

Дмитрий Зугаев

Проблемный проект

Чек-лист проблемного web-проекта

Материал помогает за 15-20 минут собрать первые признаки риска: почему проект застрял, какие факты запросить у подрядчика или команды, когда можно продолжать работу, а когда нужен независимый технический разбор.

Если подрядчик затягивает разработку Проверьте не эмоции вокруг срока, а факты: что реально готово, что блокирует релиз и за что сейчас просят оплату.

Симптомы технического тупика Отметьте признаки, которые уже повторяются, а не появились один раз на отдельной задаче.

Если проект нельзя передать или развивать Передача проекта срывается не из-за отсутствия одного файла, а из-за отсутствия управляемой картины системы.

Документы, которые нужно запросить Документы нужны не для формальности, а чтобы проверить, можно ли управлять проектом по фактам.

Если портал ломается после релизов Повторные поломки после доработок обычно показывают проблему в архитектуре, тестировании, релизном процессе или интеграциях.

Вопросы на ближайшем статусе Эти вопросы помогают отделить нормальную рабочую сложность от потери управляемости.

Дмитрий Зугаев web/AI-архитектор, СТО

<https://zugaev.ru/lead-magnets/checklist-problemnogo-proekta/>

Кому подходит

Когда использовать материал

подрядчик переносит релиз, просит оплатить этап или новый бюджет, но не показывает прозрачную готовность

проект сложно передать другой команде: нет документации, карты модулей, доступов и понятного backlog

портал или личный кабинет сложно развивать без новых ошибок

нужно понять, продолжать с текущей командой, подключать технический контроль или готовить восстановление

AI-функция, интеграции или автоматизация упираются в качество данных, архитектуры и процессов

Чек-лист

Что проверить

Если подрядчик затягивает разработку

Проверьте не эмоции вокруг срока, а факты: что реально готово, что блокирует релиз и за что сейчас просят оплату.

- есть список задач этапа со статусами: готово, на проверке, заблокировано, не начато
- подрядчик показывает рабочее демо, а не только обещает, что «почти готово»
- понятно, какие конкретные причины сдвинули срок и какие из них уже закрыты

- новый срок привязан к задачам и рискам, а не назван как очередная дата без плана
- оплата этапа опирается на проверяемые артефакты: демо, доступы, changelog, тесты, документы

Симптомы технического тупика

Отметьте признаки, которые уже повторяются, а не появились один раз на отдельной задаче.

- оценки похожих задач растут от релиза к релизу
- новые доработки ломают соседние функции
- подрядчик отвечает общими формулировками без карты причин
- команда не может показать актуальную схему модулей и интеграций
- бизнес не понимает, что действительно готово, а что только почти готово

Если проект нельзя передать или развивать

Передача проекта срывается не из-за отсутствия одного файла, а из-за отсутствия управляемой картины системы.

- нельзя быстро объяснить, из каких модулей состоит система и за что отвечает каждый модуль
- нет списка доступов, окружений, доменов, репозиторий, интеграций и ответственных
- новая команда не может оценить доработки без долгого погружения в код и переписки
- backlog содержит задачи без бизнес-контекста, критериев готовности и приоритета
- текущая команда держит критичные знания в голове, чатах или личных заметках

Документы, которые нужно запросить

Документы нужны не для формальности, а чтобы проверить, можно ли управлять проектом по фактам.

- актуальный список задач и статус ближайшего релиза
- схема модулей, ролей пользователей и ключевых интеграций
- описание процесса деплоя и тестирования
- список известных технических рисков и ограничений
- инструкции по доступам, окружениям и передаче проекта

Если портал ломается после релизов

Повторные поломки после доработок обычно показывают проблему в архитектуре, тестировании, релизном процессе или интеграциях.

- после каждого релиза есть список изменённых модулей и связанных рисков
- критичные сценарии проверяются до выкладки, а не только после жалоб пользователей
- ошибки повторяются в одних и тех же местах: интеграции, роли, заказы, отчёты, платежи
- команда может показать, почему новая доработка затронула соседнюю функцию
- есть план стабилизации, а не только быстрая починка очередного симптома

Вопросы на ближайшем статусе

Эти вопросы помогают отделить нормальную рабочую сложность от потери управляемости.

- что именно сейчас блокирует релиз
- какие задачи требуют предварительного исследования перед оценкой
- какие модули чаще всего создают ошибки после доработок
- какие риски команда уже знает, но пока не закрывает

- что нужно сделать, чтобы новая команда могла безопасно принять проект

Если AI или интеграции упираются в хаос

AI и новые интеграции не исправляют слабую архитектуру автоматически. Часто они быстрее показывают старые проблемы в данных, ролях и документации.

- понятно, какие данные считаются источником истины для портала, CRM, 1С и внешних сервисов
- описаны роли и ограничения: кто может видеть, менять и выгружать информацию
- техническая документация позволяет понять, где брать контекст для AI/RAG или автоматизации
- есть критерии качества ответа, обмена или автоматического действия
- известны сценарии, где AI или интеграция могут навредить из-за неполных данных

Красные флаги

Если несколько пунктов совпадают, лучше не запускать новые крупные доработки без диагностики.

- каждый перенос срока объясняется фразой «там всё сложно»
- документация откладывается до конца проекта
- демонстрации заменяются устными обещаниями
- нет списка критичных ошибок и плана их закрытия
- подрядчик просит новый бюджет без объяснения, что изменилось в проекте
- для AI, интеграций или нового раздела предлагают сразу разработку без проверки данных и архитектуры

Когда нужен аудит

Аудит нужен не всегда. Но он полезен, когда цена следующего решения выше стоимости независимой проверки.

- нужно понять, продолжать ли работу с текущей командой
- проект готовится к передаче другой команде
- сроки сорваны, но причина не подтверждена фактами
- архитектура и документация не позволяют планировать развитие
- портал ломается после релизов, а команда чинит симптомы без карты причин
- AI или интеграции планируются поверх непонятных данных и процессов
- бизнесу нужен план восстановления без конфликта и хаоса

Важно

Чек-лист не заменяет технический аудит. Он помогает собрать первые признаки и подготовить разговор, но выводы по архитектуре, рискам и передаче проекта нужно делать после проверки фактов.

Нужен разбор проекта?

Опишите ситуацию на сайте zugaev.ru. Дмитрий сможет подсказать, с какой диагностики начать: аудит, AI-ready проверка, контроль подрядчика или восстановление проекта.

<https://zugaev.ru/lead-magnets/checklist-problemnogo-proekta/>