
НЕПРЕРЫВНОЕ РАЗВЕРТЫВАНИЕ ПО

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ
СБОРКИ, ТЕСТИРОВАНИЯ
И ВНЕДРЕНИЯ НОВЫХ ВЕРСИЙ
ПРОГРАММ

CONTINUOUS DELIVERY

Jez Humble and David Farley



ADDISON-WESLEY

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

НЕПРЕРЫВНОЕ РАЗВЕРТЫВАНИЕ ПО

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ
СБОРКИ, ТЕСТИРОВАНИЯ
И ВНЕДРЕНИЯ НОВЫХ ВЕРСИЙ
ПРОГРАММ

Джез Хамбл
Дейвид Фарли



Москва • Санкт-Петербург • Киев
2011

ББК 32.973.26-018.2.75

X18

УДК 681.3.07

Издательский дом “Вильямс”

Главный редактор *С.Н. Тригуб*

Зав. редакцией *В.Р. Гинзбург*

Перевод с английского канд. техн. наук *А.Г. Сысолюка*

Под редакцией канд. техн. наук *А.Г. Сысолюка* и *В.Р. Гинзбурга*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:
info@williamspublishing.com, <http://www.williamspublishing.com>

Хамбл, Джек, Фарли, Дейвид.

X18 Непрерывное развертывание ПО: автоматизация процессов сборки, тестирования и внедрения новых версий программ. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2011. — 432 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1739-3 (рус.)

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Addison-Wesley Publishing Company, Inc.

Authorized translation from the English language edition published by Addison-Wesley, Copyright © 2011.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the publisher.

Russian language edition is published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2011.

Научно-популярное издание

Джек Хамбл, Дейвид Фарли

Непрерывное развертывание ПО: автоматизация процессов сборки, тестирования и внедрения новых версий программ

Литературный редактор *Е.П. Перестюк*

Верстка *М.А. Удалов*

Художественный редактор *В.Г. Павлютин*

Корректор *Л.А. Гордиенко*

Подписано в печать 16.07.2011. Формат 70x100/16

Гарнитура Times. Печать офсетная

Усл. печ. л. 34,8. Уч.-изд. л. 31,5

Тираж 1000 экз. Заказ № 0000

Отпечатано по технологии СтР в ОАО “Первая Образцовая типография”
обособленное подразделение “Печатный двор”.

197110, Санкт-Петербург, Чкаловский пр., 15

ООО “И. Д. Вильямс”, 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1739-3 (рус.)

ISBN 978-0-321-60191-9 (англ.)

© Издательский дом “Вильямс”, 2011

© Pearson Education, Inc., 2011

Оглавление

Предисловие Мартина Фаулера	18
Введение	20
Благодарности	27
Об авторах	28
Часть I. Основы непрерывного развертывания	29
Глава 1. Проблема развертывания программного обеспечения	31
Глава 2. Стратегии управления конфигурациями	55
Глава 3. Непрерывная интеграция	77
Глава 4. Реализация стратегии тестирования	103
Часть II. Конвейер развертывания	119
Глава 5. Структура конвейера развертывания	121
Глава 6. Сценарии сборки и развертывания	153
Глава 7. Стадия фиксации	177
Глава 8. Автоматическое приемочное тестирование	193
Глава 9. Тестирование нефункциональных требований	227
Глава 10. Развертывание и выпуск приложений	251
Часть III. Процесс поставки	273
Глава 11. Управление инфраструктурой и средами	275
Глава 12. Управление данными	317
Глава 13. Управление компонентами и зависимостями	335
Глава 14. Управление версиями	367
Глава 15. Управление непрерывным развертыванием	399
Список литературы	423
Предметный указатель	425

Содержание

Предисловие Мартина Фаулера	18
Введение	20
Для кого предназначена книга	21
Структура книги	23
Часть I. Основы непрерывного развертывания	23
Часть II. Конвейер развертывания	24
Часть III. Процесс поставки	24
Веб-ссылки в книге	25
Изображение на обложке	25
Благодарности	27
Об авторах	28
Часть I. Основы непрерывного развертывания	29
Глава 1. Проблема развертывания программного обеспечения	31
Введение	31
Распространенные антишаблоны поставки релизов	32
Антишаблон: развертывание ПО вручную	33
Антишаблон: развертывание в среде производственного типа только после завершения разработки	34
Антишаблон: ручное управление конфигурацией рабочих сред	36
Можно ли улучшить технологию развертывания	37
Как мы собираемся достичь наших целей	38
Любое изменение должно инициировать обратную связь	39
Обратная связь должна срабатывать как можно быстрее	40
Команда поставки должна получить информацию и отреагировать на нее	41
Масштабирование процесса непрерывного развертывания	42
Преимущества непрерывного развертывания	42
Расширение полномочий команды	43
Уменьшение количества ошибок	43
Смягчение стрессов	45
Гибкость развертывания	46
Достижение совершенства на практике	47
Релиз-кандидат	47
Каждое изменение может привести к выпуску новой версии	48
Принципы развертывания ПО	49
Создайте надежный повторяющийся процесс развертывания	49
Автоматизируйте все, что только можно	50
Контролируйте все с помощью системы управления версиями	50
Если операция болезненная, выполняйте ее чаще	50
Встраивайте качество в продукцию	51

СОДЕРЖАНИЕ	7
Готово — значит выпущено	51
Каждый отвечает за процесс поставки	52
Непрерывное улучшение	52
Резюме	53
Глава 2. Стратегии управления конфигурациями	55
Введение	55
Управление версиями	56
Контролируйте абсолютно все с помощью системы управления версиями	57
Регулярно регистрируйте изменения на магистрали (основной ветви)	59
Используйте информативные сообщения фиксации	60
Управление зависимостями	61
Управление внешними библиотеками	62
Управление компонентами	62
Управление конфигурациями	63
Конфигурация и гибкость	63
Типы конфигураций	64
Управление конфигурацией приложения	66
Управление конфигурациями нескольких приложений	69
Принципы управления конфигурациями приложений	70
Управление средами	71
Инструменты управления средами	74
Управление процессом изменения	74
Резюме	75
Глава 3. Непрерывная интеграция	77
Введение	77
Реализация непрерывной интеграции	78
Начальные требования	78
Базовая система непрерывной интеграции	79
Обязательные условия непрерывной интеграции	81
Регулярно регистрируйте изменения	81
Создайте полный набор автоматических тестов	81
Процессы сборки и тестирования должны быть быстрыми	82
Управление средой разработки	83
Использование программ непрерывной интеграции	84
Базовые возможности	84
Расширенные возможности	84
Важные методики	86
Не регистрируйте изменения в нерабочей сборке	87
Всегда заставляйте делать это сервер непрерывной интеграции либо заставьте делать это сервер непрерывной интеграции	87
Дождитесь завершения тестов фиксации	88
Не уходите домой, когда есть нерабочая сборка	88
Всегда будьте готовы вернуться к предыдущей версии	89
Включите секундомер	90
Не отключайте тесты в случае сбоя	90
Вы отвечаете за все свои, произошедшие из-за ваших изменений	91
Разработка через тестирование	91
Предлагаемые методики	92

Экстремальное программирование	92
Отмена сборки из-за нарушения архитектурных ограничений	92
Отмена сборки из-за медлительности тестов	93
Отмена сборки из-за предупреждений и нарушений стиля кодирования	93
Распределенные команды	94
Влияние на процесс	95
Централизованная непрерывная интеграция	95
Технические проблемы	96
Альтернативные подходы	97
Распределенные системы управления версиями	98
Резюме	101
Глава 4. Реализация стратегии тестирования	103
Введение	103
Типы тестов	104
Тесты, ориентированные на деловую логику и поддерживающие процесс разработки	104
Тесты, ориентированные на технологию и поддерживающие процесс разработки	108
Тесты, ориентированные на деловую логику и критикующие проект	108
Тесты, ориентированные на технологию и критикующие проект	109
Тестовые двойники	110
Реальные ситуации и стратегии	111
Начало проекта	111
Середина проекта	112
Устаревшие системы	113
Интеграционное тестирование	114
Создание тестов	116
Управление списками неисправленных дефектов	117
Резюме	118
Часть II. Конвейер развертывания	119
Глава 5. Структура конвейера развертывания	121
Введение	121
Что такое конвейер развертывания	122
Базовый конвейер развертывания	126
Методики применения конвейера развертывания	127
Каждая сборка двоичного кода должна быть единственной	128
Используйте один и тот же способ развертывания в каждой среде	130
Выполняйте дымовые тесты развертываний	131
Развертывайте приложение в копии рабочей среды	132
Каждое изменение должно немедленно продвигаться по конвейеру развертывания	132
Если любая часть конвейера терпит неудачу, остановите конвейер	134
Стадия фиксации	134
Рекомендуемые методики этапа фиксации	135
Автоматические приемочные тесты	136
Рекомендуемые методики автоматического приемочного тестирования	138
Поздние стадии тестирования	139

Ручное тестирование	141
Тестирование нефункциональных требований	141
Подготовка к выпуску	142
Автоматизация развертывания и поставки релиза	142
Откат изменений	144
Стратегия успеха	145
Реализация конвейера развертывания	145
Моделирование потока создания ценности и построение рабочего каркаса	145
Автоматизация процессов сборки и развертывания	146
Автоматизация модульных тестов и анализ кода	147
Автоматизация приемочных тестов	148
Развитие конвейера развертывания	148
Метрики	149
Резюме	152
Глава 6. Сценарии сборки и развертывания	153
Введение	153
Обзор инструментов сборки	154
Make	156
Ant	157
NAnt и MSBuild	158
Maven	158
Rake	159
Buildr	160
Psake	160
Принципы создания сценариев сборки и развертывания	160
Создавайте сценарии для каждой стадии конвейера развертывания	160
Используйте наиболее подходящие инструменты для развертывания	161
Применяйте одни и те же сценарии в каждой среде	162
Используйте инструменты пакетирования, предоставляемые операционной системой	162
Обеспечьте идемпотентность процесса развертывания	164
Совершенствуйте систему развертывания инкрементным способом	165
Структура проекта с приложениями для JVM	165
Структура проекта	165
Сценарии развертывания	168
Слой развертывания и тестирования	169
Тестирование конфигурации среды	170
Советы и трюки	171
Всегда используйте относительные пути	171
Устраняйте ручные этапы	172
Встраивайте средства отслеживания версий	172
При сборке не регистрируйте двоичные файлы в системе управления версиями	173
Неуспешные тесты не должны отменять сборку	173
Ограничивайте приложение интегрированными дымовыми тестами	174
Советы и трюки для .NET	174
Резюме	174

Глава 7. Стадия фиксации	177
Введение	177
Принципы и методики стадии фиксации	178
Создайте быструю и надежную обратную связь	178
Что должно завершать стадию фиксации	180
Непрерывно улучшайте стадию фиксации	180
Передайте полномочия разработчикам	181
В очень больших командах назначайте администратора сборок	182
Результаты стадии фиксации	182
Хранилище артефактов	182
Принципы и методики создания набора тестов фиксации	185
Избегайте пользовательского интерфейса	186
Используйте внедрение зависимостей	186
Не обращайтесь к базам данных	186
Избегайте асинхронности в модульных тестах	187
Используйте тестовые двойники	187
Минимизируйте состояния в тестах	190
Поддельвайте время	191
Применяйте метод “грубой силы”	191
Резюме	192
Глава 8. Автоматическое приемочное тестирование	193
Введение	193
Важность автоматического приемочного тестирования	194
Создание набора приемочных тестов, удобных для сопровождения	196
Тестирование графического пользовательского интерфейса	198
Создание приемочных тестов	199
Роль аналитиков и тестировщиков	199
Значение анализа в итеративных проектах	199
Приемочные критерии как выполняемые спецификации	200
Слой драйверов приложения	203
Представление приемочных критериев	206
Шаблон драйверов окон: отделение тестов от графического интерфейса пользователя	206
Реализация приемочных тестов	209
Состояния в приемочном тестировании	209
Границы процессов, инкапсуляция и тестирование	211
Управление асинхронностью и тайм-аутами	212
Использование тестовых двойников	214
Стадия приемочного тестирования	217
Контроль приемочных тестов	218
Тесты развертывания	220
Производительность приемочных тестов	221
Выполняйте рефакторинг общих задач	222
Сделайте дорогостоящие ресурсы общими	223
Параллельное тестирование	223
Использование вычислительных решеток	224
Резюме	225

Глава 9. Тестирование нефункциональных требований	227
Введение	227
Управление нефункциональными требованиями	228
Анализ нефункциональных требований	229
Программирование с учетом производительности	230
Измерение производительности	232
Как определять успех и неудачу в тестах производительности	233
Среда тестирования производительности	235
Автоматизация тестов производительности	238
Тестирование производительности посредством пользовательского интерфейса	241
Запись взаимодействия со службой или открытым программным интерфейсом	242
Использование шаблонов записанных взаимодействий	242
Использование заглушек при разработке тестов производительности	244
Добавление тестов производительности в конвейер развертывания	245
Дополнительные преимущества системы тестирования производительности	247
Резюме	248
Глава 10. Развертывание и выпуск приложений	251
Введение	251
Создание стратегии поставки релиза	252
План выпуска	253
Поставка коммерческого программного продукта	253
Развертывание и продвижение приложения	254
Первое развертывание	254
Моделирование процесса поставки и продвижения сборок	255
Продвижение конфигураций	257
Согласование сред и приложений	258
Развертывание в отладочных средах	258
Откат развертываний и релизы с нулевым временем простоя	259
Откат путем повторного развертывания последней хорошей версии	260
Релизы с нулевым временем простоя	260
Сине-зеленое развертывание	261
Канаречные релизы	262
Аварийные исправления	265
Непрерывное внедрение	266
Непрерывная поставка приложений, устанавливаемых пользователями	267
Советы и трюки	269
Люди, выполняющие развертывание, должны участвовать в создании процесса развертывания	269
Создавайте журналы развертывания	269
Старые файлы нужно не удалять, а перемещать	270
За развертывание должна отвечать вся команда	270
У серверных приложений не должно быть графического интерфейса	270
Задавайте период “раскачки” для нового развертывания	271
Быстро реагируйте на неудачи	271
Не вносите изменения непосредственно в рабочей среде	271
Резюме	272

Часть III. Процесс поставки	273
Глава 11. Управление инфраструктурой и средами	275
Введение	275
Потребность в администраторах	277
Документация и аудит	278
Оповещения о нештатных событиях	278
Планирование непрерывности обслуживания	279
Используйте технологии, знакомые администраторам	279
Моделирование и контроль инфраструктуры	280
Управление доступом к инфраструктуре	282
Внесение изменений в инфраструктуру	283
Управление установкой и конфигурациями серверов	284
Развертывание серверов	285
Непрерывное управление серверами	286
Управление конфигурацией промежуточного ПО	291
Управление конфигурацией	291
Исследуйте продукт	293
Проанализируйте, как промежуточное ПО обрабатывает состояния	294
Найдите программный интерфейс конфигурации	294
Примените лучшую технологию	295
Управление службами инфраструктур	295
Многоканальные системы	296
Виртуализация	298
Управление виртуальными средами	300
Виртуальные среды и конвейер развертывания	302
Параллельное тестирование с помощью виртуальных сред	304
Облачные вычисления	306
Инфраструктура в облаке	307
Платформы в облаке	308
Одного рецепта от всех болезней не существует	309
Критика облачных вычислений	309
Мониторинг инфраструктуры и приложений	310
Сбор данных	311
Журналы	313
Создание информационных панелей	313
Мониторинг на основе функционирования	315
Резюме	316
Глава 12. Управление данными	317
Введение	317
Управление базами данных с помощью сценариев	318
Инициализация баз данных	318
Инкрементные изменения	319
Управление версиями баз данных	319
Управление согласованными изменениями	321
Откат баз данных и релизы с нулевым временем простоя	322
Откат без потери данных	322
Разделение процессов миграции базы данных и развертывания приложения	323

Управление тестовыми данными	324
Имитация баз данных для модульных тестов	325
Управление связями тестов с данными	325
Изоляция тестов	326
Установка данных	327
Специальные тестовые ситуации	327
Управление данными и конвейер развертывания	328
Данные для тестов стадии фиксации	328
Данные для приемочных тестов	329
Данные для тестов производительности	331
Данные для других стадий тестирования	331
Резюме	332
Глава 13. Управление компонентами и зависимостями	335
Введение	335
Поддержка готовности приложения к выпуску	336
Временное сокрытие новой функциональности	337
Вносите все изменения инкрементным способом	338
Ветвление по абстракции	339
Зависимости	341
Ад зависимостей	341
Управление библиотеками	343
Компоненты	344
Разбиение кодовой базы на компоненты	345
Компоненты и конвейер развертывания	348
Интеграционный конвейер	349
Управление графами зависимостей	351
Создание графа зависимостей	351
Графы зависимостей конвейера	353
Когда следует запускать сборки	355
Стратегия осторожного оптимизма	357
Циклические зависимости	358
Управление двоичными кодами	359
Как должно работать хранилище артефактов	360
Как конвейер развертывания должен взаимодействовать с хранилищем артефактов	361
Управление зависимостями с помощью программы Maven	361
Рефакторинг зависимостей в Maven	363
Резюме	365
Глава 14. Управление версиями	367
Введение	367
Краткая история систем управления версиями	368
CVS	368
Subversion	369
Коммерческие системы управления версиями	371
Отключите пессимистическую блокировку	371
Ветвления и слияния	373
Слияние	374
Ветви в системе непрерывной интеграции	375

Распределенные системы управления версиями	377
Что такое распределенная система управления версиями	377
Краткая история распределенных систем управления версиями	379
Распределенные системы управления версиями в корпоративных средах	380
Использование распределенных систем управления версиями	381
Потоковые системы управления версиями	383
Что такое потоковая система управления версиями	383
Потоковые модели разработки	385
Статические и динамические представления	386
Непрерывная интеграция в потоковых системах управления версиями	387
Разработка на магистрали	388
Внесение сложных изменений без ветвления	389
Ветвь для выпуска	391
Ветвление по функциональным средствам	392
Ветвление по командам	395
Резюме	397
Глава 15. Управление непрерывным развертыванием	399
Введение	399
Модель зрелости процессов, связанных с управлением конфигурациями и поставкой	400
Как использовать модель зрелости	401
Жизненный цикл проекта	403
Идентификация задачи	404
Начальная фаза проекта	404
Инициализация	406
Разработка и выпуск продукта	407
Эксплуатация	409
Управление рисками	410
Стратегии управления рисками	410
Схема управления рисками	411
Применение стратегий управления рисками	412
Симптомы и причины проблем развертывания	413
Редкие развертывания или много ошибок при развертывании	414
Низкое качество приложения	414
Неэффективность процесса непрерывной интеграции	415
Плохое управление конфигурациями	416
Соответствие стандартам и аудит	416
Автоматическая документация	417
Обеспечение доступности процессов для отслеживания	418
Изоляция команд	419
Управление изменениями	420
Резюме	421
Список литературы	423
Предметный указатель	425

Отзывы о книге

“Если вам нужно выпускать очередные версии приложений чаще, чем вы это делаете сейчас, — эта книга для вас. Применение технологии непрерывного развертывания уменьшает риски, исключает рутинную работу и увеличивает надежность кода. Я буду использовать непрерывное развертывание во всех моих текущих и будущих проектах”.

Кент Бек, Three Rivers Institute

“Эту книгу необходимо прочитать независимо от того, понимают ли ваши разработчики, что непрерывная интеграция так же необходима, как и управление исходными кодами. Книга уникальна тем, что в ней связаны воедино процессы разработки и развертывания. В книге описаны не только инструменты и методы, но также сами принципы развертывания. Эту книгу должен прочитать каждый член команды, включая программистов, тестировщиков, системных администраторов, администраторов баз данных и менеджеров проекта”.

Лайза Криспин, один из авторов книги “Гибкое тестирование”

“Для многих организаций концепция непрерывного развертывания — не просто очередная технология программирования, а неотъемлемая часть бизнеса. В книге показано, как воплотить непрерывное развертывание в реальность в вашем проекте”.

Джеймс Тернбулл, автор книги “Pulling Strings with Puppet”

“Написанная понятным и доступным языком, эта книга дает читателю исчерпывающее представление о том, чего можно ожидать от процессов развертывания. Авторы шаг за шагом раскрывают возможности технологии непрерывного развертывания и трудности, стоящие на пути ее реализации. Эта книга должна быть в библиотеке каждого программиста”.

Лейна Котран, Институт информатики, Калифорнийский университет в Ирвайне

“Авторы рассказывают о том, что делает приложения успешными на рынке программного обеспечения. Необходимость внедрения технологии непрерывного развертывания все более очевидна. Книга охватывает многие уровни разработки, что крайне важно для непрерывного развертывания”.

Джон Оллспо, вице-президент компании Etsy.com, автор книг “The Art of Capacity Planning” и “Web Operations”

“Если вы занимаетесь созданием и развертыванием служб на основе программного обеспечения, то вам будут полезны концепции, рассмотренные в данной книге. Но авторы не ограничиваются только концепциями — книга очень полезна как практическое руководство по внедрению надежных технологий развертывания”.

Деймон Эдуардс, президент компании DTO Solutions, один из редакторов сайта dev2ops.org

“Я считаю, что каждый, кто имеет дело с развертыванием программного обеспечения, сможет найти в этой книге полезную информацию, просто открыв ее на любой странице. Но еще полезнее — прочитать ее от корки до корки, что позволит значительно усовершенствовать технологию развертывания, применяемую в вашей организации. Это хороший справочник по тестированию и развертыванию программного обеспечения”.

Сара Эдри, Гарвардская школа бизнеса

“Непрерывное развертывание — естественный следующий этап после непрерывной интеграции для любой команды программистов. В книге поставлена амбициозная цель — добиться непрерывной поставки обновлений программных продуктов клиентам. Эта цель достигается на основе набора эффективных принципов и методов развертывания, рассматриваемых в данной книге”.

Роб Санхайм, руководитель компании Relevance, Inc.

Предисловие Мартина Фаулера

В конце 1990-х годов я навесил Кента Бека, который тогда работал в швейцарской страховой компании. Он ознакомил меня со своим проектом. Примечательной особенностью его дисциплинированной команды было то, что уже тогда они развертывали программный продукт в рабочей среде почти ежедневно. Регулярное развертывание предоставляло им ряд преимуществ: готовое ПО не ожидало своей очереди на развертывание, команда разработчиков могла быстро реагировать на проблемы и открывающиеся возможности, быстрая “оборачиваемость” способствовала сотрудничеству между разработчиками и командой заказчика.

Последние десять лет я работаю в компании ThoughtWorks, и общей темой в наших проектах было сокращение продолжительности цикла — интервала времени между идеей и окончательной поставкой программного продукта. Я имел дело с огромным количеством проектов, и почти все их менеджеры из всех сил пытались сократить рабочий цикл. Обычно проекты не развертываются ежедневно, однако уже сейчас нередко можно встретить поставку очередного релиза каждую неделю или дважды в неделю.

Дейв и Джек — активные участники этого межконтинентального процесса, развивающего культуру частого надежного развертывания. Они (как и другие наши коллеги и единомышленники) неустанно пропагандируют переход от развертывания программных продуктов раз в год к технологии непрерывного развертывания, в которой поставка новых версий становится ежедневной рутинной.

Фундаментом непрерывного развертывания (как минимум для команды разработчиков) служит концепция непрерывной интеграции (Continuous Integration — CI). Технология непрерывной интеграции предполагает синхронизацию усилий всех членов команды разработчиков и устранение задержек, связанных с проблемами взаимодействия разных частей проекта. Несколько лет назад Поль Дюваль написал книгу о непрерывной интеграции [14], но это лишь первый шаг на пути к непрерывному развертыванию. Программное обеспечение, которое было успешно интегрировано в главный поток проектирования кода, еще не может служить отправной точкой непрерывного развертывания. Дейв и Джек подхватили идеологию непрерывной интеграции и успешно прошли “последнюю милю” — концепцию конвейера разработки, превращающего интегрированный код в готовый продукт.

Технологии создания программного обеспечения, в которых вся идеология разработки вращается вокруг развертывания, длительное время были незаслуженно отодвинуты на второй план. Они как бы попали в “черную дыру” между командами разработки и сопровождения приложений. Поэтому неудивительно, что данная книга обращена к обеим командам, вовлекая в этот же процесс и тестировщиков, поскольку тестирование — ключевой элемент создания надежных релизов. Естественно, все эти процессы должны быть максимально автоматизированы, чтобы работа выполнялась быстро и согласованно.

Запустить всю технологию непрерывного развертывания нелегко, но затраченные усилия вознаграждаются сторицей. Большие промежутки между версиями и традиционные авралы перед поставками очередных релизов уходят в прошлое. Пользователи увидят идеи, воплощенные в реальный код, уже через несколько дней после их появления. Но, наверное, важнее всего то, что будет устранен мощный источник стрессов для разработ-

чиков. Бессонные ночи в последний день перед выпуском (как правило, с воскресенья на понедельник) не нравятся никому.

Мне кажется, что книга о технологии непрерывного развертывания программного обеспечения без традиционных стрессов будет полезна прежде всего самим разработчикам.

Введение

Вчера начальник попросил вас продемонстрировать клиентам новые замечательные средства, внедряемые в приложение, но вы не смогли показать им ничего. Все ваши работчики находятся на полпути к финальному релизу, и ни один из них не готов запустить приложение прямо сейчас. У вас есть код, он компилируется и проходит все тесты на вашем сервере непрерывной интеграции (Continuous Integration — CI), но нужно еще несколько дней, чтобы развернуть новую версию в среде приемочного тестирования (User Acceptance Testing — UAT). Имеет ли смысл показывать демонстрационную версию по первому требованию?

В программе обнаружена критическая ошибка. Компания ежедневно теряет на этом деньги. Вы знаете, что нужно исправить: единственную строчку в библиотеке, используемой во всех трех слоях трехуровневой системы. Кроме того, нужно внести соответствующие изменения в таблицу базы данных. Но при поставке предыдущей версии программы эта работа заняла все выходные вплоть до трех часов ночи в понедельник, причем разработчик, осуществлявший развертывание, вскоре уволился, заявив, что этот сумасшедший дом не для него. Вы понимаете, что и теперь за выходные не успеть, а значит, приложение окажется недоступным в течение какого-то периода в рабочие дни. Вряд ли это понравится клиентам компании.

Описанные выше проблемы, хоть и встречаются весьма часто, не являются необходимым следствием цикла разработки программного обеспечения. Скорее, наоборот: они служат признаком того, что в рабочем процессе что-то не так. Поставка очередной версии программы должна быть быстрым, часто повторяющимся процессом. В наши дни многие компании выпускают по несколько версий *ежедневно!* Это вполне возможно даже в очень сложных проектах, содержащих огромный объем кода. В данной книге мы покажем вам, как это делается.

Мэри и Том Поппендик [24] задавались вопросами: “Сколько времени необходимо организации на развертывание новой версии приложения при изменении одной строки кода? Является ли этот процесс повторяющимся и надежным часто?” Интервал времени между принятием решения об изменении кода и поставкой новой версии продукта называется *продолжительностью цикла*. Это важный показатель в любом проекте.

Во многих организациях продолжительность цикла колеблется от нескольких недель до месяцев, причем процесс подготовки релиза не является ни повторяющимся, ни надежным. Часто подготовка выполняется вручную и требует участия команды разработчиков для развертывания программы даже в тестовой и отладочной средах, не говоря уже о рабочей среде. Тем не менее нам часто встречались проекты, которые начинались, как описано выше, однако со временем были существенно доработаны, в результате чего продолжительность цикла была сокращена до нескольких часов и даже минут. Это возможно благодаря созданию полностью автоматизированного, повторяющегося и надежного процесса внесения изменений на различных стадиях сборки, тестирования и развертывания приложений. Ключевой элемент данного процесса — автоматизация, позволяющая буквально одним щелчком на кнопке выполнять разнообразные задачи, связанные с развертыванием программного обеспечения (ПО).

В данной книге описывается, без преувеличения, революционная технология развертывания, сокращающая продолжительность цикла и делающая процесс развертывания надежным и безопасным.