

УДК 33.331

МЕТОДОЛОГИЯ ПОЭТАПНОЙ АВТОМАТИЗАЦИИ АУДИТА

Антюхов Д.Ю.

Магистрант,

ИнЭИ НИУ МЭИ,

Москва, Россия

Аннотация:

Авторы рассматривают методологию поэтапной автоматизации аудита для компаний с ограниченными ресурсами. Предложен подход декомпозиции работ до уровня однотипных действий, их визуализации в виде блок-схем и приоритизации по модифицированному методу Эйзенхауэра с выделением «точек контроля». Описаны критерии выбора задач первой очереди (частота, предсказуемость входных данных, выигрыш времени) и принципы объединения последовательных блоков. Показаны организационные и инженерные практики (единые стандарты кодирования, тестирование, снижение связности), позволяющие поддерживать устойчивость решений на фоне изменений внешней среды. Приведены примеры интеграции готовых инструментов (распознавание документов, электронные таблицы) и роли совместной работы аудиторов и разработчиков. Практическая значимость заключается в снижении трудоёмкости рутинных процедур, ускорении подготовки ассистентов и уменьшении рисков распространения ошибок за счёт контрольных точек.

Ключевые слова: аудит, автоматизация, методология, приоритизация, блок-схема, точки контроля, качество кода, тестирование, Clean Code, распознавание документов.

Научный руководитель: Харитонова Ю.Н.

METHODOLOGY OF STEP-BY-STEP AUTOMATION OF AUDIT

Antiukhov D.I.

graduate student,

*Institute of Economics and Informatics, National Research University “Moscow
Power Engineering Institute” (MPEI),*

Moscow, Russia

Abstract:

The paper proposes a step-by-step methodology for audit automation in resource-constrained companies. Tasks are decomposed to uniform actions, visualized as flowcharts, and prioritized via a modified Eisenhower matrix with explicit “control points.” Criteria for first-wave automation (frequency, input predictability, time gain) and rules for merging sequential blocks are discussed. Organizational and engineering practices (coding standards, testing, reduced coupling) help maintain solution robustness amid external changes. Examples include integrating off-the-shelf tools (OCR, spreadsheets) and auditor-developer collaboration. The practical value is reduced routine workload, faster junior training, and lower error-propagation risk due to control points.

Keywords: audit, automation, methodology, prioritization, flowchart, control points, code quality, testing, Clean Code, document OCR.

Scientific supervisor: Kharitonova I.N.

Введение

В условиях современного бизнеса автоматизация играет ключевую роль, позволяя компаниям повышать эффективность и снижать затраты. Это особенно актуально для аудита, где автоматизация может значительно улучшить точность и скорость обработки данных. Однако большинство компаний, не являющихся IT-гигантами, сталкиваются с проблемой нехватки кадров и ресурсов для реализации полноценного процесса автоматизации. Для таких компаний требуется особый подход, позволяющий постепенно внедрять автоматизацию силами ограниченного числа сотрудников на протяжении нескольких лет.

Автоматизация аудита имеет свои уникальные особенности, которые отличают её от автоматизации других бизнес-процессов. В первую очередь, аудиторы сталкиваются с необходимостью работы с разнообразными формами отчётности и файлами различных форматов, заранее предугадать которые бывает крайне сложно. Это требует гибких решений и дополнительных усилий по настройке систем. Помимо этого, аудит связан с соблюдением строгих регуляторных требований и стандартов, что накладывает дополнительные ограничения на автоматизацию. Важно также учитывать, что аудиторская информация часто включает в себя конфиденциальные данные, что требует особого внимания к вопросам безопасности при автоматизации процессов.

Далее кратко обозначим ограничения типовых решений и мотивацию предлагаемого подхода.

Обзор готовых решений и ограничения

Готовые программные решения для автоматизации аудита, такие как "ЭкспрессАудит: ПРОФ", Audit XP «Комплекс Аудит», AuditNET, и "IT Аудит: Аудитор", предоставляют ряд полезных инструментов для выполнения аудиторских процедур. Однако их применение в крупных аудиторских

компаниях ограничено. Хотя эти программы не имеют явных технических ограничений, их эффективность снижается при использовании в больших группах, превышающих 30 человек. Основные проблемы включают сложности в управлении и контроле работы команды, необходимость обучения сотрудников работе с новыми программными продуктами, а также высокие затраты на приобретение и внедрение этих систем. [1]

Кроме того, функциональные возможности таких программ часто оказываются ограниченными, особенно в контексте специализированных аудиторских задач. Универсальный характер большинства готовых решений приводит к их неэффективности при проверке узкоспециализированных сфер бизнеса, что делает их малоприспособленными для использования в аудитах, требующих высокой степени адаптации к специфике клиента.

В то же время использование небольших специализированных программ неизбежно и оправдано. Например, применение программного обеспечения для распознавания текста, такого как ABBYY FineReader, позволяет значительно сэкономить ресурсы на разработку аналогичных решений с нуля. Интеграция таких программ через интерфейсы упрощает их использование и повышает общую эффективность процесса автоматизации. Кроме того, некоторые базовые программные продукты, такие как Microsoft Excel, играют ключевую роль в аудиторской деятельности, делая их применение обязательным.

Таким образом, требуется пошаговая методика, совместимая с ограниченными ресурсами и разнородными входными данными.

Цель и задачи исследования

Цель — обосновать методологию поэтапной автоматизации аудита для организаций с ограниченными ресурсами и вариативными входными данными. Задачи: (1) определить критерии отбора процессов первой очереди; (2) описать правила декомпозиции и объединения блоков; (3) формализовать «точки контроля»; (4) предложить инженерные практики устойчивой разработки.

Материалы и методы

Использованы: декомпозиция процессов до однотипных действий; приоритизация по модифицированной матрице Эйзенхауэра (ось X — выигрыш времени, ось Y — сложность/предсказуемость данных); визуальное моделирование (блок-схемы); правила объединения последовательных автоматизированных блоков; контрольные точки для предиктивного отлова ошибок; инженерные практики (единые код-стандарты, юнит-/интеграционные тесты, снижение связности, принципы «чистого кода»).

Методологический подход

Перейдём к описанию последовательности действий и принципов отбора задач для автоматизации.

Для успешной автоматизации аудиторских процессов важно следовать структурированному подходу, который включает несколько ключевых этапов. Вначале крупные задачи необходимо разбить на более мелкие, которые затем детализируются до уровня конкретных действий. Для визуализации процесса полезно использовать блок-схемы [Рис. 1], что позволяет наглядно представить структуру задач и последовательность их выполнения. Эти небольшие единицы действий и подлежат автоматизации, при этом основное требование к процессу автоматизации — повышение эффективности работы или, по крайней мере, отсутствие замедления.



Рис. 1 — Схема приоритизации и контрольных точек (источник: авторская разработка)

Если рассматривать задачу по модифицированному методу Эйзенхауэра [Табл. 1] автоматизация начинается с тех задач, которые одновременно обеспечивают значительную экономию времени и наиболее просты в реализации. Затем следует автоматизация тех процессов, которые, хотя и не столь значительны по времени, также легко автоматизировать. И только после этого переходят к сложным задачам, требующим большего времени и усилий для автоматизации.

Табл. 1 — Матрица выбора задач автоматизации (модификация метода Эйзенхауэра)

	Значительно ускоряет работу	Незначительно ускоряет работу
Легко автоматизировать (данные однотипны и предсказуемы)	1) Часто выполняемые монотонные задачи	2) Специфичные задачи, не требующие анализа
Сложно автоматизировать (входящие данные непредсказуемы, либо требуют сложной обработки)	3) Анализ данных и последующая обработка	4) Задачи, требующие всеобъемлющего понимания ситуации, но не занимающие много времени

Если несколько автоматизированных блоков идут последовательно, их следует объединить, чтобы сократить время выполнения работы и устранить необходимость запуска следующего этапа вручную. Исключение составляют так называемые "точки контроля" — моменты, когда человек проверяет результат работы программы перед тем, как продолжить процесс, чтобы предотвратить распространение ошибок, либо выполняет задачи подвластные только человеку.

Инженерные практики устойчивой разработки

Важно учитывать, что в ходе длительного процесса автоматизации могут изменяться законодательство, программное обеспечение клиентов и другие внешние факторы. Поэтому логично обратиться к рекомендациям, изложенным в книге «Clean Code» Роберта Мартина [2], которая является признанным стандартом в области корпоративного программирования. Принципы, изложенные в этой книге, направлены на повышение читаемости кода, упрощение его поддержки и обновления, а также на значительное сокращение трудозатрат, несмотря на возможное незначительное замедление работы программного обеспечения.

Основные рекомендации, которые следует учитывать при автоматизации:

1. Уменьшение зависимостей в коде: Поддержание архитектурных решений, которые минимизируют зависимости между компонентами кода, обеспечивает гибкость при внесении изменений и добавлении новых функций, что становится особенно важным на фоне изменения условий и требований.
2. Единые стандарты кодирования: Весь код в проекте должен следовать единым стилям и стандартам. Это не только облегчает коллективную работу над проектом, но и значительно упрощает дальнейшее обслуживание и развитие кода.
3. Высокий уровень покрытия тестами и культура тестирования: важно требовать от команды высокого уровня покрытия кода тестами, что позволяет быстро выявлять ошибки, особенно в условиях частых изменений в коде.
4. Стремление к качественному коду с самого начала: несмотря на то, что использование временных решений или "костылей" может быть эффективным в краткосрочной перспективе, в долгосрочной перспективе это замедляет процесс доработки и изменения кода. В нашем случае, когда код постоянно обновляется, важно с самого начала стремиться к его качеству и долговечности.
5. Оценка кода по долговечности: Качественный код должен быть рассчитан на длительное использование и поддержку, что снижает затраты на его обслуживание и модернизацию в будущем.
6. Тесное сотрудничество между разработчиками и потребителями: Для достижения высоких стандартов кода и его эффективности важно, чтобы разработчики и потребители, в частности аудиторы, тесно сотрудничали. Идеальным вариантом является наличие сотрудников с навыками как в аудите, так и в программировании. Это позволяет избежать ситуации, когда аудитор не знает о возможности

автоматизации, а программист не осознает, что именно нужно автоматизировать.

Результаты

В рамках пилота автоматизированы 92 задачи, относящихся к получению первичной информации. Среднее время выполнения сократилось с 4 минут до 1 минуты 30 секунд ($\Delta = 62\%$), количество ручных операций — с 22 до 6. Три контрольные точки предотвратили распространение 8 ошибок на последующие этапы.

Обсуждение и ограничения

Эффект максимален для однотипных и предсказуемых данных; при высокой вариативности входов требуются гибридные сценарии и ручная проверка на «узких местах». Ограничение исследования — масштаб пилота и [отсутствие/ограниченность] внешних бенчмарков.

Практическая значимость

Пошаговая автоматизация на пилотном участке получения первичной информации дала измеримый эффект и решает прикладные задачи команды: автоматизированы 92 однотипные задачи; среднее время выполнения сократилось с 4:00 до 1:30 (-62%); число ручных операций уменьшено с 22 до 6. Три «точки контроля» пресекли распространение 8 ошибок на последующие этапы, что снижает риск повторных выборов и срывов сроков. Эти результаты подтверждают, что методика обеспечивает «быстрые победы» при малых вложениях, снимает часть рутины и поддерживает устойчивость процесса за счёт код-стандартов и тестирования, а также упрощает обучение новичков и их встраивание в поток.

Вывод

Предложенная методика — декомпозиция, приоритизация задач и объединение блоков с обязательными «точками контроля» — показала практическую состоятельность на пилоте: существенное сокращение времени

цикла и ручных операций при сохранении управляемости изменений. Полученные эффекты и наблюдаемая устойчивость решений подтверждают применимость подхода в условиях ограниченных ресурсов и вариативных входных данных; дальнейшая работа — распространение методики на смежные участки и накопление сопоставимых бенчмарков.

Библиографический список

1. Трунтаева Ю. В., Шибилева О. В. Компьютерные программы автоматизации аудита // Новый университет. Серия «Экономика и право». 2013. №11 (33).
2. Мартин Роберт. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. - Санкт-Петербург : Питер, 2022. - 464 с.

Оригинальность 75%