

Обеспечение наблюдаемости ПО

Как достичь безупречной
работы системы

**Чарити Мейджорс,
Лиз Фонг-Джонс
и Джордж Миранда**

УДК 004.45
ББК 32.972.1
М45

Observability Engineering: Achieving Production Excellence
Charity Majors, Liz Fong-Jones & George Miranda

© 2025 “ASTANA international publishing” Authorized Russian translation of the English edition of Observability Engineering ISBN 9781492076445 © 2022 Hound Technology Inc. This translation is published and sold by permission of O’Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Мейджорс, Чарити.

М45 Обеспечение наблюдаемости ПО. Как достичь безупречной работы системы / Чарити Мейджорс, Лиз Фонг-Джонс, Джордж Миранда : [перевод с английского Е. Жевлаковой]. — Алматы : Астана иностранная пресса, 2025. — 352 с. — (O’Reilly. Книги по программированию).

ISBN 978-601-12-3552-5

Наблюдаемость — основа успешного создания и управления программным обеспечением. Она ускоряет доставку кода, упрощает выявление аномалий и улучшает пользовательский опыт. В этой книге раскрывается ценность наблюдаемости, предлагаются рекомендации по переходу с устаревших инструментов и приводятся методы внедрения наблюдаемости в разработку.

Вы узнаете, как применять наблюдаемость для управления ПО на всех этапах жизненного цикла, инструментировать код для упрощения отладки, писать поддерживаемый код и анализировать данные для быстрого устранения проблем.

УДК 004.45
ББК 32.972.1

ISBN 978-601-12-3552-5

© Жевлакова Е. В., перевод на русский язык, 2025
© Издание на русском языке, оформление.
ТОО «Издательство «Астана иностранная пресса», 2025

Все права защищены. Любое воспроизведение, полное или частичное, в том числе на интернет-ресурсах, а также запись в электронной форме для частного или публичного использования возможны только с разрешения правообладателя.

Оглавление

Предисловие	11
Введение	14
Для кого предназначена эта книга	14
Чему вы научитесь	16
Условные обозначения, используемые в этой книге	17
Использование примеров кода	18
Онлайн-обучение в O'Reilly	18
Как с нами связаться	19
Благодарности	19

ЧАСТЬ I. ПУТЬ К НАБЛЮДАЕМОСТИ

ГЛАВА 1. Что такое наблюдаемость?	23
Математическое определение наблюдаемости	23
Наблюдаемость в контексте программных систем	24
Неверное толкование наблюдаемости в контексте программного обеспечения	28
Почему наблюдаемость актуальна	29
Действительно ли это лучший способ?	30
Почему метрик и мониторинга недостаточно?	30
Отладка с использованием метрик и наблюдаемости. Различия	32
Роль кардинальности	35
Роль размерности	36
Отладка в условиях наблюдаемости	38
Наблюдаемость предназначена для современных систем	39
Выводы	40
ГЛАВА 2. Наблюдаемость и мониторинг: как отличаются методы отладки	41
Как данные мониторинга используются для отладки	41
Устранение неполадок при использовании дашбордов	44
Ограничения в устранении неполадок с помощью интуиции	45
Традиционный мониторинг по своей сути является реакционным	47
Как наблюдаемость помогает при отладке	49
Выводы	51
ГЛАВА 3. Масштабирование приложения без обеспечения наблюдаемости.	
Уроки	53
Знакомство с Parse	53
Масштабирование в Parse	55
Эволюция в сторону современных систем	58
Переход к современным практикам	62
Изменение методов работы в Parse	64
Выводы	67

ГЛАВА 4. Как наблюдаемость соотносится с DevOps, SRE и Cloud Native	69
Кратко о DevOps, SRE и нативных облачных технологиях	69
Наблюдаемость: отладка тогда и сейчас	71
Наблюдаемость расширяет возможности DevOps и SRE-практик	73
Выводы	74

ЧАСТЬ II. ОСНОВЫ НАБЛЮДАЕМОСТИ

ГЛАВА 5. Структурированные события — это основополагающий компонент наблюдаемости	77
Отладка с использованием структурированных событий	78
Недостатки метрик как основного компонента	79
Ограничения традиционного метода логов как основополагающего компонента	81
Неструктурированные логи	82
Структурированные логи	83
Свойства событий, полезные при отладке	84
Вывод	86
ГЛАВА 6. Объединение событий в трассировки	88
Что такое распределенная трассировка и почему она сейчас необходима	88
Компоненты трассировок	90
Инструментирование трассировки. Пишем код сами	92
Добавление пользовательских полей в интервалы трассировки	96
Объединение событий в трассировки	98
Выводы	99
ГЛАВА 7. Инструментирование OpenTelemetry	100
Краткое знакомство с инструментированием	101
Открытые стандарты инструментирования	101
Инструментирование на примерах с кодом	102
Начинаем с автоматического инструментария	104
Добавление пользовательских инструментов	106
Отправка инструментальных данных в бэкенд	109
Выводы	111
ГЛАВА 8. Анализируем события для достижения наблюдаемости	112
Отладка на основе известных условий	113
Отладка по методу первых принципов	115
Использование цикла основного анализа	116
Автоматизация метода перебора в цикле основного анализа	117
Ложные обещания AIOps	121
Выводы	123
ГЛАВА 9. Как сочетаются наблюдаемость и мониторинг	124
Для чего подходит мониторинг	124
Для чего подходит наблюдаемость	126
Соотношение системы и программного обеспечения	127
Определение потребностей вашей организации	129
Исключение — это мониторинг инфраструктуры, который нельзя игнорировать	131
Примеры из жизни	132
Выводы	133

ЧАСТЬ III. НАБЛЮДАЕМОСТЬ ДЛЯ КОМАНД

ГЛАВА 10.

Применение методов наблюдаемости в вашей команде	137
Присоединяйтесь к группе сообщества	137
Начните с самых болезненных точек	139
Покупайте, а не создавайте	140
Дополняйте инструментирование постепенно	142
Ищите возможности для использования уже затраченных усилий	143
Приготовьтесь к самому сложному последнему рывку	145
Выводы	146

ГЛАВА 11. Разработка на основе наблюдаемости

Разработка через тестирование	148
Наблюдаемость в цикле разработки	150
Где проводить отладку	150
Отладка времени работы микросервисов	152
Как инструментирование помогает обеспечивать наблюдаемость	153
Внедрение наблюдаемости на более ранних стадиях жизненного цикла	154
Использование наблюдаемости для ускорения доставки программного обеспечения	155
Выводы	157

ГЛАВА 12. Использование целей уровня обслуживания для обеспечения надежности

Как при использовании традиционных методов мониторинга возникает опасная усталость от оповещений	158
Оповещение по пороговым значениям предназначено только для известных неизвестных	160
Пользовательский опыт — это Полярная звезда	163
Что такое цель уровня обслуживания?	164
Надежное оповещение на основе SLO	165
Как изменить культуру и начать использовать систему оповещений на основе SLO. Пример из практики	168
Выводы	171

ГЛАВА 13. Действия при получении SLO-оповещений и отладка

Оповещение перед исчерпанием бюджета ошибок	172
Рамки времени для скользящего окна	174
Прогнозирование для создания предупреждающего оповещения об исчерпании	176
Прогнозное окно	178
Окно базовой линии	186
Действия при срабатывании SLO-оповещения о расходовании бюджета	187
Что лучше для SLO: данные наблюдаемости или временных рядов	189
Выводы	192

ГЛАВА 14. Наблюдаемость и цепочка поставок программного обеспечения

Почему для Slack нужна наблюдаемость	196
Инструментирование: общие клиентские библиотеки и измерения	197
Введение в действие цепочки поставок. Тематическое исследование	201

Понимание контекста с помощью инструментов	202
Внедрение требующих реагирования оповещений	204
Как узнать, что изменилось	206
Выводы	207

ЧАСТЬ IV. НАБЛЮДАЕМОСТЬ В МАСШТАБЕ

ГЛАВА 15. Разрабатывать самим или покупать. Прибыль на инвестиции	211
Как проанализировать рентабельность инвестиций в наблюдаемость	212
Реальные затраты на самостоятельную разработку	213
Скрытые издержки использования «бесплатного» программного обеспечения	214
Преимущества самостоятельной разработки	215
Риски, связанные с самостоятельной разработкой	216
Реальные затраты на покупку программного обеспечения	218
Скрытые финансовые затраты коммерческого программного обеспечения	218
Скрытые нефинансовые издержки коммерческого программного обеспечения	220
Преимущества покупки коммерческого программного обеспечения	221
Риски, связанные с покупкой коммерческого программного обеспечения	222
Покупка или разработка — это не двоичный выбор	222
Выводы	224
ГЛАВА 16. Эффективное хранение данных	226
Функциональные требования наблюдаемости	226
Базы данных временных рядов недостаточны для обеспечения наблюдаемости	228
Другие хранилища данных	230
Стратегии хранения данных	232
Реализация Honeyscomb Retriever	236
Разделение данных по времени	236
Хранение данных по столбцам в сегментах	238
Выполнение запросов	240
Запрос трассировок	242
Запрос данных в режиме реального времени	243
Оптимизация стоимости благодаря уровневой структуре	243
Ускоряем это с помощью параллелизма	244
Работа с высокой кардинальностью	245
Стратегии масштабирования и долговечности	246
Рекомендации по созданию эффективного хранилища данных	248
Выводы	249
ГЛАВА 17. Экономичный и достаточно точный метод — выборка	250
Оптимизация сбора данных с помощью выборки	250
Разные методы выборки	252
Выборка с постоянной вероятностью	252
Выборка по недавнему объему трафика	253
Выборка на основе содержания событий (ключей)	254

Методы комбинации ключей и метод исторических данных	254
Параметры динамической выборки	255
Когда следует проводить выборку для трассировок	255
Программирование стратегий выборки	256
Базовый вариант	256
Выборка с фиксированной частотой	257
Запись частоты выборки	257
Последовательная выборка	259
Запланированная частота выборки	260
Когда у вас несколько статичных частот выборки	262
Выборка по ключу и запланированная частота	263
Выборка с динамической частотой по произвольному количеству ключей	265
Объединяем начальные и итоговые данные для определения целевой частоты выборки по ключам	267
Выводы	268
ГЛАВА 18. Управление телеметрией с помощью конвейеров	270
Атрибуты конвейеров телеметрии	272
Маршрутизация	272
Безопасность и соответствие	272
Изоляция рабочих нагрузок	273
Буферизация данных	273
Управление емкостью	274
Фильтрация и дополнение данных	275
Преобразование данных	276
Обеспечение качества и согласованности данных	276
Управление конвейером телеметрии: анатомия	277
Сложности в управлении конвейером телеметрии	279
Производительность	280
Корректная работа	280
Доступность	280
Надежность	280
Изоляция	281
Свежесть данных	281
Управление телеметрией. Пример использования в Slack	282
Агрегация метрик	282
Логи и события трассировок	283
Альтернативы с открытым исходным кодом	286
Управление конвейером телеметрии: создать самим или купить	287
Выводы	288

ЧАСТЬ V. РАСПРОСТРАНЕНИЕ КУЛЬТУРЫ НАБЛЮДАЕМОСТИ

ГЛАВА 19. Экономическое обоснование наблюдаемости	291
Реактивный подход к внедрению изменений	291
Окупаемость инвестиций в наблюдаемость	294
Проактивный подход к внедрению изменений	295

10 Оглавление

Внедрение наблюдаемости как практики	297
Использование соответствующих инструментов	299
Инструментирование	299
Хранение данных и аналитика	300
Постепенное внедрение новых инструментов в работу команды	301
Как понять, что наблюдаемость обеспечена на достаточном уровне	301
Выводы	303
ГЛАВА 20. Заинтересованные стороны и союзники наблюдаемости	304
Какие потребности в наблюдаемости есть у неинженеров	304
Как приобрести союзников по наблюдаемости	307
Команды поддержки клиентов	308
Отдел заботы о клиентах и продуктовые команды	309
Отделы продаж и руководители	310
Использование инструментов наблюдаемости или бизнес-аналитики	311
Время выполнения запроса	312
Точность	312
Свежесть данных	313
Структура	313
Временные окна	314
Эфемерность	314
Совместное использование инструментов наблюдаемости и бизнес-аналитики на практике	315
Выводы	316
ГЛАВА 21. Модель зрелости наблюдаемости	317
Примечание о моделях зрелости	317
Почему для наблюдаемости необходима модель зрелости	318
О модели зрелости для практики наблюдаемости	320
Способности, описанные в ООМ	321
Устойчивое реагирование на системные сбои	322
Создание высококачественного кода	324
Управление сложностями и техническим долгом	325
Релизы с предсказуемой частотой	326
Понимание поведения пользователя	327
Использование ООМ в вашей организации	328
Выводы	329
ГЛАВА 22. Как развиваться дальше	331
Наблюдаемость в прошлом и сейчас	331
Дополнительные ресурсы	333
Прогнозы относительно того, как будет развиваться практика наблюдаемости	335
Алфавитный указатель	340
Об авторах	348
Послесловие	349

Предисловие

За последние пару лет термин «наблюдаемость» перешел из употребления в узкоспециализированном сообществе системной инженерии в обиходный язык разработчиков программного обеспечения. По мере распространения термина его также постигла (увы, неизбежно) участь взаимозаменяемости с другим смежным названием — «мониторингом».

И тогда возникли неизбежные и досадные последствия: создатели и поставщики инструментов мониторинга взяли на вооружение тот же самый язык и лексику, что и те, кто пытается провести различие между философскими, техническими и социотехническими основами наблюдаемости и мониторинга. Такая неразбериха, мягко говоря, не принесла пользы. Здесь может произойти отождествление понятий «наблюдаемость» и «мониторинг», что еще больше затруднит проведение содержательных и подробных разговоров об их различиях.

Рассматривать разницу между мониторингом и наблюдаемостью как чисто семантическую — это безумие. Наблюдаемость — это не чисто техническое решение, которое можно получить, купив «инструмент для обеспечения наблюдаемости» (что бы там ни говорил поставщик) или приняв открытый современный стандарт. Напротив, наблюдаемость — это больше социотехническое понятие. Успешное внедрение наблюдаемости в такой же степени, если не в большей, зависит от наличия соответствующих культурных основ для поддержки методов разработки, развертывания, отладки и сопровождения программного обеспечения, а также от наличия подходящего инструмента для его удаления.

В большинстве (возможно, даже во всех) сценариев командам необходимо использовать и мониторинг, и наблюдаемость для успешного создания и эксплуатации своих сервисов. Но любая такая успешная реализация требует, чтобы практики сначала осознали философские различия между ними.

Мониторинг от наблюдаемости отличает пространство состояний поведения системы, а также то, как можно исследовать его и на каком именно уровне детализации. Под «пространством состояний» я подразумеваю все возможные варианты поведения, которые система может демонстрировать

на различных этапах: начиная с момента проектирования до разработки, тестирования, развертывания, предоставления пользователям и до отладки в течение всего срока службы. Чем сложнее система, тем более обширным и многообразным становится пространство состояний.

Наблюдаемость позволяет составить тщательную схему этого пространства состояний и исследовать его в мельчайших деталях вдоль и поперек. Такое кропотливое исследование часто нужно для лучшего понимания непредсказуемых, длинных или мультимодальных распределений в поведении системы. Мониторинг, напротив, позволяет получить приблизительное представление об общем состоянии системы в крупных чертах.

Из этого следует, что все, начиная от данных, которые собираются с этой целью, и заканчивая тем, как они хранятся и как анализируются для лучшего понимания поведения системы, варьируется в зависимости от целей мониторинга и наблюдаемости.

За последние пару десятилетий идеалы мониторинга повлияли на разработку множества инструментов, систем, процессов и практик, многие из которых де-факто стали отраслевыми стандартами. Поскольку эти инструменты, системы, процессы и методы были разработаны специально для мониторинга, они превосходно справляются с этой задачей. Однако они не могут — и не должны — подвергаться ребрендингу или продаваться ничего не подозревающим клиентам в качестве инструментов или процессов для обеспечения «наблюдаемости». Это принесло бы мало ощутимой пользы, зато появился бы риск потери огромного количества времени, усилий и денег для клиента.

Более того, инструменты — это только одна часть проблемы. Создание или внедрение инструментов и практик обеспечения наблюдаемости, которые доказали свою эффективность в других компаниях, не обязательно решит все проблемы, возникшие в вашей организации. Готовый продукт не рассказывает о том, как развивались инструменты и сопутствующие процессы, какие комплексные проблемы он решал, какие неявные допущения были заложены в продукт и многое другое.

Разработка или покупка правильного инструмента наблюдаемости не станет панацеей, если заранее не созданы благоприятные культурные условия внутри компании, которая настраивает команды на успех. Мышление и культура, основанные на штампах мониторинга — дашбордах, оповещениях, статических пороговых значениях, — не помогают полностью раскрыть потенциал наблюдаемости. Инструменты наблюдаемости могут иметь доступ к очень большому объему подробных данных, но для успешного анализа такой горы данных — а это главный критерий общей жизнеспособности и полезности инструмента и, возможно, самой наблюдаемости! — требуется мышление с использованием гипотез и итеративной отладки.

Простой доступ к самым современным инструментам не культивирует такое мышление у практикующих специалистов автоматически. И не способствует этому красноречивое воспевание туманных философских различий между мониторингом и наблюдаемостью, когда эти идеи не превратились в сквозные практические решения. Например, в этой книге есть главы, в которых авторы критически высказываются о том, что логи, метрики и трассировки — это «три столпа наблюдаемости». Такая критика не лишена оснований: логи, метрики и трассировки долгое время были единственными конкретными примерами телеметрии, которые специалисты имели в своем распоряжении для отладки реальных систем в реальном мире, и поэтому вокруг них неизбежно возникла история о «трех столпах».

То, что лучше всего подходит практикам, создающим системы в реальном мире, — это не абстрактные, далекие от жизни идеи, а практическая схема, рассматривающая и предлагающая решения насущных технических и культурных проблем, с которыми они сталкиваются. Этой книге удалось перекинуть мост через пропасть, которая зияет между философскими принципами наблюдаемости и их практическим применением, предлагая конкретную (хотя и категоричную) схему того, как может выглядеть применение этих идей на практике.

Вместо того чтобы сосредотачиваться на протоколах, стандартах или даже низкоуровневых представлениях различных телеметрических сигналов, книга рассматривает три столпа наблюдаемости как триаду из структурированных событий (или трассировку без контекстного поля, как я люблю их называть), итеративной проверки гипотез (или отладку на основе гипотез, как я люблю это называть) и «цикла основного анализа». Эта первоначальное целостное переосмысление основополагающих компонентов наблюдаемости помогает подчеркнуть, что сами по себе телеметрические сигналы (или инструменты для их использования) не делают поведение системы максимально наблюдаемым. Эта книга не обходит стороной проблемы, с которыми могут столкнуться команды при внедрении культуры наблюдаемости, и дает ценные рекомендации об использовании наблюдаемости на постоянной основе, что сослужит хорошую службу специалистам в долгосрочной перспективе.

— Синди Шридхаран, инженер по инфраструктуре
Сан-Франциско, 26 апреля 2022 года

Введение

Благодарим вас за то, что вы выбрали нашу книгу по обеспечению наблюдаемости для современных программных систем. Наша цель — помочь вам внедрить практику обеспечения наблюдаемости в вашей инженерной организации. Эта книга основана на нашем практическом опыте использования инструментов наблюдаемости и их разработки для пользователей, которые хотят улучшить свои методы работы в этой области.

Как решительные сторонники внедрения методов наблюдаемости в программную инженерию, мы надеемся, что эта книга сможет дать четкое представление о том, что означает наблюдаемость в контексте программных систем. Термин «наблюдаемость» в последнее время довольно широко используется в экосистеме разработки программного обеспечения. Цель этой книги — помочь вам отделить факты от шумихи и тщательно проанализировать:

- что означает наблюдаемость в контексте доставки и эксплуатации программного обеспечения;
- как собрать основополагающие компоненты, которые помогут вам достичь наблюдаемости;
- как наблюдаемость влияет на динамику команды;
- как обеспечить наблюдаемость при масштабировании;
- практические способы создания культуры наблюдаемости в вашей организации.

Для кого предназначена эта книга

Наблюдаемость в основном нацелена на то, чтобы лучше понимать, как программное обеспечение работает в реальном мире. Поэтому эта книга наиболее полезна инженерам-программистам, ответственным за разработку приложений для продакшна*. Однако любой, кто поддерживает работу продакшна программного обеспечения, тоже найдет много полезного в этой книге.

* Продакшн (production — *англ.*) — рабочая версия системы, доступная для пользователей. — *Прим. пер.*

Кроме того, руководители команд по поставке и эксплуатации программного обеспечения, которым интересно разобраться в том, как наблюдаемость может принести пользу их организации, тоже оценят эту книгу, особенно в главах, посвященных динамикам в командах, культуре и масштабированию.

Все, кто помогает командам разрабатывать и эксплуатировать продакшн программного обеспечения (например, продакт-менеджер, инженер технической поддержки и заинтересованные стороны), и все, кому интересно узнать об этом новом понятии — «наблюдаемости» — и о том, почему люди говорят о нем, тоже найдут эту книгу полезной для себя.

Почему мы написали эту книгу

Наблюдаемость уже стала популярной темой; она быстро привлекла к себе большой интерес и внимание. С ростом популярности термин стали неправильно употреблять как синоним «мониторинга» или «системной телеметрии». Наблюдаемость — это характеристика программных систем. Кроме того, это свойство может эффективно использоваться только в продакшне программных систем, когда команды внедряют новые практики, которые поддерживают непрерывную разработку. Таким образом, внедрение наблюдаемости в ваши системы — это одновременно техническая и культурная задача.

Мы относимся к теме наблюдаемости с особым трепетом и много о ней говорим. Мы настолько увлечены этим, что основали компанию, единственная цель которой — донести возможности наблюдаемости до всех команд, управляющих продакшном программного обеспечения. Мы запустили новую категорию инструментов наблюдаемости, и другие поставщики последовали нашему примеру.

Хотя мы все работаем на `Noneusomb`, мы написали эту книгу не с целью продать вам наши инструменты. Мы хотим объяснить, как и почему мы адаптировали первоначальную концепцию наблюдаемости для управления современными программными системами. Вы можете добиться наблюдаемости с помощью разных инструментов и разными способами. Однако мы считаем, что наша приверженность продвижению практики наблюдаемости в индустрии программного обеспечения делает нас обладателями уникальной квалификации для написания руководства, в котором очень подробно описываются общие проблемы и эффективные решения. При создании продакшна программных систем с хорошей наблюдаемостью вы можете применять концепции, изложенные в этой книге, независимо от выбранного вами инструмента.

Цель этой книги — дать вам представление о различных мнениях, возможностях и сложностях, которые касаются команд, практикующих наблюдаемость для управления продакшна программных систем. Иногда в качестве

примера решения общей проблемы мы можем рассказать о том, что делает Honeycomb. Мы это делаем не для рекламы Honeycomb, а для практической иллюстрации абстрактных концепций. Наша цель — показать вам, как применять те же самые принципы в другом окружении, независимо от используемых вами инструментов.

Чему вы научитесь

Вы узнаете, что такое наблюдаемость, как определить наблюдаемую систему и почему наблюдаемость лучше всего подходит для управления современными программными системами. Вы узнаете, чем наблюдаемость отличается от мониторинга, а также почему и когда необходимо применять другой подход. Мы также рассмотрим, почему отраслевые тенденции помогли популяризировать потребность в наблюдаемости и как это вписывается в новые пространства, такие как нативные облачные экосистемы.

Кроме того, мы опишем основы наблюдаемости. Мы рассмотрим, почему структурированные события являются основополагающими компонентами наблюдаемых систем и как объединить эти события в трассировках. События генерируются с помощью телеметрии, встроенной в ваше программное обеспечение. Вы узнаете о первых программах с открытым исходным кодом, таких как OpenTelemetry, которые помогают активизировать процесс инструментирования. Вы узнаете о процессе расследования на основе данных, который используется для выявления источника проблем в наблюдаемых системах, и о том, чем он отличается от интуитивного расследования в традиционном мониторинге. Вы также узнаете, как могут сосуществовать наблюдаемость и мониторинг.

После ознакомления с этими фундаментальными техническими концепциями вы узнаете о социальных и культурных элементах, которые часто сопровождают внедрение наблюдаемости. Управление программным обеспечением в продакшне — это командный вид спорта, и вы узнаете, как наблюдаемость помогает улучшать командные динамики. Вы узнаете о том, как наблюдаемость вписывается в бизнес-процессы, влияет на цепочку поставок программного обеспечения и выявляет скрытые риски. Вы также познакомитесь с тем, как применять на практике эти технические и социальные концепции, когда мы будем рассматривать использование целей уровня обслуживания для более эффективного оповещения и углубимся в технические детали, объясняющие почему они делают оповещения более действенными и отлаживаемыми при использовании данных наблюдаемости.

Далее вы узнаете о характерных проблемах, возникающих при масштабном внедрении решений для обеспечения наблюдаемости. Мы начнем с факторов, которые вам следует учитывать при принятии решения

о покупке инструмента обеспечения наблюдаемости или о его создании. Существенным свойством решений по обеспечению наблюдаемости является то, что они должны давать быстрые ответы в ходе итеративных исследований. Поэтому мы покажем вам, как решить неизбежные проблемы, связанные с эффективным хранением и извлечением данных при управлении чрезвычайно большими массивами данных. Вы также узнаете, когда следует внедрять такие решения, как выборка событий, и как найти правильный подход, соответствующий вашим потребностям. Вы также узнаете, как управлять чрезвычайно большими объемами данных с помощью конвейеров телеметрии.

Наконец, мы рассмотрим организационные подходы к внедрению культуры наблюдаемости. Помимо того, как познакомить вашу команду с наблюдаемостью, вы узнаете практические способы масштабирования этой практики во всей организации. Вы узнаете, как выявлять ключевые заинтересованные стороны и работать с ними, использовать технические подходы для привлечения союзников и обосновывать внедрение методов наблюдаемости.

Мы начали писать эту книгу почти три года назад. Одна из причин, по которой у нас ушло на это так много времени, заключается в том, что область знаний по наблюдаемости быстро менялась, а методы улучшались. Мы считаем, что эта книга — наиболее современный и всеобъемлющий обзор существующих методов наблюдаемости на момент ее публикации. Мы надеемся, что это путешествие покажется вам таким же увлекательным, как и нам.

Условные обозначения, используемые в этой книге

В этой книге используются следующие типографские условные обозначения:

Курсив

Применяется для новых терминов, URL-адресов, адресов электронной почты, имен и расширений файлов.

Моноширинный шрифт

Используется при перечислении программ, а также в параграфах для обозначения элементов программы, таких как имена переменных или функций, базы данных, типы данных, переменные среды, инструкции и ключевые слова.



Этот элемент обозначает общее замечание.