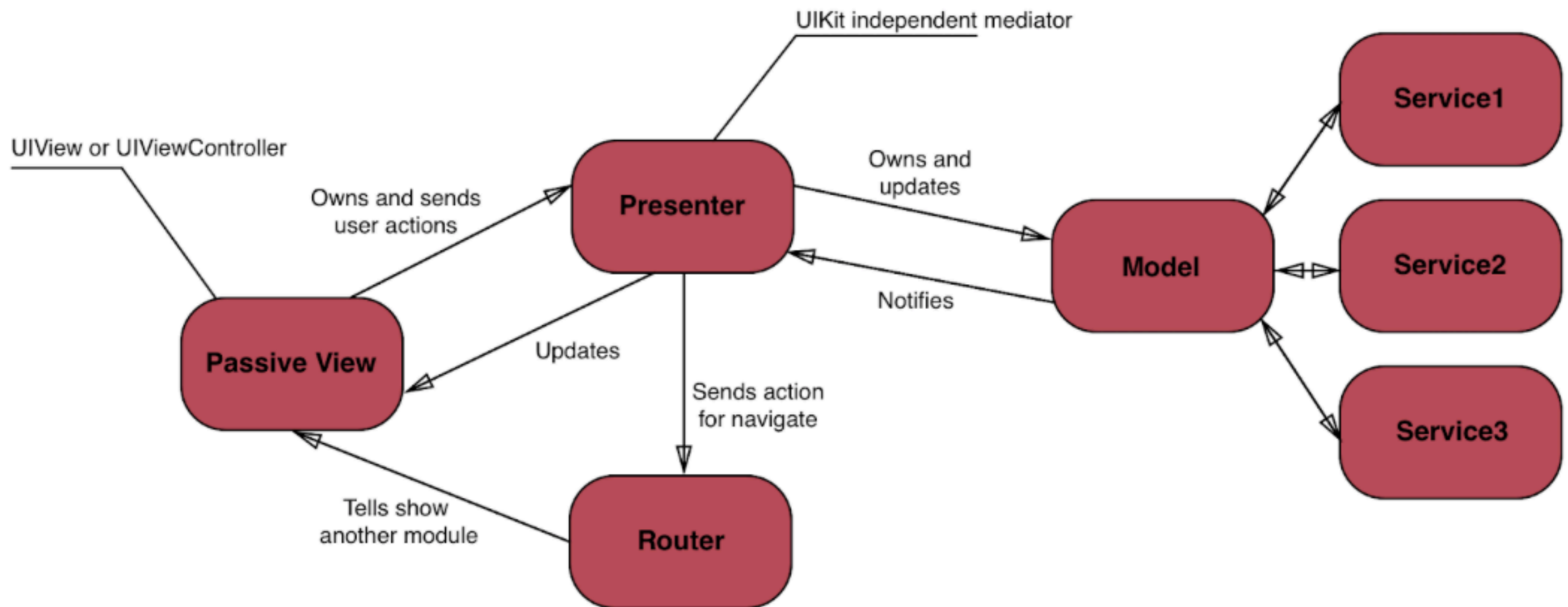


VIPER





Surf MVP 2.0



Проектирование

- Избавьтесь от каши в голове
- Имейте четкое представление о разбивке задачи и общем пути ее решения
- Следуйте плану
- Не бойтесь задавать вопросы
- Думайте о коллегах

Проектирование

Хорошее проектирование
поможет предотвратить будущие
проблемы и дать более точную
оценку

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Полезные инструменты

Ответственность

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Полезные инструменты

Ответственность

Главная проблема начинающего разработчика

Не установился свой внутренний
code style

Code style

- Важно, чтобы все члены команды оформляли код одинаково
- Корпоративный code style, проектный code style
- Переход разработчиков между проектами
- Переход участников проекта между его функциональными частями
- Проведение code review

Code review

- Review на уровне проектирования приложения
- Review на уровне проектирования спринта
- Review на уровне задачи
- Использование PR (pull request)

Pull request

- Улучшается качество кода
- Находятся «глупые» ошибки (опечатки) в реализации
- Повышается степень совместного владения кодом
- Код приводится к единому стилю написания
- Хорошо подходит для обучения «новичков», быстро набирается навык, происходит выравнивание опыта, обмен знаниями.

Pull request

- В одном PR'е - одно законченная фича/задача/мысль
- Подробный комментарий: что сделал, какие особенности, возможные недочеты, как сделать что-то лучше, как тестировать
- Создал PR - просмотри его сам

SOLID

- Принцип единственной ответственности (Single responsibility)
- Принцип открытости/закрытости (Open-closed)
- Принцип подстановки Барбары Лисков (Liskov substitution)
- Принцип разделения интерфейса (Interface segregation)
- Принцип инверсии зависимостей (Dependency Inversion)

Single responsibility

- «На каждый объект должна быть возложена одна единственная обязанность»
- Один из наиболее простых в понимании, но в то же время наиболее часто нарушающийся
- Признаки: громадные классы, длинные методы, длинные названия методов

Взаимодействие в команде

Работа с зависимостями

Архитектура

Чистый код

Полезные инструменты

Ответственность

Работа с зависимостями

Архитектура

Чистый код

**Полезные
инструменты**

Ответственность

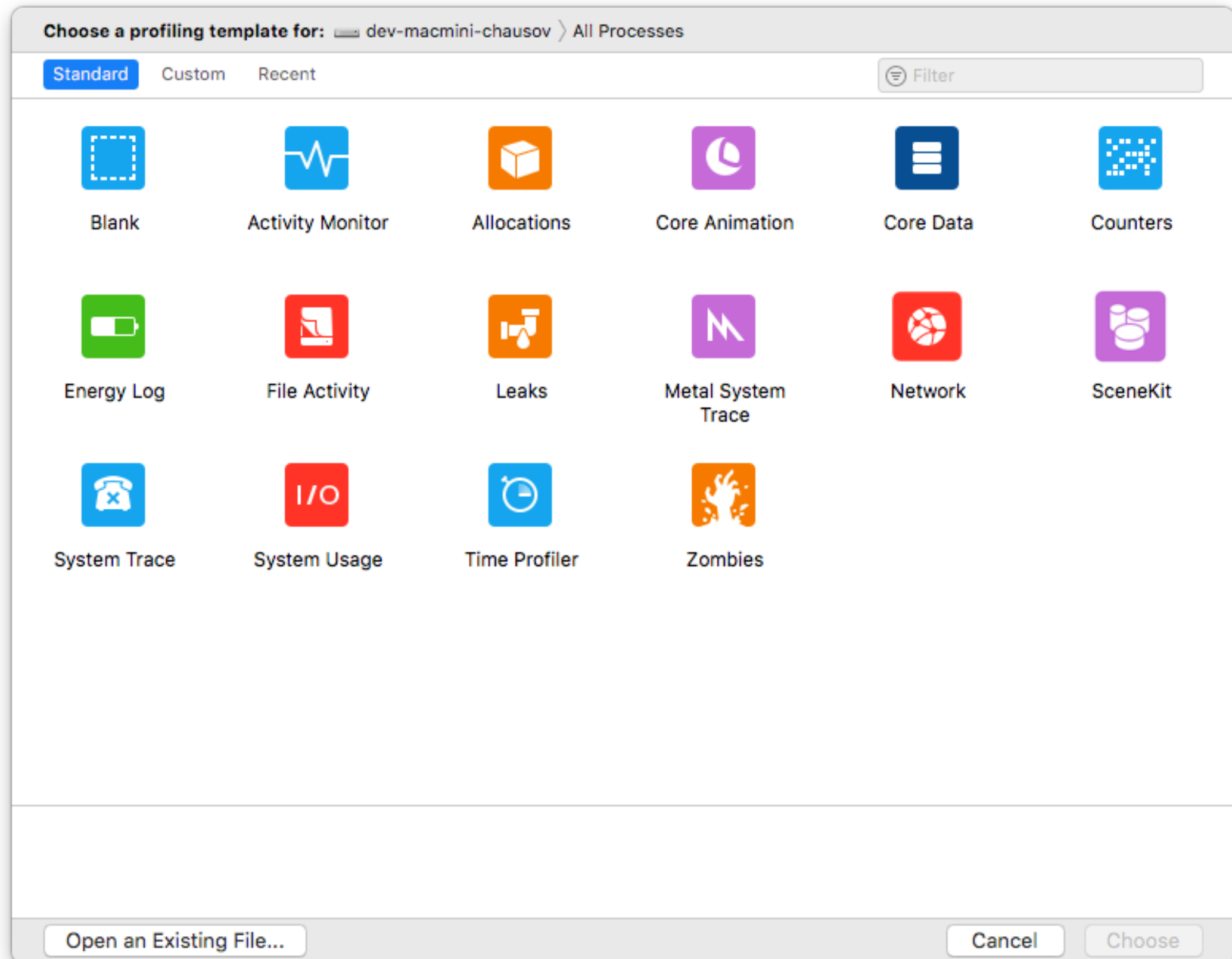
Полезные инструменты

- Instruments
- Мессенджеры
- Системы управления проектами
- Дизайн
- Network-flow
- Работа со сборками

Полезные инструменты

- Сервисы для сбора информации о сбоях
- GIT
- Аналитика
- CI
- Others

Instruments



Мессенджеры

- HipChat
- Stride
- Slack

Системы управления проектами

- Jira
- Redmine
- Trello
- Projects в github

Дизайн

- Zeplin
- Figma

Network-flow

- Charles
- Postman

Работа со сборками

- Test flight
- Fabric
- Hockey app

Сервисы для сбора информации о сбоях

- Crashlitics
- Firebase Crash Reporting

GIT

- SourceTree
- GitHub Desktop
- GitHub
- Bitbucket
- GitLab

Аналитика

- Google Analytics
- Firebase
- Flurry
- YandexAppMetrica
- Localytics

CI (continuius integration)

- Jenkins
- Travis CI
- SwiftLint - автоматическая проверка Swift кода, упрощает проверку кода на соответствие code style
- Codebeat - анализирует сложность кода, дублирование и ставит оценка вашему проекту
- Codescov - анализирует покрытие кода тестами

Others

- Alcatraz
- Sublime
- Joxi
- Quiver
- Pusher

Generamba

- Кодогенератор для iOS разработки
- Простота добавления и изменения шаблонов
- Понятный синтаксис (шаблонный движок liquid)

SwiftGen

- Автогенератор кода для ресурсов проекта
- Изображения, цвета, шрифты, локализованные строки

Работа с зависимостями

Архитектура

Чистый код

**Полезные
инструменты**

Ответственность

Архитектура

Чистый код

Полезные инструменты

Ответственность

Ответственность разработчика

- Ответственность растёт с профессиональным ростом
- Разработчик должен отвечать за написанный код
- И нести ответственность за approve на PR'е

Итог

- Думай
- Пиши просто и понятно
- Учись проектировать
- Задавай вопросы
- Не бойся говорить «я не знаю»
- Учись оценивать
- И еще раз думать
- Совершенствуйся :)

Архитектура

Чистый код

Полезные инструменты

Ответственность

Материалы

- Carthage
<https://habr.com/post/270805/>
<https://github.com/Carthage/Carthage>
<https://www.raywenderlich.com/165660/carthage-tutorial-getting-started-2>
- CocoaPods
<https://cocoapods.org>
<https://www.raywenderlich.com/156971/cocoapods-tutorial-swift-getting-started>
- Динамические и статические библиотеки
<https://www.ca.com/en/blog-developers/dynamic-versus-static-framework-in-ios.html>

Материалы

- Архитектурные iOS паттерны (блог Badoo)
<https://habr.com/company/badoo/blog/281162/>
- VIPER
<https://github.com/strongself/The-Book-of-VIPER>
- Review кода
<https://habr.com/post/142564/>
<https://medium.com/@same7mabrouk/the-checklist-of-my-code-review-18cc6f6fb5b3> (+ code style)
<https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>

Материалы

- Шпаргалка по SOLID-принципам
<https://habr.com/post/208442/>
- Обзор систем управления проектами
<https://habr.com/post/173633/>
- Generamba
<https://github.com/rambler-digital-solutions/Generamba>
<https://habr.com/company/rambler-co/blog/276275/>
- SwiftGen
<https://github.com/SwiftGen/SwiftGen>

Материалы

- SwiftLint
<https://github.com/realm/SwiftLint>
- SwiftLint - чистота и порядок в iOS проекте
<https://habr.com/company/tinkoff/blog/317892/>
- CI for iPhone apps
<https://medium.com/ios-os-x-development/continuous-integration-for-iphone-apps-7b64158987c9>
- Surf Swift codestyle
<https://github.com/surfstudio/SwiftCodestyle>

