Мое видение - подход к AI-трансформации корпоративного ИТ

Шок-прогноз: минус 50 % ИТ-штата к 2028 году и 70 % к 2030-му, как минимум в традиционных универсальных банках. Как я бы проводил AI-трансформацию корпоративного ИТ по этапам.

Коротко. Я всерьез полагаю, что к концу 2028 года штат корпоративного ИТ, как минимум в традиционных универсальных банках, можно сократить на 50 %, к концу 2030 года - на 70 % от численности конца 2026 года. Раздать сотрудникам подписки и AI-инструменты для этого мало: нужны мандат акционеров на сокращения, глубокий аудит ИТ силами агентов, пилоты, начиная с некритических направлений, и плановая экономия, посчитанная вместе с финансовым директором. Ниже расскажу, как вижу процесс.

Мой шок-прогноз: к концу 2028 года штат корпоративного ИТ, как минимум в традиционных универсальных банках, можно сократить на 50 %, а к концу 2030 года - на 70 % от численности конца 2026 года. Скриньте. Я уверен, что это не пустые слова, а реальность, которая нас ждет. К сожалению или к счастью. Альтернативный сценарий - компаниям вдруг понадобится в разы больше ИТ-инфраструктуры и людей для проектов текущего бизнеса, чтобы выживать в новом мире, поэтому расходы на ИТ останутся, но кратно вырастут производительность и объем работы. Я в это не верю, честно говоря.

Почему именно 50 и 70 %? Прямого прогноза «минус половина ИТ-штата» у лидеров консалтинга нет, это моя прикидка. Но направление авторитеты подтверждают. В Google доля нового кода, который пишет AI, выросла с 25 % в 2024 году до 75 % в апреле 2026 года, и этот код проверяют инженеры частично. Gartner считает, что большую часть кода будет генерировать AI, а до 2027 года переучиваться придется 80 % инженеров. Глава Anthropic Дарио Амодеи в 2025 году предупредил, что за 1-5 лет AI может убрать до половины начальных офисных позиций. А 40 % работодателей в опросе Всемирного экономического форума планируют до 2030 года сокращать штат там, где AI автоматизирует задачи. На мой взгляд, рынок перешел из стадии «отрицания необходимости преобразований» в стадию гнева от того, что привычный уклад поменяется. Дальше, как вы знаете, будет торг. Лучше сразу приходить к принятию и начинать действовать сейчас, пока есть время.

Мой расчет простой. **Сократить штат на 50 % при той же нагрузке можно, если каждый оставшийся сотрудник с агентами делает вдвое больше, на 70 % - в 3,3 раза больше.** Когда агенты пишут три четверти нового кода, двукратный рост к 2028 году по всему ИТ я считаю достижимым. Для трехкратного к 2030 году агенты должны так же глубоко войти и в Run: администрирование, поддержку, эксплуатацию и инциденты.

Не можешь остановить - возглавь. Поэтому программу AI-трансформации я бы строил уже сейчас и под этот горизонт. Иначе можно несколько лет улучшать рабочие инструменты, пропустить

конкурентов вперед, сохраняя прежнюю стоимость ИТ. А она, по моему опыту, доходит до 20-30 % всех расходов современной корпорации с цифровым уклоном.

Почему я так уверен в том, что говорю? Снижать стоимость ИТ я умею и без AI: за год снял с банка треть ИТ-затрат без остановки продуктов. А с AI-инструментами я уже сам разрабатывал корпоративное ПО. Качество оказалось намного выше ожиданий, а модели с 2024 года буквально становятся лучше каждый месяц. В действительности LLM-модели - идеальные переводчики с одного языка на другой, и при наличии детальных требований к функционалу и архитектуре они спокойно сделают перевод человеческих требований в код - так это работает.

Сразу обозначу границы. Это не полный гайд по AI-трансформации, а мой взгляд на ключевые шаги: мандат, аудит, деньги, пилоты, контроль и люди. Как любая трансформация, она делается как костюм на заказ - под конкретную компанию, ее системы и людей. В статье - общие подходы. Как переконфигурировать оргструктуру и управление ИТ после AI и какой оргподход и governance выбрать для внедрения в каждом домене - большие отдельные темы. Я их называю, но не раскрываю. Если будете активны на LinkedIn - продолжу писать о наболевшем.

Масштаб: весь Change и весь Run

AI надо внедрять не в одну разработку. Он нужен и в постановке требований, архитектуре, разработке, тестировании, DevOps, то есть во всем развитии (Change). Также он нужен в сопровождении (Run): поддержке пользователей, мониторинге и инцидентах, эксплуатации, работе с вендорами и даже в администрировании и SRE. Плюс контроли и проверки внутри каждого процесса. **Масштаб огромный, и смотреть на него надо сразу целиком.** Боитесь пустить агента даже в тестовую инфраструктуру или управлять корпоративным облаком через API? **Через 2 года конкурент вас обгонит.** Поэтому вопросы информационной безопасности надо решать уже сейчас, а не после первых пилотов.

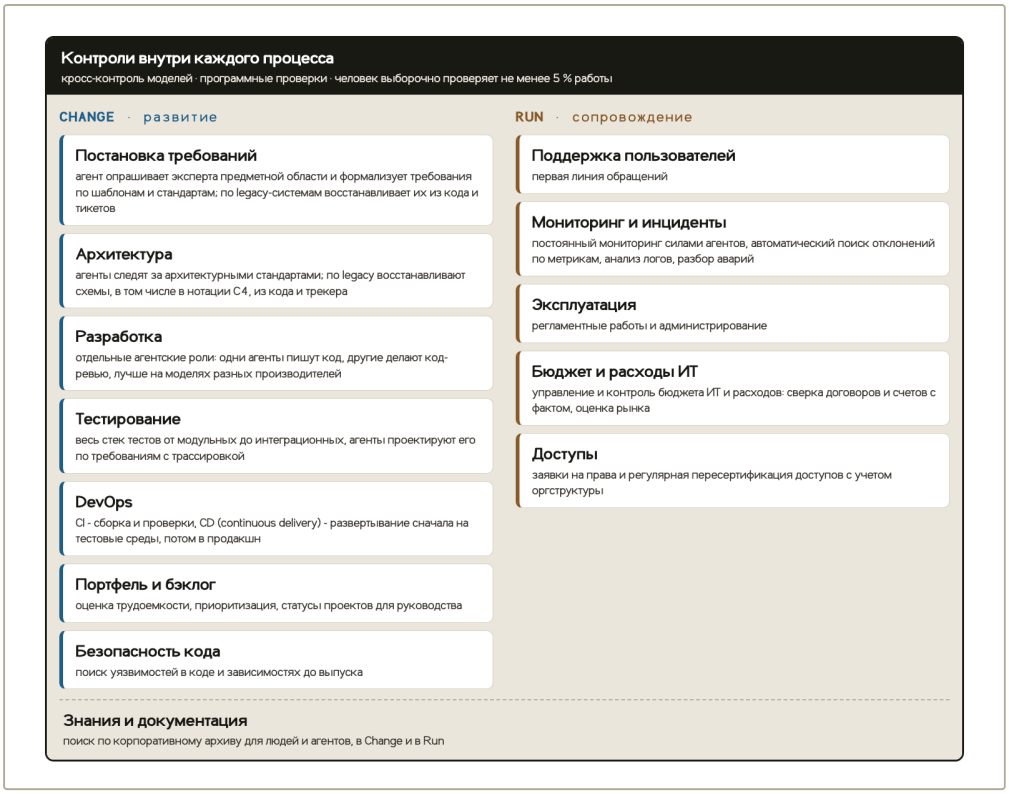


Рисунок 1. Где применять AI в корпоративном ИТ и что менять уже сейчас

Параллельно я бы построил поиск по архиву знаний компании, как мы сделали это в FlyArchive: агентам на всех этапах нужен контекст, а он лежит в документах, задачах и переписке.

Пререквизиты

Первое - жесткий мандат CEO и акционеров на изменение стоимости и статус-кво корпоративного ИТ. Нужны полномочия пересматривать процессы, состав команд и договоры и решать, кого сокращать. Если заранее пообещать сохранить каждое подразделение, его бюджет и legacy-команды с уникальной экспертизой, результат упрется в эти обещания.

Сопротивление будет. Руководители будут объяснять, почему именно их команды трогать нельзя и почему AI не подходит к их системе, присылать мемы о том, как AI ошибается. Относиться к этому надо спокойно. **Хорошие отношения с менеджером среднего звена не основание сохранять его бюджет в корпорации.**



Рисунок 2. Мем про грибы - его вам пришлют одним из первых

Второе - данные. Порядок работы с ними согласуется с инфраструктурой ИТ и информационной безопасностью: либо часть данных разрешено передавать внешним моделям, либо компания ставит у себя открытые модели семейств DeepSeek, Kimi или GLM. Их качества уже хватает для 95 % задач. Свой AI-кластер считается под нагрузку: процессу с 7-10 одновременно работающими агентами нужна мощность на них. Покупать кластер до расчета экономики я бы не стал, арендовал бы в облачном ЦОДе для начала.

Третье - цифры до старта. С CEO, акционерами и финансовым директором сразу согласуются исходная стоимость ИТ, ограничения по качеству и доступности и цель: снижение расходов, скорость изменений, обслуживание роста бизнеса прежним составом.

Этап 1. Аудит силами AI-агентов

Начинать надо со сверки всей ткани ИТ: код, процессы, бюджеты, договоры с подрядчиками, что компания заказывает, что получает и сколько платит. Аудит делают агенты. Им нужен доступ на чтение:

- к коду и тестовым контурам;
- к трекерам и базам знаний;
- к архитектуре и интеграциям;
- к мониторингу и истории аварий;
- к решениям правления;
- к бюджетам и договорам;
- к регламентам процессов.

Каждый вывод одной модели проверяет модель другой компании по первоисточникам, и каждый вывод идет со ссылками на код, задачу и документ. Оценка агента - повод для проверки, а кадровое решение требует подтвержденных фактов.

«Жир» на практике обычно сидит в трех местах:

- раздутые команды;
- договоры сопровождения и доработок;
- работа, которую делают по привычке.

Для большинства проектов сейчас хватит команды из 3-5 человек, которые работают с агентами. Сильный менеджер умеет защищать бюджет, но с эффективностью подразделения это путать нельзя. Сокращать при этом надо не всех: топ-синьоров с AI-компетенцией и средний слой менеджмента я бы оптимизировал, но многих оставил, так как на первых держатся hard skills, а вопросы между подразделениями сами собой не решатся. **Удивительно, но именно болтуны и коммуникаторы внутри компаний станут полезнее молчунов-кодеров с дальнейшим развитием моделей.** Еще раз, понятно, это не про всех, и технические навыки будут нужны людям для контроля машин. Только людей таких нужно сильно меньше.

Результат этого этапа - карта ИТ-доменов со стоимостью, владельцами и проблемами и короткий список предполагаемых изменений: какой процесс меняем, какой эффект ждем, чем проверяем, кто отвечает и когда останавливаемся.

Этап 2. Деньги и TTM с первого дня

Начинать надо, сразу измеряя эффект. Если быть честным перед акционером, эффект - это в основном деньги и TTM (time to market, срок вывода изменений на рынок). **Здесь проходит граница между игрушкой с удобными инструментами и реальным изменением бизнеса.**

Внешние исследования дают ориентир, но не ответ. В экспериментах с 4 867 разработчиками в Microsoft, Accenture и компании из Fortune 100 AI-помощник дал в среднем примерно на 26 % больше завершенных задач (Microsoft Research), но это был уже далекий 2025 год. **Поэтому мерить надо у себя.**

Раньше для этого покупали внешние бенчмарки: например, ранее я использовал неплохие инструменты типа BlueOptima, которые считают производительность разработчиков по изменениям в исходном коде и сравнивают компании между собой. Сейчас, на мой взгляд, это можно сделать самим: агенты разбирают репозиторий и трекер, считают поставку по командам и сами исследуют, как ее корректно сравнивать. Внешний бенчмарк уже для этого не нужен.

Чтобы не утонуть в метриках, у программы я бы оставил две «Полярные звезды» (North Star):

1. **Общая стоимость ИТ:** C = ФОТ штата + аутстафф + подрядчики + лицензии + инфраструктура + сопровождение вендорских систем + расходы на AI. Она должна валово идти вниз каждый квартал. Экономия на людях, которую съели подписки и кластер, не считается.
2. **Срок поставки (TTM):** время от согласованной задачи до работающего результата, включая ожидание согласований. Он не должен расти, иначе сокращение просто переложило работу

в очередь.

Качество - не звезда, а ограничение: дефекты после внедрения и откаты не растут. Для разработки срок и качество удобно мерить метриками DORA. По отчету DORA за 2025 год AI связан с ростом пропускной способности разработки и со снижением стабильности поставки, поэтому проверки усиливаются вместе с объемом изменений. Токены и сессии показывают расходы, а не пользу: неважно, сколько токенов сожгла та или иная задача, бизнесу важен результат.

Акционеру из первой метрики хватит двух формул:

1. **Эффект для компании** = d × ΔC, где d - доля ИТ в расходах компании, ΔC - снижение стоимости ИТ. При d = 25 % и снижении стоимости ИТ на треть расходы всей компании падают примерно на 8 %.
2. **Окупаемость в месяцах:** T = R / (ежемесячная экономия - ежемесячные расходы на AI), где R - разовые расходы перехода: обучение, выходные пособия, параллельная работа старого и нового процесса.

У эффекта должен быть владелец, и работает он вместе с финансовой функцией. Высвободились часы - принимается решение: сократить подрядчиков, отказаться от найма, уменьшить штат или отдать мощность бизнесу. **Одни и те же часы нельзя записать и в экономию, и в дополнительную разработку.** Красивого сводного процента недостаточно. Главное - уметь идентифицировать факт появления свободного времени.

Этап 3. Пилоты по цене ошибки

Исходя из первого этапа берем в работу низко висящие фрукты. Первыми в эксперимент идут периферийные системы и повторяемые операции поддержки: ошибка там обходится дешево, а результат виден быстро. Затем - в очереди более значимые системы. До mission critical, то есть систем, отказ которых останавливает основные операции, - идем по мере накопления результатов. Критичность системы и права агента - разные вещи: код критической системы агент может анализировать раньше, чем получит право что-то в ней менять. Как я сказал выше, аудит и анализ можно и нужно делать с помощью агентов уже сейчас.

Главное начнется в середине пилота. Это «бюрократия», которая может «съесть всю оптимизацию». Если агент ускорил подготовку кода, а изменение по-прежнему неделями ждет согласований, экономика не изменилась. Придется менять очереди, границы команд и полномочия людей. Здесь комфортный эксперимент с инструментами заканчивается. Тут придется менять процессы, так как бутылочное горлышко может быть не в аналитике, разработке и тестировании, а в подписании бумаг.

Этап 4. Контроль

Образец для меня - автоматизированный автомобильный завод: операции выполняют роботы, люди следят за качеством и останавливают линию при браке. Человек, который механически нажимает «одобрить», контроля не создает.

Модель - инструмент, а ответственность за результат остается на людях. Поэтому оставшиеся в ИТ сотрудники становятся операторами, как на таком заводе. Они отвечают за модели, с которыми работают, знают, где те ошибаются, и умеют перепроверить их результат другой моделью, программной проверкой и своей экспертизой.

Контроль держится на трех механизмах:

1. **Кросс-контроль моделей.** Мой любимый. Модель другой компании сначала составляет свою оценку, затем разбирает расхождения с исполнителем. Я много раз проверял результаты Claude с помощью Codex и наоборот. Это очень эффективно работает.
2. **Программные проверки:** тесты, статический анализ, сверки, права. Сходимость проводки проверяется вычислением, а не объяснением модели. Никто не отменяет инструменты типа SonarQube и им подобные даже с появлением моделей.
3. **Выборочная проверка человеком:** новые типы задач, расхождения моделей, случайная часть обычной работы. Действия с высокой ценой ошибки человек согласует всегда. **Также человек должен проверять выборочно минимум 5 процентов объема работы модели.**

Полномочия агента растут ступенями: чтение, изменения в изолированной среде, изменения на согласование, ограниченное самостоятельное выполнение. **Люди должны научиться контролировать машины.** Очень важная проблема - агенты внутри корпорации не знают про оргструктуру, правила, ограничения, пока им явно это не указано. В этом смысле агенту все равно, младший или старший аналитик имеет право смотреть витрину данных. Выстраивая контроль, надо также выстраивать многоуровневую систему защиты от «вышедших за флажки агентов». Но это отдельная тема.

Этап 5. Люди и жесткие решения

Горизонт 2028 года означает, что ждать, пока все добровольно освоят AI, нельзя. **Скорее всего, придется принимать жесткие решения: убирать часть неадаптировавшихся людей, сохраняя нагрузку на подразделение.** Тогда оставшимся придется работать с AI-инструментами, иначе они не справятся.

Помогать при этом надо. Обучение начинается с тех, кто уже уверенно работает с агентами, и заканчивается рабочим результатом на реальных задачах: решенным обращением, принятым изменением, проверенным отчетом. На обучение выделяется время, так как поверх полной загрузки оно останется в планах.

Дальше у команд цели по стоимости, срокам и качеству, и их достижение меняет бюджет и численность. Руководитель, который получил инструменты и время, отвечает за результат. Пока статус руководителя зависит от числа подчиненных, у него прямой стимул сохранять штат, поэтому критерии успеха менеджеров тоже меняются.

Не всякое возражение - саботаж: конкретный риск надо воспроизвести и проверить. Но «у нас это невозможно» без примера программу не останавливает. **С теми, кто после переходного периода не может или не хочет работать по-новому, надо расставаться, включая руководителей.** Специалиста со скрытыми знаниями о системе отпускают только после передачи этих знаний в документацию и проверки. Сокращение стоит денег и времени, и это закладывается в экономику первого года.

После сокращений оргструктуру и управление ИТ придется переконфигурировать: другие роли, другая подчиненность, операторы моделей вместо исполнителей. Это большая тема, она требует отдельной проработки.

Вывод

Начинайте подход к снаряду у себя в компаниях уже сейчас. Не ждите. Иначе все будет, как в середине 2010-х, когда ряд известных топ-менеджеров съездили в Кремниевую долину и там обнаружили автономные agile-команды, после чего тема понеслась по просторам СНГ. Как по мне - уже очень скоро будут первые такие корпоративные примеры, которые до глубины души поразят акционеров и бизнес, и все захотят трансформацию, поэтому начинать пора сейчас.

За рамками статьи осталось несколько тем, каждая тянет на отдельный текст:

1. Защита от агентов, «вышедших за флажки»: права, оргструктура и многоуровневый контроль.
2. Свой AI-кластер или облако: как посчитать нагрузку и экономику.
3. Данные и регулятор: что можно передавать внешним моделям и кто отвечает за ошибку агента.
4. Как провести сокращения: трудовое право, передача знаний, разговор с командой.
5. Как мерить продуктивность разработки агентами по репозиторию вместо внешних бенчмарков.
6. Новая оргструктура и управление ИТ после AI: роли, уровни, подчиненность.
7. Оргподход к внедрению в каждом домене: какой нужен governance и кто что решает.
8. Как мотивировать и вдохновлять людей, которые остаются.

Ваше мнение может не совпадать с моим. Напишите в комментариях на LinkedIn, согласны ли вы и какую из этих тем разобрать в следующей статье.

Для тех, кто дочитал до конца, - в подарок архив с набором скиллов для AI-агентов ai-sdlc-kit 0.1.3 для реверс-инжиниринга вашего любого legacy-кода в требования и архитектуру. По коду готовой системы агент сам пишет ее описание: из каких частей она состоит, какие в ней процессы, что она

должна делать и как это проверить. Второй агент сверяет написанное с кодом и возвращает на доработку то, что не сошлось. На своем проекте я так описал 25 процессов, и попутно набралось около 480 замечаний к самому коду, среди них настоящие ошибки безопасности.



Андрей Сабынин

Основатель FlyOptima – AI-native разработчика решений для банков и финтеха.