

Антон Кузьмин

Разработка фронтенд- приложений



Санкт-Петербург • Москва • Минск

2025

КРАТКОЕ СОДЕРЖАНИЕ

Введение	17
От издательства	23
Глава 1. Фронтенд-разработка	24
Глава 2. С чем имеет дело фронтенд-разработчик	35
Глава 3. Основы проектирования и разработки	96
Глава 4. Состояние	135
Глава 5. UI	206
Глава 6. Управление рендерингом компонентов с помощью изменяющегося URL	325
Глава 7. Файлы приложения	364
Глава 8. Разработка фронтенд-приложения	370
Глава 9. Совместная работа над приложением	467
Глоссарий	484

ОГЛАВЛЕНИЕ

Введение	17
Что вам даст эта книга	17
Почему стоит прочесть эту книгу, если вы планируете заняться фронтенд-разработкой.....	17
Для кого написана эта книга.....	18
Кратко о главах книги.....	18
Последовательность чтения глав	19
TypeScript.....	20
Как читать эту книгу	20
Почему упражнения и практические задания отделены от изучаемого материала.....	22
От издательства	23
Глава 1. Фронтенд-разработка	24
1.1. С чего начиналась фронтенд-разработка.....	24
1.1.1. Осознание способов решения проблем	24
1.1.2. Приложения на основе страниц.....	24
1.1.3. Приложения на основе AJAX и библиотек.....	25
1.1.4. Фронтенд-приложения.....	26
1.1.5. Эпоха искусственного интеллекта и виртуальной реальности	28
1.2. Разработка фронтенд-приложения	30
1.2.1. Окружение разработки	30
1.2.2. Управление зависимостями	31
1.2.3. Язык и кодирование. Трансляция.....	32
1.2.4. Сборка приложения	32
1.2.5. Поиск проблем в коде.....	33
1.2.6. Оформление кода.....	33
1.2.7. Тестирование	33
1.2.8. Контроль версий и обмен кодом.....	33
1.2.9. Код-ревью	33
1.2.10. Заключение	34

Глава 2. С чем имеет дело фронтенд-разработчик	35
2.1. Возможности программ на JavaScript.....	35
2.1.1. С чем и как взаимодействует программа на JS.....	35
2.1.2. Работа с данными.....	36
2.1.3. Платформенная функциональность	38
2.2. Платформы.....	39
2.2.1. Что представляют собой платформы и зачем они нужны.....	39
2.2.2. Уникальность функциональности.....	39
2.2.3. Важные выводы.....	40
2.3. Браузер и его объекты функциональности.....	40
2.3.1. Где платформа, а где JavaScript.....	40
2.3.2. Браузер — платформа с набором программ, предоставляющих API	42
2.3.3. Библиотеки	43
2.3.4. Библиотеки React и ReactDOM	44
2.3.5. Преимущество библиотек.....	44
Упражнения и практические задания.....	45
2.4. Загрузка приложения.....	45
2.4.1. Как осуществляется загрузка	45
2.4.2. Парсинг содержимого строки, помеченной как HTML	45
2.4.3. Прослушивание событий	46
2.4.4. Жизнь приложения	47
2.4.5. Другой взгляд на HTML.....	47
2.5. DOM.....	47
2.5.1. Как все могло бы быть сложным	47
2.5.2. Что такое модель	48
2.5.3. Что такое документ	49
2.5.4. Что такое Document Object Model (DOM).....	50
2.5.5. Свойство document и операции в DOM.....	53
2.5.6. Операции поиска узлов в дереве	53
2.5.7. Обход дерева DOM	54
2.5.8. Создание и добавление узлов в дерево.....	62
2.5.9. Удаление узлов из DOM-дерева.....	66
2.5.10. Изменение атрибутов и свойств узлов DOM-дерева.....	68
2.5.11. Узлы дерева DOM и их свойства	72
Упражнения и практические задания.....	74
2.6. События.....	75
2.6.1. Что такое событие.....	75
2.6.2. Некоторые виды событий и источники	75
2.6.3. Физические действия.....	76
2.6.4. Действия, генерируемые программой	76
2.6.5. Событие — это процесс	76
2.6.6. Работа с событиями в коде	77
2.6.7. Группировка событий по типу	79

2.6.8. Объект события	84
2.6.9. Три стадии события	85
2.6.10. Иная точка зрения на события.....	88
Упражнения и практические задания.....	88
2.7. Асинхронное программирование	89
2.7.1. Инструкции языка программирования.....	89
2.7.2. Кратко и упрощенно о работе браузера в операционной системе	89
2.7.3. Что такое асинхронное программирование	90
2.7.4. Последовательное выполнение программы	90
2.7.5. Event Loop	91
2.7.6. Выполнение асинхронного кода в браузере. Задачи и микрозадачи	93
2.7.7. Главная разница между очередями задач и микрозадач	94
2.7.8. Дополнительно	95
Глава 3. Основы проектирования и разработки	96
3.1. О разработке приложения	96
3.1.1. Введение.....	96
3.1.2. Смена точки зрения на разработку.....	96
3.1.3. Разработка информационных систем.....	97
3.2. Как разрабатываются информационные системы в организациях.....	97
3.2.1. Работающее веб-приложение — это готовый продукт.....	97
3.2.2. С чего начинается то, что мы видим как веб-приложение	99
3.2.3. Требования.....	99
3.2.4. Краткое описание всего процесса разработки	100
3.2.5. Команда	101
3.2.6. За что несет ответственность фронтенд-разработчик.....	102
3.3. Граф. Дерево	103
3.3.1. Введение.....	103
3.3.2. Граф. Вершина. Ребро. Дуга.....	103
3.3.3. Неориентированный и ориентированный графы.....	105
3.3.4. Взвешенный граф.....	108
3.3.5. Компоненты связности.....	109
3.3.6. Маршрут. Цепь. Путь. Цикл	110
3.3.7. Дерево	113
3.3.8. Свойства графов, дающие отличительные особенности.....	116
3.3.9. Моделирование с помощью графов.....	117
3.3.10. Продолжайте изучать графы	118
Упражнения и практические задания.....	118
3.4. Система.....	118
3.4.1. Введение.....	118
3.4.2. Что такое система.....	118
3.4.3. Строение и действие системы	119
3.4.4. Понятия, описывающие составляющие.....	120

3.4.5. Понятия, описывающие действия системы.....	121
3.4.6. Признак системы	122
3.4.7. Примеры связей.....	122
3.4.8. Виды структур.....	122
3.4.9. Важное свойство структуры фронтенд-приложения.....	126
3.4.10. Заключение.....	126
3.5. Проектирование	126
3.5.1. Что такое проектирование.....	126
3.5.2. Двойная работа	129
3.5.3. Проектирование приложения — проектирование системы.....	129
3.5.4. Определение компонентов — страниц приложения.....	130
3.5.5. Создание структуры UI-компонентов приложения.....	130
3.5.6. Определение источников данных	131
3.5.7. Состояние	132
3.5.8. Не только UI-компоненты.....	132
3.5.9. Откладывание решения.....	132
3.6. Реализация приложения	132
3.6.1. Реализация	132
3.6.2. Разница между проектированием и реализацией.....	133
3.6.3. Цель реализации	133
3.6.4. Порядок реализации	133
3.6.5. На какие области обратить внимание при реализации	134
3.6.6. Взаимная зависимость проектирования и реализации	134
Глава 4. Состояние.....	135
4.1. Библиотека Redux	135
4.1.1. Состояние приложения	135
4.1.2. Для чего нужна библиотека redux.....	135
4.1.3. Развитие и текущее состояние	135
4.1.4. Redux.....	136
4.1.5. Порядок действий.....	137
4.2. Стор.....	139
4.2.1. Главный компонент библиотеки	139
4.2.2. Ключевые принципы стора	139
4.2.3. Работа с библиотекой в приложении.....	139
4.2.4. Методы стора.....	140
4.2.5. Некоторые доводы в пользу библиотеки.....	140
4.3. Редьюсер	141
4.3.1. Что такое редьюсер.....	141
4.3.2. Чистая функция	142
4.3.3. Неизменяемость	143
4.3.4. Какую структуру стора выбрать	144
4.3.5. Начальное состояние.....	147
4.3.6. Наименование редьюсера.....	147

4.4. Экшен. Селектор.....	147
4.4.1. Что такое экшен.....	147
4.4.2. Наименование экшена.....	148
4.4.3. Action creators.....	149
4.4.4. Хранение наименований типов экшенов в виде констант.....	150
4.4.5. Что такое селектор.....	150
4.4.6. Зачем нужен селектор.....	150
Упражнения и практические задания.....	152
4.5. Middleware. Enhancer.....	152
4.5.1. Что такое middleware.....	152
4.5.2. Разница между next и store.dispatch.....	154
4.5.3. Как использовать middleware для асинхронного кода.....	154
4.5.4. Enhancer.....	156
4.5.5. Как создать свой энхансер.....	158
Упражнения и практические задания.....	158
4.6. Redux Toolkit.....	158
4.6.1. Что представляет собой библиотека.....	158
4.6.2. Redux Thunk.....	159
4.6.3. Reselect.....	160
4.6.4. Immer.....	161
4.6.5. Autodux.....	162
4.6.6. Redux DevTools.....	163
4.6.7. RTK Query.....	164
4.6.8. normalizr.....	164
4.7. Установка Redux Toolkit и настройка стора.....	165
4.7.1. Установка Redux Toolkit.....	165
4.7.2. Создание и настройка стора.....	166
4.7.3. Объект настроек configureStore.....	167
4.7.4. Middleware в configureStore.....	168
4.7.5. Энхансеры в configureStore.....	170
4.8. createSlice.....	171
4.8.1. Что делает createSlice.....	171
4.8.2. Свойства конфигурационного объекта createSlice.....	172
4.8.3. initialState.....	172
4.8.4. reducers.....	173
4.8.5. extraReducers.....	175
4.8.6. reducerPath.....	175
4.8.7. selectors.....	175
4.8.8. Возвращаемое значение из createSlice.....	176
4.8.9. Функция в свойстве reducers.....	176
4.8.10. combineSlices.....	178
4.8.11. inject и injectInto.....	179
4.8.12. extraReducers.....	181
Упражнения и практические задания.....	182

4.9. RTK Query.....	183
4.9.1. Для чего предназначен RTK Query	183
4.9.2. Использование RTK Query	184
4.9.3. Внедрение в код для совместного использования с другим состоянием Redux Toolkit.....	185
4.9.4. createApi	185
4.9.5. Объект конфигурации createApi.....	186
4.9.6. baseQuery и queryFn.....	186
4.9.7. endpoints.....	188
4.9.8. reducerPath.....	189
4.9.9. tagTypes	189
4.9.10. Оставшиеся свойства	190
4.9.11. setupListeners.....	191
4.9.12. Возвращаемое значение функцией createApi	191
4.9.13. middleware	192
4.9.14. endpoints.....	192
4.9.15. reducer.....	192
4.9.16. Другие свойства и методы	192
Упражнения и практические задания.....	193
4.10. Redux DevTools.....	193
4.10.1. Redux DevTools и установка.....	193
4.10.2. Подключение к приложению.....	193
4.10.3. Интерфейс DevTools	194
4.10.4. Режимы экшенов	195
4.10.5. Действия с экшенами	195
4.10.6. Выбор приложения	195
4.10.7. Выбор инструмента обзора данных	196
4.10.8. Хронология экшенов.....	197
4.10.9. Выбор источника	198
4.10.10. Вид представления данных и их детальный просмотр	198
4.10.11. Диспатчер, слайдер	199
Упражнения и практические задания.....	201
4.11. Как разрабатывать приложение с использованием Redux Toolkit.....	201
4.11.1. Что позволяет делать Redux Toolkit (RTK).....	201
4.11.2. Процедура разработки	201
4.11.3. Как выделить состояние и примеры	202
4.11.4. Шаг 3 процедуры разработки приложения	202
Упражнения и практические задания.....	203
4.12. Что нужно изучить самостоятельно.....	203
4.12.1. Уже изучено.....	203
4.12.2. Redux Toolkit. Редьюсеры и экшены	203
4.12.3. Redux Toolkit. Middleware	204
4.12.4. RTK Query.....	204

Глава 5. UI	206
5.1. О библиотеке React.js	206
5.1.1. Для чего нужна React	206
5.1.2. Необходимость библиотеки	206
5.1.3. Ответственность библиотеки	207
5.1.4. Как устроено взаимодействие библиотек с браузером	208
5.1.5. Немного о компоненте	209
5.2. Элемент, компонент, приложение	209
5.3. Вывод компонента: JSX	210
5.3.1. Чем является компонент	210
5.3.2. История развития компонентов в React	212
5.3.3. Обработка данных современного компонента на основе функции	214
5.3.4. Эффекты компонента	214
5.3.5. Вывод компонента и JSX	215
5.3.6. Что возвращает компонент	216
5.4. Вывод компонента: композиция	217
5.4.1. Композиция компонентов	217
5.4.2. Более сложная композиция	219
Упражнения и практические задания	221
5.5. Обработка данных и их источники: свойства	222
5.5.1. Источники данных компонента	222
5.5.2. Что такое свойства	222
5.5.3. Константы	223
Упражнения и практические задания	224
5.6. Обработка данных и их источники: состояние	224
5.6.1. Состояние	224
5.6.2. Как создается состояние компонента	224
5.6.3. Хук	225
5.6.4. Хуки в React	225
5.6.5. Хук useState	226
5.6.6. Изменение состояния и рендер	228
5.6.7. Изменение состояния в компоненте	229
5.6.8. Рендеры и их отображение на веб-странице	229
5.6.9. Хуки React и рендеры	231
5.6.10. Если не изменилось состояние	231
5.6.11. Общепринятый способ именования	231
5.6.12. Важные выводы	231
Упражнения и практические задания	232
5.7. Эффекты	232
5.7.1. Что такое эффект	232
5.7.2. Хук useEffect	232
5.7.3. Зависимости	234
5.7.4. Сторонние API и события	236

5.7.5. Зависимости, не принимающие участия ни в установке, ни в функции очистки.....	236
5.7.6. Порядок вызова.....	236
Упражнения и практические задания.....	238
5.8. Дополнительно о компонентах.....	238
5.8.1. «Виды» компонентов.....	238
5.8.2. Свойство children.....	239
5.8.3. Использование метода массивов map для вывода.....	240
5.8.4. Слот для JSX.....	241
5.8.5. Передача компонентов по ссылке.....	243
Упражнения и практические задания.....	244
5.9. Обработка событий.....	244
5.9.1. Как назначить обработчик событий.....	244
5.9.2. Порядок действий для назначения обработчика.....	246
5.9.3. Виды событий в React.....	246
5.9.4. Именованье обработчиков событий.....	246
5.9.5. Объект события React.....	247
5.9.6. Назначение событий за пределами React.....	248
Упражнения и практические задания.....	249
5.10. Группа компонентов, работающих в связке.....	250
5.10.1. Обмен данными между компонентами.....	250
5.10.2. Обмен данными между компонентами одного уровня.....	251
5.10.3. Подъем состояния.....	252
5.10.4. Несколько компонентов, которые должны знать состояние друг друга.....	254
5.10.5. Если компоненты импортируются из сторонней библиотеки.....	254
5.10.6. Если ближайший общий родитель не является непосредственным родителем обоих компонентов.....	254
5.11. Контекст и хук useContext.....	254
5.11.1. Что такое контекст и зачем нужно с ним работать.....	254
5.11.2. Как использовать контекст в коде.....	257
5.11.3. Изменение контекста.....	259
5.11.4. Обратная связь от дочерних компонентов в корневой компонент.....	260
5.11.5. Несколько контекстов.....	261
Упражнения и практические задания.....	262
5.12. Формы.....	262
5.12.1. Что такое формы.....	262
5.12.2. Входные данные.....	262
5.12.3. Формы и React.....	263
5.12.4. Теги для создания форм.....	263
5.12.5. Виды тегов, используемых для создания форм.....	264
5.12.6. Управляемые компоненты.....	264

5.12.7. input	265
5.12.8. События	267
5.12.9. form	268
5.12.10. FormData	269
5.12.11. Валидация	271
5.12.12. Дополнительные источники	274
5.13. НОС	275
5.13.1. Что такое НОС	275
5.13.2. Перерендеры и измерение производительности	276
5.13.3. Оптимизация перерендеров	279
5.13.4. Второй параметр НОС мемо	280
5.14. useReducer — второй хук для работы с состоянием	281
5.14.1. Хук useReducer	281
5.14.2. Механизм работы составляющих	282
5.14.3. Задание начального состояния	284
5.14.4. Порядок действий для создания состояния в компоненте с помощью useReducer	285
5.14.5. Использование useReducer	285
Упражнения и практические задания	285
5.15. useRef	286
5.15.1. Хук useRef	286
5.15.2. Где чаще всего применяется хук	287
5.15.3. Совместная работа хука useRef и атрибута элемента ref	287
5.15.4. Практика применения useRef в DOM-узлах	288
5.15.5. Хранение ссылки на DOM-узел в родителе	289
Упражнения и практические задания	291
5.16. Другие хуки React	291
5.16.1. useMemo	291
5.16.2. Когда имеет смысл использовать useMemo	293
5.16.3. Хук useCallback	293
5.16.4. В каких случаях использовать хук useCallback	294
5.16.5. Как использовать хук useCallback	295
5.16.6. Разница между useMemo и useCallback	296
5.16.7. Хук useId	297
5.17. Кастомные хуки	298
5.17.1. Что такое кастомный хук	298
5.17.2. Переиспользование обработки данных	298
5.17.3. Составляющие кастомного хука и некоторые правила	298
5.17.4. Как написать кастомный хук	301
5.17.5. Хуки React, задействуемые в кастомных хуках	302
5.17.6. Как приспособиться писать кастомный хук или просто облегчить его создание	303
Упражнения и практические задания	303

5.18. Объединение библиотек React и Redux.....	304
5.18.1. Библиотека React-Redux.....	304
5.18.2. Provider.....	304
5.18.3. Хуки.....	305
5.19. Новый взгляд на React.....	306
5.19.1. Что здесь освещено	306
5.19.2. На что направлять внимание.....	306
5.19.3. Структура приложения и роли компонентов в ней.....	307
5.19.4. Чтение компонента	310
5.19.5. Обмен данными между компонентами	311
5.19.6. Все есть функция, но роли разные.....	318
5.19.7. DAG.....	320
5.19.8. DAG React-компонентов.....	320
5.19.9. DAG вызовов → DOM.....	321
Упражнения и практические задания.....	323
5.20. Что нужно изучить самостоятельно.....	323
5.20.1. Хуки.....	323
5.20.2. lazy.....	324
5.20.3. startTransition и useTransition	324
Глава 6. Управление рендерингом компонентов с помощью изменяющегося URL	325
6.1. О библиотеке React-Router-Dom	325
6.1.1. Назначение библиотеки.....	325
6.1.2. React Router позволяет создавать SPA.....	326
6.2. Визуализация работы React Router.....	327
6.2.1. Граф компонентов приложения без библиотеки React Router.....	327
6.2.2. Граф компонентов приложения, управляемый с помощью библиотеки React Router	328
6.2.3. Заключение	331
6.3. Конструирование графа приложения с помощью конфигурации.....	331
6.3.1. Конфигурация — главный настраиваемый элемент.....	331
6.3.2. Способы задания конфигурации приложения.....	331
Упражнения и практические задания.....	334
6.4. Настройка Route	334
6.4.1. Указание более одного сегмента в пути	335
6.4.2. Случай первый, когда указано более одного сегмента в пути.....	335
6.4.3. Случай второй, когда маршрут содержит вложенные маршруты.....	336
6.4.4. Динамический сегмент и параметры URL.....	338
Упражнения и практические задания.....	339
6.5. Загрузка данных и обработка ошибок.....	340
6.5.1. Что такое загрузка данных.....	340
6.5.2. Динамический сегмент и загрузка данных	340

6.5.3. Параметр загрузчика request.....	341
6.5.4. Возврат ответов	341
6.5.5. Утилита json.....	342
6.5.6. Возврат ошибки.....	342
6.5.7. errorElement	342
6.5.8. Всплытие ошибки	344
6.5.9. Заключение	344
6.6. Компонент Form.....	344
6.6.1. Что представляет собой компонент Form	344
6.6.2. action.....	345
6.6.3. method	346
6.6.4. state	349
6.6.5. Заключение	350
6.7. Лэйаут	350
6.7.1. Псевдолэйаут.....	352
6.7.2. Лэйаут для одного маршрута.....	352
6.7.3. Маршрут с лэйаутом.....	352
6.7.4. Link.....	353
Упражнения и практические задания.....	353
6.8. Составляющие библиотеки	354
6.8.1. Функции.....	354
6.8.2. Хуки.....	355
6.8.3. Компоненты	358
6.9. Что изучить самостоятельно.....	360
6.9.4. Определения терминов, используемых в React Router.....	362
Глава 7. Файлы приложения.....	364
7.1. Как организовать файлы приложения	364
7.1.1. Цель	364
7.1.2. Элементы приложения, размещаемые в файлах.....	364
7.1.3. Принцип 1: роли компонентов — главный критерий их распределения по папкам.....	365
7.1.4. Принцип 2: группировка и глубина размещения остальных элементов в папках зависит от количества ролей компонентов, в которых они задействуются.....	366
7.1.5. Чтение.....	367
7.1.6. Количество элементов в файле и условие, уточняющее принцип 2.....	368
7.1.7. Файл index.js.....	368
7.1.8. Компоненты, используемые только в одном месте.....	368
7.1.9. Конкретное назначение функциональности	369
7.1.10. Рост приложения	369
Упражнения и практические задания.....	369

Глава 8. Разработка фронтенд-приложения.....	370
8.1. Проектирование ZRApp.....	370
8.1.1. Основное выполняемое действие. Что принять во внимание.....	370
8.1.2. Процедура проектирования.....	371
8.1.3. Структура приложения.....	389
8.1.4. Качество исполнения схем.....	390
8.1.5. Если изменять последовательность шагов процедуры или игнорировать проектирование вовсе и сразу начинать с написания кода.....	390
8.1.6. Дополнительный стимул, или Как не делать ненужную работу.....	392
Упражнения и практические задания.....	392
8.2. Реализация ZRApp. Настройки.....	392
8.2.1. Настройки.....	393
8.2.2. Подготовка файлов приложения перед написанием кода.....	394
8.2.3. Аналитика.....	396
8.2.4. Выбор подхода использования CSS.....	398
8.3. Реализация ZRApp. Написание кода.....	403
8.3.1. Что подразумевается под реализацией.....	403
8.3.2. Очередность реализации компонентов.....	403
8.3.3. Пример очередности реализации компонентов.....	404
8.3.4. Объект конфигурации стора ReduxToolkit.....	406
8.3.5. Объект конфигурации маршрутизатора React Router.....	406
8.3.6. Составляющие, необходимые для реализации компонента.....	406
8.3.7. Исходные данные в отношении поведения, реализуемого в компоненте.....	409
8.3.8. Решение, которое нужно придумать для реализации компонента.....	412
8.3.9. Реализация компонента.....	412
8.3.10. Хуки.....	425
8.3.11. Время, затрачиваемое на реализацию.....	426
8.3.12. Семантический HTML.....	427
Упражнения и практические задания.....	427
8.4. Пример реализации компонентов с помощью конструктора.....	427
8.4.1. Layout.....	428
8.4.2. ShedulePage.....	429
8.4.3. SheduleItemDetails.....	434
8.4.4. Заключение.....	438
8.5. Использование процедуры проектирования ZRApp и реализация ZRApp-приложения.....	438
8.5.1. Этап 1. Применение процедуры проектирования ZRApp-приложения.....	439
8.5.2. Этап 2. Реализация ZRApp.....	451
Упражнения и практические задания.....	457

8.6. Использование нейросетей во фронтенд-разработке	458
8.6.1. Причина.....	458
8.6.2. Новое поколение инструментов	458
8.6.3. GigaChat.....	458
8.6.4. Как задействовать	458
8.6.5. Немного про «общение» с ИИ.....	460
8.6.6. Используем GigaChat для написания кода.....	461
8.6.7. Используем GigaCode для написания кода	463
Упражнения и практические задания.....	466
Глава 9. Совместная работа над приложением	467
9.1. Как изучать существующее приложение.....	467
9.1.1. Метод погружения.....	467
9.1.2. Формирование структуры	469
9.1.3. Вывод	470
9.2. Код-ревью	471
9.2.1. Что такое код-ревью.....	471
9.2.2. Эмоции и код-ревью	471
9.2.3. Качество код-ревью.....	472
9.2.4. Неочевидное становится явным	473
9.2.5. Код-ревью — это групповой процесс	473
9.2.6. Быть в курсе изменений	473
9.2.7. Обучаться	474
9.2.8. Обучать.....	474
9.2.9. Код-ревью — это общение.....	474
9.3. Налаживаем процедуру код-ревью в группе.....	475
9.3.1. Код-ревью — этап процесса производства программного продукта.....	475
9.3.2. Составляющие код-ревью.....	476
9.3.3. Пример взаимодействия при код-ревью.....	477
9.3.4. Группа в мессенджере.....	477
9.3.5. Время прохождения код-ревью. Где могут возникать трудности.....	478
9.3.6. Инструмент и метод создания качественного ПР.....	478
9.3.7. Как внедрить или улучшить процедуру код-ревью	479
9.4. Как быстро стать своим в новом коллективе	479
9.4.1. Нужны ли написанные правила.....	480
9.4.2. Правила крупных компаний	480
9.4.3. Новые договоренности.....	481
9.4.4. Как составить чек-лист	481
9.4.5. Как составить чек-лист, если у группы нет практики код-ревью	483
Глоссарий.....	484