

Оглавление



Вступительное слово	18
Предисловие	21
Благодарности	23
О книге	25
Для кого эта книга	25
Структура и план книги	25
Форум liveBook	27
Об авторе	28
От издательства	28
ЧАСТЬ 1. ЧТО ТАКОЕ НЕПРЕРЫВНАЯ ДОСТАВКА.	29
Глава 1. Добро пожаловать в «Грокаем Continuous Delivery»	30
Нужна ли вам непрерывная доставка	31
Зачем нужна непрерывная доставка	32
Непрерывная доставка	33
Интеграция	35
Непрерывная интеграция	36
Что мы доставляем	37
Доставка	38
Непрерывная доставка / непрерывное развертывание	39
Элементы непрерывной доставки	41
Заключение	42
Итоги	42
Далее... ..	42

Глава 2. Базовый пайплайн	43
Веб-сайт Cat Picture	44
Исходный код сайта Cat Picture	45
Пайплайны сайта Cat Picture	46
Что же такое пайплайн и что такое задача	47
Основные задачи в пайплайне CD	48
Шлюзы и преобразования	49
CD: шлюзы и преобразования	50
Пайплайн сервисов сайта Cat Picture	51
Запуск пайплайна	52
Запуск один раз в сутки	53
Пробуем использовать непрерывную интеграцию	54
Использование уведомлений	55
Масштабирование в ручном режиме	56
Автоматизация посредством веб-хуков	57
Масштабирование при наличии веб-хуков	58
Не отправляйте изменения при сбое пайплайна	59
Непрерывная доставка для сайта Cat Picture	60
Как насчет терминологии?	61
Заключение	62
Итоги	62
Далее...	62

ЧАСТЬ 2. ПОДДЕРЖАНИЕ ПО В СОСТОЯНИИ ГОТОВНОСТИ К ДОСТАВКЕ

63

Глава 3. Контроль версий — единственно верный путь	64
Стартап Саши и Сары	65
Все виды данных	66
Исходный код и программы	67
Репозитории и версии	68
Непрерывная доставка и контроль версий	68
Git и GitHub	70
Первый коммит — с багом!	71
Нарушение работы главной ветки	73
Отправка (push) и вытягивание (pull) изменений	74
Это действительно непрерывная доставка?	76
Поддерживайте возможность выпуска релизов	77
Срабатывание при изменениях в системе контроля версий	78
Запуск пайплайна сервиса пользователей	79

Сборка сервиса пользователей	80
Сервис пользователей в облаке	81
Подключение к базе данных RandomCloud	82
Управление сервисом пользователей	83
Сбой в работе сервиса пользователей	84
Автоматизация всех переиграла	85
В чем «источник истины»?	86
Принцип «конфигурация как код» для сервиса пользователей	88
Настройка Deployaker	90
Принцип «конфигурация как код»	91
Раскатка изменений в ПО и конфигурации	92
Заключение	94
Итоги	94
Далее...	94
Глава 4. Эффективное использование линтинга	95
Бекки и проект Super Game Console	96
Линтинг спешит на помощь!	97
Вся правда о линтинге	98
Сказ о Pylint и множестве ошибок	99
Унаследованный код: системный подход	100
Шаг 1. Настройка линтера в соответствии со стандартами написания кода	101
Шаг 2. Определение базового уровня	102
Шаг 3. Принудительная проверка при поступлении данных	103
Добавление принудительной проверки в пайплайн	104
Шаг 4. «Разделяй и властвуй»	106
Изолирование: не все нужно исправлять	107
Принудительное изолирование	108
Не все проблемы одинаковы	109
Какие проблемы выявляет линтинг	110
Сначала баги, потом стиль	111
Новые препятствия	112
Заключение	114
Итоги	114
Далее...	114
Глава 5. Работа с нестабильными тестами	115
Непрерывная доставка и тесты	116
Сбой в работе сервиса Ice Cream for All	117

Сигнал и шум	118
«Нестабильный успех»	119
Как провалы тестов становятся шумом	120
От шума к сигналу	122
Цель — зеленый режим	123
Опять сбой!	124
Успешно прошедшие тесты могут оказаться нестабильными	125
Исправление ошибок в тестах	126
Виды провала тестов: flaky-тесты	127
Реагирование на сбой	128
Код или тест: что менять, исправляя тесты?	129
Опасности повторного тестирования	130
Пересмотр повторного тестирования	131
Зачем же перезапускать тесты?	133
Добившись «зеленого» статуса, поддерживайте его	133
Заключение	136
Итоги	136
Далее...	136
Глава 6. Ускорение медленных наборов тестов	137
Веб-сайт Dog Picture	138
Когда простое слишком просто	139
Новый разработчик пытается отправить код	140
Тесты и непрерывная доставка	141
Диагноз: слишком медленно	142
Пирамида тестов	143
Сначала быстрые тесты	144
Два пайплайна	145
Соблюдение баланса	146
Меняем пирамиду тестов	147
Безопасная корректировка тестов	148
Покрытие тестами	149
Как обеспечить тестовое покрытие	150
Тестовое покрытие в пайплайне	151
Перемещение тестов по пирамиде с учетом покрытия	152
Как двигаться вниз по пирамиде?	153
Унаследованные тесты и FUD	155
Параллельный запуск тестов	156
Когда можно выполнять тесты параллельно?	157
Обновление пайплайнов	158

Все еще слишком медленно!	159
Шардинг тестов, он же параллельный режим ++	160
Как использовать шардинг	161
Более сложный шардинг	162
Шардированный пайплайн	163
Шардинг браузерных тестов	164
Шардинг в пайплайне	165
Пайплайны сайта Dog Picture	166
Заключение	170
Итоги	170
Далее...	170

Глава 7. Нужные сигналы в нужное время 171

Веб-сайт CoinExCompare	172
Жизненный цикл изменения	173
Проводите непрерывную интеграцию только перед слиянием	174
Последовательность появления багов при внесении изменений	175
CI, проводимая только перед слиянием, пропускает ошибки	176
История двух графиков: установка значения по умолчанию «семь дней» ..	177
История двух графиков: установка значения по умолчанию «30 дней»	178
Конфликтующие изменения не всегда удастся отловить	179
А как же юнит-тесты?	180
Запуск на основе PR не защищает систему от багов	181
Непрерывная интеграция до И после слияния	183
Вариант 1: периодический запуск CI	183
Вариант 1: настройка периодической CI	184
Вариант 2: постоянное обновление веток	186
Вариант 2: какой ценой?	187
Вариант 3: автоматическое слияние силами CI	188
Вариант 3: запуск CI с последней версией главной ветки	189
Вариант 3: события слияния	190
Вариант 3: очереди слияний	191
Вариант 3: очередь слияния для CoinExCompare	192
Где еще могут возникнуть баги?	194
Флейки и CI, инициируемая пул-реквестом	195
Отлавливание флейков периодическими тестами	196
Баги и сборка	197
Непрерывная интеграция либо сборка и развертывание	198
Сборка и развертывание с использованием одинаковой логики	199
Усовершенствованный пайплайн CI с процессом сборки	200

Еще раз о временной шкале изменений	201
Заключение	203
Итоги	203
Далее...	203

ЧАСТЬ 3. СДЕЛАЕМ ПРОЦЕСС ДОСТАВКИ ПРОЩЕ 205

Глава 8. Простая доставка начинается с контроля версий 206

Тем временем на сайте Watch Me Watch	207
Метрики DORA	208
Метрики скорости для Watch Me Watch	209
Время доставки изменений	211
Сайт Watch Me Watch и наилучшие показатели	212
Повышаем скорость работы в Watch Me Watch	213
Интеграция с AllCatsAllTheTime	214
Инкрементная доставка функций	216
Коммит с пропущенными тестами	217
Проверка кода и неполный код	218
Двигаемся дальше	220
Коммит незавершенного кода	221
Ревью незавершенного кода	222
Вернемся к сквозным тестам	223
Оценим преимущества	224
Сокращение времени доставки изменений	226
Продолжим работу над AllCatsAllTheTime	227
Окна развертывания и заморозка кода	228
Повышение скорости	229
Заключение	230
Итоги	230
Далее...	230

Глава 9. Безопасная и надежная сборка 231

Веб-сайт Top Dog Maps	232
Когда процесс сборки — это документ	233
Атрибуты безопасных и надежных сборок	234
Постоянная готовность к релизу	236
Автоматические сборки	236
Сборка как код	237
Использование сервиса непрерывной доставки	238
Временные среды для сборки	240

План Мигеля	241
От документации к скрипту в системе контроля версий	242
Автоматические контейнерные сборки	243
Безопасная и надежная сборка	245
Изменения интерфейса и баги	247
Когда сборки вызывают баги	248
Сборки и коммуникация	249
Семантическое версионирование	250
Важность версионирования	251
Очередной сбой!	254
Ошибки в зависимостях во время сборки	255
Привязка зависимостей	256
Привязка к версии не панацея	258
Привязка к хешам	259
Заключение	261
Итоги	261
Далее...	261
Глава 10. Надежное развертывание	262
Множество проблем с развертыванием	263
Метрики DORA для стабильности	264
Метрики DORA для сайта Plenty of Woofs	266
А если делать развертывания реже?	267
А если делать развертывания чаще?	268
Ежедневные развертывания и сбои	269
Как увеличить частоту развертываний	270
Устраняем недостатки разработки	271
Скользящие обновления	272
Исправление багов при скользящем обновлении	273
Откаты	274
Откат = немедленное улучшение	275
Политика откатов в действии	277
Сине-зеленые развертывания	278
Быстрое восстановление при сине-зеленом развертывании	279
Быстрее и стабильнее с «канарейками»	280
Требования к канареечным развертываниям	281
Канареечное развертывание параллельно с базовым	283
Время восстановления работоспособности при канареечном развертывании	284
Увеличение частоты развертывания	286

Метрики DORA при ежедневных канареечных развертываниях	287
Непрерывное развертывание	288
Когда использовать непрерывное развертывание	289
Обязательные этапы контроля качества	290
Контроль качества и непрерывное развертывание	291
Суперэффективность согласно метрикам DORA	293
Заключение	294
Итоги	294
Далее...	294

ЧАСТЬ 4. РЕАЛИЗАЦИЯ НЕПРЕРЫВНОЙ ДОСТАВКИ 295

Глава 11. Стартовые наборы: с нуля до CD 296

Стартовые наборы: обзор	297
Универсальные задачи пайплайна CD	298
Прототип пайплайна выпуска	299
Прототип пайплайна CI	300
Пайплайны и их инициализация	301
Проект с нуля: переходим к CD	303
Проект Gulpy	304
Проект greenfield: с нуля до CD	305
Первый шаг: выполняется ли сборка?	306
Выбор системы CD	307
Настройка начальной автоматизации	308
Состояние кода: линтинг	309
Состояние кода: юнит-тесты	310
Состояние кода: покрытие	311
Что дальше: публикация	312
Развертывание	313
Расширяем тестирование	314
Задачи для интеграционных и сквозных тестов	315
Завершаем пайплайн CI	316
Окончательные пайплайны проекта Gulpy	317
Унаследованный проект: переход к CD	318
Проект Rebellious Hamster	319
Первый шаг: установление приоритетов для инкрементных целей	320
В первую очередь «болевые точки»	321
«Болевые точки» в проекте Rebellious Hamster	322
Всегда узнавать о сбоях	323
Изолируем код и добавляем тесты	324

Добавление большего количества тестов в унаследованный пайплайн ...	325
Увеличение степени автоматизации развертывания	326
Создание пайплайна выпуска	327
Пайплайн выпуска Rebellious Hamster	328
Окончательный вид пайплайнов Rebellious Hamster	329
Заключение	330
Итоги	330
Далее...	330
Глава 12. Скрипты — это тоже код	331
Онлайн-банк Purrfect Bank	332
Проблемы непрерывной доставки	333
Схема непрерывной доставки в онлайн-банке Purrfect Bank	334
Bash-библиотеки подразделения платежей	335
Пайплайн сервиса транзакций	336
Избавляемся от одного большого скрипта	337
Принципы хорошо спроектированной задачи	338
Разделение гигантской задачи на части	339
Обновленный пайплайн сервиса транзакций	340
Отладка bash-библиотек	341
Исследуем баг в bash-библиотеке	342
Почему появился это баг?	343
Для чего нужен bash	344
Когда bash не подходит	345
Сравнение языков shell-скриптов и языков общего назначения	347
От языка shell-скриптов к языку общего назначения	348
План перехода	349
От библиотеки bash к задаче с bash-скриптом	350
Многоразовый bash-скрипт внутри задачи	351
Переход от bash к Python	352
Задачи как код	353
Скрипты CD — это тоже код	354
Заключение	355
Итоги	355
Далее...	355
Глава 13. Реализация пайплайнов	356
Проект PetMatch	357
Пайплайны CD сервиса подбора соответствий	358
Проблемы пайплайнов CD	359

Пайплайн сквозного тестирования	360
Пайплайн сквозного тестирования и ошибки	361
Finally-логика	362
Концепция Finally в виде графа	363
Принцип Finally в пайплайне сервиса подбора соответствий	364
Пайплайн сквозного тестирования и скорость	365
Параллельное выполнение задач	366
Пайплайн сквозного тестирования и скорость выполнения тестов	367
Параллельное выполнение и шардинг тестов	367
Пайплайн сквозного тестирования с шардингом	369
Пайплайн сквозного тестирования и сигналы	370
Единый пайплайн CI	371
Пайплайн выпуска релизов и сигналы	372
Различия в пайплайнах CI и выпуска релизов	373
Комбинация пайплайнов	374
Пайплайн выпуска релизов	375
Хардкодинг в пайплайне выпуска релизов	376
Повторное использование пайплайна с помощью параметризации	377
Применение повторно используемых пайплайнов	378
Обновленные пайплайны	379
Решение проблем непрерывной доставки в проекте PetMatch	380
Важные особенности непрерывной доставки	380
Заключение	382
Итоги	382
Далее...	382

ПРИЛОЖЕНИЯ 383

Приложение А. Системы непрерывной доставки 384

Приложение В. Системы контроля версий 393