

Нужно мыслить целостно

Наконец, реальным препятствием, которое необходимо преодолеть, является сложность сценариев, подлежащих автоматизации. Прежде чем работать над автоматизацией, пересмотрите подлежащую обновлению систему — сможете ли вы сначала упростить ее или рабочий процесс?

Обращайте внимание на все особенности автоматизированного рабочего процесса, а не только на те, которые создают трудности для вас лично. Проводите тестирование с непосредственно вовлеченными в работу людьми и активно обращайтесь к ним за советами и помощью. Если они ошибаются, узнайте, как сделать ваш пользовательский интерфейс лучше и понятнее и какие дополнительные проверки безопасности понадобятся. Убедитесь, что ваша автоматизация не приводит к появлению дополнительной рутины, например к созданию излишних заданий, требующих реакции человека.

Пример 2. Вывод из эксплуатации домашних каталогов, использующих файловые серверы



Стратегии сокращения рутинной работы, выделенные в примере 2.

- Рассмотрите вопрос о выводе из эксплуатации устаревших систем.
- Продвигайте сокращение рутины как новую функцию.
- Заручитесь поддержкой от руководства и коллег.
- Откажитесь от рутинной работы.
- Начните с интерфейсов, поддерживаемых человеком.
- Предоставьте методы самообслуживания.
- Начните с малого, а затем улучшайте.
- Используйте обратную связь для улучшения.

Вводная информация

В первые дни существования Google команда SRE Corp Data Storage (CDS) предоставляла домашние каталоги всем сотрудникам компании. Подобно решению Active Directory Roaming Profiles, обычному для корпоративных ИТ, работники Google могли использовать на всех рабочих станциях и платформах одни и те же домашние каталоги. Команда CDS также предоставляла Team Shares (командные разделяемые ресурсы) для совместной работы в общем пространстве хранения. Мы предоставляли домашние каталоги и Team Shares через парк файл-серверов,

или файловиков (filers), — устройств хранения Netapp Storage Appliances — и протоколы сетевых файловых систем NFS/CIFS. Это хранилище было дорогим в эксплуатации, но предоставляло столь необходимую для Google услугу.

Постановка задачи

По прошествии многих лет эти решения в своем большинстве устарели по сравнению с другими более совершенными решениями для хранения данных, такими как системы контроля версий (Piper¹/Git-on-borg²), Google Drive, Google Team Drive, Google Cloud Storage и внутренняя, общая, глобально распределенная файловая система под названием x20. Эти альтернативы были лучше по ряду причин.

- ❑ Протоколы NFS/CIFS никогда не предназначались для работы в Глобальной сети, поэтому пользовательское взаимодействие быстро ухудшалось даже при задержке несколько десятков миллисекунд. Это создавало проблемы для удаленных работников или глобально распределенных групп, поскольку данные могли находиться только в одном месте.
- ❑ Эти устройства, по сравнению с альтернативными, были дорогими в плане эксплуатации и масштабирования.
- ❑ Чтобы совместить протоколы NFS/CIFS с моделью сетевой безопасности Google Beyond Corp³, потребовалось бы приложить значительные усилия.

Наиболее же важным относительно данной главы было то, что работа с домашними каталогами и Team Shares порождала много рутины. Во многом работа с хранилищами управлялась с помощью тикетов. Хотя эти рабочие процессы были частично заскриптованы, они представляли собой значительный кусок работы команды CDS. Мы тратили много времени на создание и настройку общих ресурсов, изменение доступа, устранение неполадок у конечных пользователей и наращивание или сокращение емкости. CDS в дополнение к конфигурированию устройств, обнов-

¹ Piper — внутренняя система контроля версий Google. Для получения дополнительной информации см.: *Potvin R., Levenberg J.* Why Google Stores Billions of Lines of Code in a Single Repository («Почему Google хранит миллиарды строк кода в одном репозитории») // Communications of the ACM, 59. — 2016, № 7. P. 78–87. <http://bit.ly/2J4jgMi>.

² У Google также есть самообслуживаемый масштабируемый Git-хостинг для кода, работающий вне Piper.

³ Beyond Corp (<https://cloud.google.com/beyondcorp/>) — инициатива по переходу от традиционной модели «периметра безопасности» к криптографической модели, основанной на идентификации. Когда ноутбук Google подключается к внутреннему сервису Google, сервис проверяет источник с помощью комбинации криптографического сертификата, идентифицирующего ноутбук, второго фактора, принадлежащего пользователю (например, ключа безопасности USB), конфигурации/состояния клиентского устройства и учетных данных пользователя.

лениям и резервному копированию управляла процессами подготовки, установки и прокладки кабелей для этого специализированного оборудования. Из-за требований к задержке нам часто приходилось выполнять развертывание непосредственно в удаленных офисах вместо дата-центров Google, и иногда сотрудники команды вынуждены были ездить в разные уголки города или страны для управления развертыванием.

Что мы решили сделать

Во-первых, мы собрали сведения: команда CDS создала инструмент под названием Moonwalk, который позволил провести анализ того, как сотрудники использовали наши сервисы. Мы собрали традиционные показатели бизнес-аналитики, например сведения о ежедневных активных пользователях (DAU) и ежемесячных активных пользователях (MAU); задали такие вопросы: «Представители каких профессий на самом деле используют свои домашние каталоги?» и «К какого рода файлам чаще всего обращаются пользователи, которые ежедневно пользуются ими?». Moonwalk в сочетании с опросами пользователей подтвердил, что бизнес-потребности, в настоящее время обслуживаемые файл-серверами, могут лучше обслуживаться альтернативными решениями с меньшими эксплуатационными и накладными расходами и меньшей стоимостью. Еще одна убедительная причина заставила нас отойти от файл-серверов: если бы мы могли перенести большинство наших сценариев использования файл-серверов в G Suite/GCP, то смогли бы улучшить эти продукты, тем самым позволяя и другим крупным предприятиям перейти на G Suite/GCP.

Ни одна альтернатива не реализовала полностью все текущие сценарии использования домашних каталогов. Однако, разбив задачу на более мелкие подзадачи, которые можно было решать по отдельности, мы обнаружили, что комбинация нескольких альтернатив позволяет охватить все наши сценарии. Альтернативные решения были более специализированными, но каждое из них обеспечивало лучшие потребительские качества, чем единое решение с файл-сервером.

- ❑ *x20*¹ — предоставляла отличный способ для команд глобально делиться статическими объектами, например двоичными файлами.
- ❑ *G Suite Team Drive* (<http://bit.ly/2soMbjQ>) — хорошо подходил для совместной работы над офисными документами и был гораздо более терпимым к пользовательской задержке, чем NFS.
- ❑ *Файловая система Google Colossus* — позволяет командам обмениваться большими файлами данных более безопасно и масштабируемо, чем NFS.

¹ *x20* — это внутренняя глобально разделяемая файловая система высокой доступности с семантикой файловой системы POSIX.

- ❑ *Piper/Git-on-Borg* — давала возможность лучше синхронизировать dotfiles (персонализированные настройки инструментов инженеров).
- ❑ *Новый инструмент «история как услуга»* — позволял сделать историю командной строки разделяемой между рабочими станциями¹.

По мере того как мы каталогизировали варианты использования и находили альтернативы, план вывода файл-серверов из эксплуатации обретал форму.

Проектирование и реализация

Отказ от файл-серверов был непрерывным, итеративным, многолетним усилием, которое привело к созданию нескольких внутренних проектов.

- ❑ *Moira*. Вывод прежней версии домашнего каталога из эксплуатации.
- ❑ *Tekmor*. Перенос большого списка пользователей домашнего каталога.
- ❑ *Migra*. Вывод из эксплуатации Team Share.
- ❑ *Azog*. Удаление инфраструктуры и связанного оборудования домашнего каталога и Team Share.

В этом практическом примере мы сосредоточимся на первом проекте, *Moira*. Последующие проекты основывались на том, что мы узнали и создали для *Moira*.

Как показано на рис. 6.8, *Moira* состоял из четырех этапов.

Первый шаг к отказу от устаревшей системы — остановить или (что более реалистично) замедлить ее дальнейшее внедрение. Ведь гораздо тяжелее что-то у пользователей отнять, чем никогда этого не предлагать. Данные *Moopwalk* показали, что сотрудники Google, не работающие инженерами, меньше всего использовали свои общие домашние каталоги. Поэтому на начальном этапе мы ориентировались на этих пользователей. По мере того как рос объем задач в рамках каждого этапа, росла и наша уверенность в альтернативных решениях для хранения, в процессе миграции и в инструментах. На каждом этапе проекта имелся соответствующий проектировочный документ, в котором предлагаемое решение рассматривалось по таким параметрам, как безопасность, масштабируемость, тестирование и запуск. Мы также уделили особое внимание взаимодействию с пользователями, их ожиданиям и общению. Наша цель состояла в том, чтобы на каждом этапе пользователи, затронутые изменениями, понимали как смысл этого проекта по выводу из эксплуатации, так и самый простой способ архивирования и переноса своих данных на новую систему.

¹ История командной строки — команды, которые набирал и выполнял пользователь через командную строку. Имеется в виду хранение и доступ с различных рабочих станций с привязкой к конкретному пользователю, а не к машине. — *Примеч. науч. ред.*

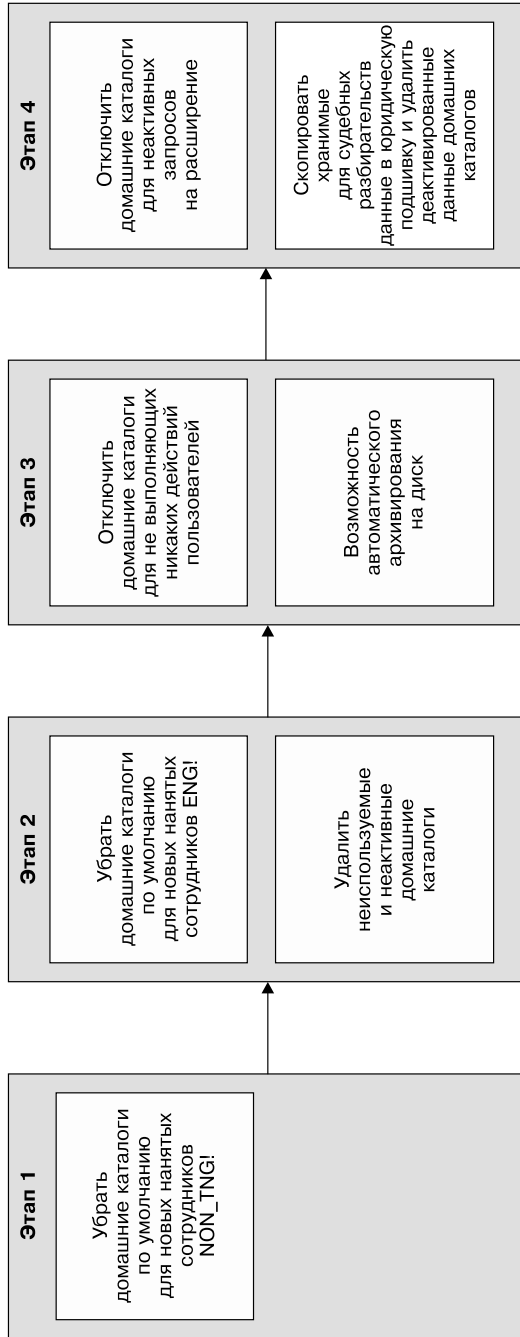


Рис. 6.8. Четыре этапа проекта Moiga

Ключевые компоненты

Moopwalk

Хотя у нас были основные статистические данные о каталогах, с которыми совместно работали наши пользователи (например, их размеры), нам нужно было понять рабочие процессы пользователей. Это позволило бы управлять бизнес-решениями при выводе системы из эксплуатации. Мы создали систему под названием Moopwalk, которая собирала и представляла отчеты с этой информацией.

Moopwalk хранила данные о том, кто, когда и к каким файлам обращался, в BigQuery. Это позволило нам создавать отчеты и выполнять специальные запросы, чтобы лучше понимать пользователей. С помощью BigQuery мы выявили паттерны доступа к 2,5 миллиарда файлов, занимающим 300 Тбайт дискового пространства. Эти данные принадлежали 60 000 пользователей POSIX в 400 дисковых томах на 124 устройствах NAS в 60 географических точках по всему миру.

Портал Moira

Из-за объема нашей пользовательской базы оказалось, что не имеет смысла даже пробовать сделать процесс управления списанием домашних каталогов ручным на основе тикетов. Нам нужно было сделать весь процесс — опрос пользователей, оповещение о причинах и целях проекта по выводу из эксплуатации и сопровождение пользователей при архивировании их данных или переходе на альтернативную систему — как можно более простым. Наши окончательные требования были такими:

- ❑ наличие домашней страницы с описанием проекта;
- ❑ постоянно обновляемый FAQ;
- ❑ информация о состоянии и использовании папки совместного доступа текущего пользователя;
- ❑ возможности запроса, деактивации, архивирования, удаления, расширения или повторной активации папки совместного доступа.

Наша бизнес-логика стала довольно сложной, потому что нам пришлось учитывать ряд пользовательских сценариев. Например, пользователь мог покинуть Google, уйти во временный отпуск или его данные могли требоваться для судебного разбирательства. На рис. 6.9 приведен пример диаграммы состояния из проектной документации, иллюстрирующий эту сложность.

Технология, приводящая портал в действие, относительно проста. Будучи разработанным на Python с использованием Flask, он записывал в Bigtable и читал из него. Кроме того, для управления его работой использовался ряд фоновых задач и планировщиков.

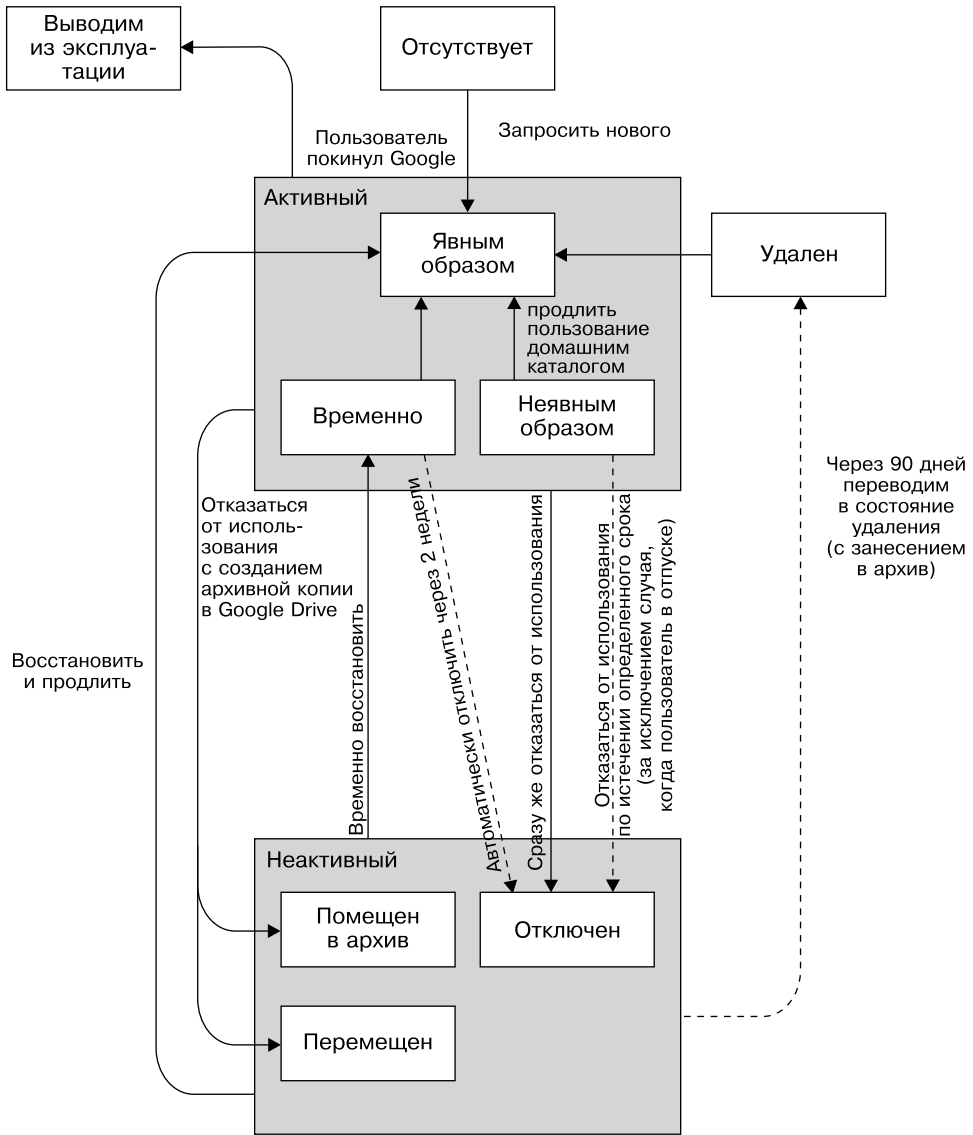


Рис. 6.9. Бизнес-логика на основе пользовательских сценариев

Автоматизация архивации и переноса

Нам потребовалось много вспомогательных инструментов, чтобы связать портал и управление конфигурацией, а также опрашивать пользователей и общаться с ними. Нам также нужно было убедиться, что мы правильно идентифицируем

пользователей для того или иного сообщения или действия. Ложные срабатывания (ошибочное сообщение о требуемых действиях) или ложные пропуски (неспособность уведомить пользователя о том, что мы у него что-то забираем) были недопустимы, а ошибки означали бы дополнительную работу, связанную с утратой доверия и необходимостью дополнительного обслуживания пользователя.

Мы работали с владельцами альтернативных систем хранения над тем, чтобы добавить некоторый функционал в их планы (дорожные карты). Как результат, в ходе развития этого проекта более пригодными для воспроизведения вариантов использования файл-серверов оказались менее зрелые (и потому менее консервативные) системы. У нас также была возможность использовать и расширять набор инструментов других команд. Например, для реализации на портале функционала автоматического архивирования мы применяли разработанный другой командой инструмент для переноса данных из Google Cloud Storage в Google Drive.

Все это требовало значительных усилий по разработке программного обеспечения в течение всего жизненного цикла проекта. В ответ на требования следующего этапа и отзывы пользователей мы настроили и перебрали каждый компонент — конвейер отчетов Moonwalk, портал и систему автоматизации — для лучшего управления изъятием из обращения и архивацией папок совместного доступа. Мы приблизились к завершающему состоянию только на третьем этапе (почти два года); и даже тогда нам потребовались дополнительные инструменты для обработки «хвоста» длиной около 800 пользователей. Таким образом, запрягли мы действительно медленно, что имело определенные преимущества и позволило нам:

- ❑ обойтись силами команды небольшого размера (в среднем три члена команды CDS);
- ❑ минимизировать помехи для рабочих процессов пользователей;
- ❑ ограничить рутину на стороне Techstop (внутренняя организация технической поддержки Google);
- ❑ создавать инструменты по мере необходимости, чтобы не тратить впустую время и силы инженеров.

Как и во всех технических решениях, у нас были и некоторые компромиссы: проект был долгосрочным, поэтому, пока решение разрабатывалось, команде приходилось продолжать заниматься привычным обслуживанием файл-серверов.

Программа была официально завершена в 2016 году. На момент написания книги мы сократили число домашних каталогов с 65 000 до 50. (Текущий проект Azog направлен на то, чтобы вывести и этих последних пользователей и полностью вы-

вести из эксплуатации оборудование последних файл-серверов.) Взаимодействие с пользователями улучшилось, а CDS вывела из эксплуатации дорогостоящее оборудование.

Усвоенные уроки

Хотя ни одно альтернативное хранилище файлов не могло полностью заменить используемые сотрудниками Google на протяжении более 14 лет файл-серверы, нам не обязательно нужна была точная и полная замена. По сути, нарастив стек вверх от гибкого и многофункционального, но ограниченного решения уровня файловой системы до нескольких специализированных решений, мы обменяли гибкость на улучшенную масштабируемость, устойчивость к задержкам и безопасность. Команде Moira нужно было предвидеть возможные действия пользователей и рассматривать альтернативы на разных стадиях развития. Нам приходилось управлять ожиданиями в отношении этих альтернатив: в совокупности они могли бы дать пользователю больше возможностей, но реализация всех требований не прошла бы для нас безболезненно. На этом пути мы усвоили следующие уроки по эффективному сокращению рутины.

Пересматривайте предположения и откажитесь от дорогих бизнес-процессов

Бизнес-требования изменяются, и постоянно появляются новые решения, поэтому стоит периодически анализировать трудоемкие бизнес-процессы. Как мы обсуждали в разделе «Стратегии управления рутинной» на с. 151, отказ от рутины (решение не выполнять рутинные задачи) часто является самым простым способом ее устранения, даже если этот подход нельзя реализовать легко и быстро. Подкрепите свое решение пользовательской аналитикой и бизнес-обоснованиями, прежде чем просто сокращать рутину. Основное коммерческое обоснование вывода из эксплуатации файл-серверов сводилось к преимуществам модели безопасности Beyond Corp. Так что, хоть Moira и позволила уменьшить объем рутинной работы команды CDS, для бизнеса гораздо более привлекательным аргументом за вывод из эксплуатации файл-серверов стало то, что это должно было привести к множеству улучшений в области безопасности.

Предоставьте методы самообслуживания

Мы создали специальный портал для Moira (что было довольно дорого), но часто есть и более простые альтернативы. Многие команды в Google настраивают свои сервисы и управляют ими с помощью систем контроля версий, а организационные запросы решают с помощью запросов на принятие изменений (называемых changelists, CL — «списки изменений»). Этот подход практически не требует

участия команды обслуживания, но при этом дает преимущества анализа и проверки кода, а также процессов непрерывного развертывания (continuous deployment) для проверки, тестирования и внедрения изменений конфигурации служебного («внутреннего») сервиса.

Начните с интерфейсов, поддерживаемых человеком

В нескольких случаях команда Moïga использовала подход «скрытый инженер», который сочетал автоматизацию с ручной работой инженеров. Например, для запросов на совместное использование создавались задания (баги) для отслеживания обработки запроса, которые наша автоматизация обновляла на соответствующих стадиях обработки. Система также назначала эти баги конечным пользователям, чтобы они не забывали задействовать общие ресурсы. Тикеты можно использовать как состряпанную на скорую руку реализацию пользовательского интерфейса для автоматизации: они ведут журнал работы, предоставляют обновления заинтересованным сторонам и обеспечивают наличие простого альтернативного плана действий на случай, если автоматизация идет не так, как надо. В нашем случае, если пользователь нуждался в помощи при миграции или автоматизация не могла обработать его запрос, баг автоматически направлялся в очередь, которую SRE обрабатывали вручную.

Стремитесь к единообразию

Для автоматизации важно единообразие. Инженеры Moïga приняли решение переоснастить систему автоматизации таким образом, чтобы либо иметь специальные обработчики для нестандартных вариантов использования разделяемых ресурсов, либо удалять или изменять эти ресурсы так, чтобы они соответствовали ожиданиям инструментария. Это позволило нам приблизиться к автоматизации большинства процессов миграции, которые в результате выполнялись без участия людей.



Интересный факт: в Google практика «изменения реальности», когда нужно соответствовать коду, а не наоборот, называется «покупкой гнома». Эта фраза связана с историей о Froogle — системе для поиска продуктов по онлайн-магазину, существующей с самых первых дней работы компании.

В начале работы Froogle часто возникала серьезная ошибка, влияющая на качество поиска: при поиске по запросу [кроссовки] система предлагала одетого в кроссовки садового гнома как результат поиска высокого ранга. После нескольких неудачных попыток исправить ошибку кто-то заметил, что этот гном был единственным экземпляром на eBay с опцией «Купить сейчас». И тогда они просто купили гнома (рис. 6.10).



Рис. 6.10. Садовый гном, который никак не исчезал

Используйте административные стимулы

Ищите способы, которыми вы сможете подтолкнуть новых пользователей к использованию лучших (и, хотелось бы надеяться, менее трудоемких) альтернатив. В рамках данного подхода Moïra требовала эскалации¹ для запросов на новые совместно используемые ресурсы или квоты, а также поощряла пользователей, которые удалили свои общие ресурсы. Важно также предоставить качественную документацию по настройке сервиса, лучшим подходам по работе с ним и дать понять, когда придет время его использовать. Команды Google часто используют так называемые codelabs (наборы практических заданий и примеров) или cookbooks («книги рецептов»), которые учат пользователей, как настраивать сервис для типовых вариантов применения. В результате большинство пользователей не нуждаются в помощи команды, обслуживающей сервис.

¹ Получения подтверждения на более высоком уровне, что усложняет и замедляет процедуру. — *Примеч. науч. ред.*

Итоги главы

Объем рутины, связанной с запущенным в эксплуатацию сервисом, растет линейно с увеличением его сложности и масштаба. Автоматизация часто является идеальным вариантом устранения рутины и может сочетаться с рядом иных подходов. Даже когда рутинная работа не стоит усилий по полной автоматизации, вы можете уменьшить трудоемкость разработки и эксплуатации с помощью таких стратегий, как частичная автоматизация или изменение бизнес-процессов.

Примеры и методы устранения рутины, описанные в этой главе, могут быть обобщены для работы с множеством других крупномасштабных сервисов. Устранив рутину, можно освободить время инженеров и направить их усилия на работу с функциями сервиса, имеющими большую значимость и ценность. Все это также позволит командам свести выполняемые вручную работы к минимуму по мере роста сложности и масштабов современной сервисной архитектуры.

Важно отметить, что устранение рутины не всегда является лучшим решением. Как уже упоминалось в этой главе, следует учитывать измеримые затраты, связанные с выявлением, проектированием и внедрением процессов или решений по автоматизации рутины. Как только будет выявлен рутинный труд, крайне важно понять, когда сокращение его объема имеет смысл. Для чего можно использовать анализ различных метрик, оценку окупаемости вложенных средств (ROI), анализ степени риска и итерационную разработку.

Поначалу рутинная работа обычно невелика, но быстро растет и через какое-то время начинает охватывать всю команду. Команды SRE должны неустанно снижать количество рутинной работы, потому что, даже если эта задача кажется сложной, результат того стоит. Каждый из описанных нами проектов требовал настойчивости и преданности со стороны соответствующих команд, которые иногда сталкивались со скептицизмом или неодобрением руководства. Мы надеемся, что эти истории помогут вам выявить рутину в вашей работе, измерить ее, а затем начать работать над ее устранением. Даже если сегодня вы не можете инвестировать в крупный проект, можете начать с небольшого примера для доказательства концепции, который позволит убедить вашу команду начать бороться с рутинной работой.