

## Как управлять рисками IT-проекта

Если какая-нибудь неприятность может произойти, она случится — гласит закон Мерфи. И сфера ИТ не исключение.

Команда SEBEKON рассказывает, как работать с рисками в IT-проекте, чтобы риски не стали управлять проектом.

В этой статье под IT-проектом подразумевается разработка сложного веб-ресурса — например, b2b-портала или интернет-магазина с множественными интеграциями с внешними системами.

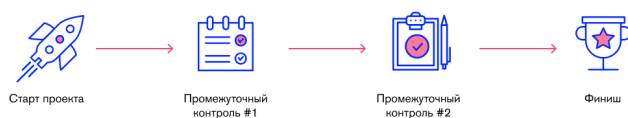
Дилемма управления рисками любого IT-проекта простая: перестраховаться и заложить все угрозы в бюджет проекта, который вырастет до небес, или пропустить часть рисков — и тогда будет велика вероятность не завершить проект вовсе или получить неудовлетворительный результат.

Но даже если принять первый вариант как единственно верный, от рисков никто не застрахован.

### Как работать с рисками

Обычно ожидаемая реализация проекта не совпадает с реальной работой:

#### Ожидание



#### Реальность



На практике управление рисками — это тонкий баланс между разумным и достаточным.

Существуют пять основных инструментов для работы с рисками:

- выявление,
- оценка,
- планирование мероприятий по предотвращению,
- предусмотрение действий при наступлении,
- мониторинг.

Рассмотрим каждый из них подробнее.

### Выявлять риски

Риски проекта определяются во время фиксации требований к проекту и отражаются в уставе или концепции проекта.

Устав проекта — это документ, в котором зафиксирована основная информация о проекте: потребности заказчика, бизнес-задачи, высокоуровневые требования к проекту, критерии успешной реализации.

Концепция проекта — документ, в котором собраны подробные характеристики проекта.

Выбор документа зависит от масштаба проекта.

Величина проекта	Длительность	Бюджет (руб.)	Перечень документов для фиксации рисков
Маленький проект	до 1 месяца	около 500 тыс.	Концепция проекта
Средний проект	до 3 месяцев	от 1 до 5 млн.	Концепция проекта
Большой проект	6–12 месяцев	от 10 до 20 млн.	Устав проекта
Мега-проект (обычно состоит из нескольких взаимосвязанных, более мелких проектов)	более 12 месяцев	от 30 до 40 млн.	Устав проекта, дополнительные документы на каждый этап (доп.соглашения, приложения и т.п.)

Пример классификации проектов и рекомендуемых документов для фиксирования рисков

Чтобы выявить риски, нужно проанализировать множество информации и сопоставить, как те или иные факторы влияют на возникновение рисков.

Рассказываем, что стоит сделать для этого.

### **Учесть опыт предыдущих проектов**

Для примера возьмём согласование дизайна страниц сайта. Дизайн всегда субъективная история, и нужно закладывать дополнительное время на решение вопросов в стиле «а давайте попробуем немного светлее или поиграем со шрифтами». Заказчик должен ещё до начала работ понимать, что любое изменение, требующее нескольких минут работы, в итоге выливается в часы и дни согласования.

Часто на стороне заказчика необходимо согласовать дизайн с несколькими департаментами — и не всегда сотрудники этих отделов могут быстро подтвердить изменения. Если речь идёт о крупной компании, то руководителю проекта придётся назначать собрание или созвон для обсуждения дизайна. И не всегда это получится сделать в ближайшие дни — часто согласование растягивается на недели, а то и на месяцы.

В нашей практике были разные проекты, но редко, когда удавалось согласовать дизайн в первой же итерации и по ходу работы не менять его. Гораздо чаще приходилось сдвигать сроки работ, потому что представители заказчика не смогли оперативно обсудить дизайн и согласовать его.

### **Оценить новизну критичных требований — как для заказчика, так и для исполнителя**

Например, в проекте предполагается разработка нового функционала, которая требует совместной работы нескольких подразделений на стороне заказчика и дополнительных исследований на стороне исполнителя. Всё это увеличивает сроки согласований и влияет на срок проекта.

### **Принять во внимание особенности внутренней инфраструктуры заказчика**

Самый очевидный пример — требования к безопасности, которые могут существенно отодвинуть срок старта проекта. Это связано с необходимостью получения доступов в систему заказчика или требований по использованию определённого, разрешённого на стороне заказчика ПО, которые повлекут за собой дополнительную проработку архитектуры проекта.

### **Не забыть про покупку серверного оборудования или про согласование хостинга**

Если про покупку хостинга заказчики обычно помнят, то про серверное оборудование в проектах могут забыть и не заложить в бюджет стоимость оборудования, лицензий, а также время на заключение договора с поставщиком оборудования и его доставку.

В зависимости от проекта серверное оборудование может быть стационарным — шкафы с оборудованием — или облачным, когда проект размещается в виртуальном пространстве.

Напоминание заказчику о закупке серверного оборудования на первых же обсуждениях проекта позволит избежать досадных простоев в работе.

### **Заложить внедрение интеграции проекта с внутренними системами заказчика**

Об этом часто забывают, поскольку обычно в организации заказчика под проект выделяются не специально нанятые люди, а текущие сотрудники, у которых есть другая, основная работа. Они воспринимают новый проект как нечто второстепенное, что можно делать по остаточному принципу.

В итоге когда приходит время интеграций проекта с CRM-, ERP- и другими системами работа останавливается на неопределённый срок, потому что менеджер проекта со стороны заказчика не может оперативно подготовить необходимую документацию и доступы для интеграций.

## Выявить заинтересованность в проекте ключевых стейкхолдеров

Стейкхолдеры — это не только топ-менеджмент, но и будущие пользователи и эксперты, которые хорошо разбираются в вопросе и могут помогать с проектом или же оказывать негативное влияние.

На начальном этапе стоит понять, как взаимодействовать с так называемыми негативно настроенными стейкхолдерами, чтобы они не вредили реализации проекта. Под негативно настроенными мы понимаем не тех, кто намеренно вставляет палки в колеса, а тех, кто не понимает последствий от своих решений.

В нашей практике не раз случалось, когда некоторые стейкхолдеры не глядя согласовывали ТЗ, а на этапе тестирования готового проекта выяснялось, что всё должно быть совсем не так.

Например, руководитель департамента логистики должен был проверить сценарий по расчёту доставки товара. Но он был занят текущей деятельностью, мельком просмотрел ТЗ и отправил с пометкой «принято». В ходе тестирования этого сценария оказалось, что он должен быть реализован другим способом. В итоге пришлось откатить проект назад: пересмотреть внесение правок в сценарий, затем внести эти правки в ТЗ, согласовать это со всеми стейкхолдерами, переделать сценарий и заново его протестировать. На всё это ушло больше месяца, а кроме того, заказчику пришлось оплатить дополнительные работы.

Для наглядности можно составить матрицу влияния стейкхолдеров на проект и методы взаимодействия с ними. Например, такую:

	1. Настроены критически	2. Настроены нейтрально	3. Настроены поддержать
Лица, принимающие решения	Понять и удовлетворить потребности.	Понять их потребности. Сделать так, чтобы они были довольны.	Сохранять и поддерживать энтузиазм.
Инфлюенсеры (те, кто влияют на принимающих решения)	Понять и удовлетворить потребности.	Понять их потребности. Сделать так, чтобы они были довольны.	Сохранять и поддерживать энтузиазм.
Заинтересованные люди (не имеющие влияния)	Приложить минимальные усилия по устранению разногласий.	Игнорировать.	Привлечь как сторонников и помощников.

**Изучить соответствующее законодательство, поговорить с экспертами и опытными коллегами**

У более опытных коллег за плечами больше реализованных проектов, они уже набили свои шишки и могут дать полезные советы. Эксперты помогут разобраться в отдельных нюансах относительно, например, сценариев пользователей или специфической функциональности.

При разработке ресурсов для государственных компаний изучение законодательства — обязательный этап в работе над таким проектом. Например, сайты и порталы для государственных образовательных организаций должны быть сделаны с учётом требований Министерства образования.

**Оценивать риски**

Риски обязательно соотносятся с критичностью или некритичностью требований к проекту в целом.

Критичные требования отвечают за целевую функцию разрабатываемого продукта, включают необходимую инфраструктуру и возможности для разработки MVP.

К некритичным относятся остальные требования.

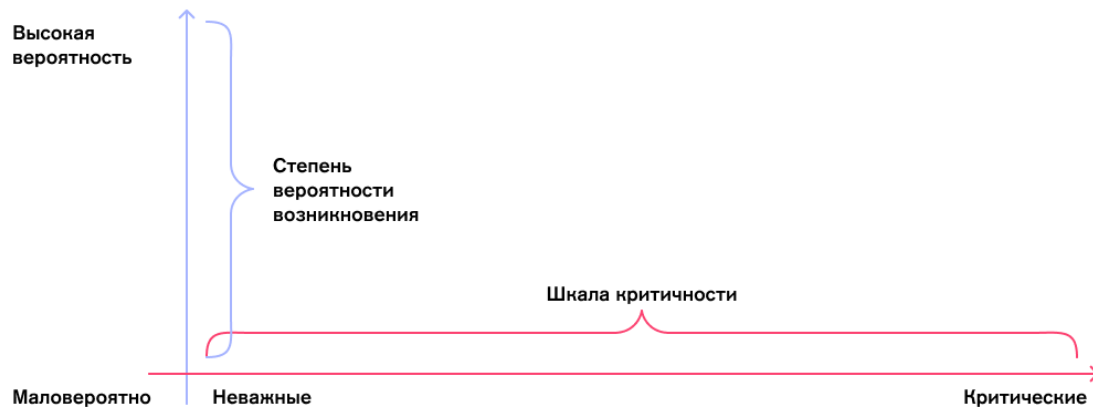
Критичные	Некритичные
Закуплено серверное оборудование, RFID-метки, лицензии	Приложение разрабатывается на Kotlin
Приложение для платформы Android с функцией NFC	Создание лендинга с презентацией приложения, наличие ссылки на скачивание приложения в Google Play
Приложение размещено в Google Play	Функционал лендинга расширен: добавлен заказ NFC-наклейки на пульт
Функция «радар» для поиска пульта	Функционал приложения расширен: добавлен чат-бот, оказывающий поддержку клиентам
Функция привязки пульта к приложению	
NFC-наклейки на пульт предоставлены для теста MVP членами рабочей группы	
Функционал для сбора обратной связи	

Пример критичных и некритичных требований

Для каждого риска выявляют следующее:

- источник — внутренний, внешний;
- вероятность его возникновения — редко, часто, очень часто;
- степень влияния на проект — слабое, среднее, сильное;
- степень управляемости — управляемый, неуправляемый или скрытый риск.

На базе этих параметров строится матрица рисков.



Шаблон для построения матрицы рисков. После классификации выявленных рисков их наносят на матрицу в виде точек. Такая наглядность поможет управлять рисками и вовремя их предупреждать

Остановимся подробнее на степени управляемости рисками.

Управляемые — риски, которые можно предугадать и акцентировать на них внимание уже на начальном этапе проекта, определив для них ответственных и держа под контролем.

Самый очевидный пример — опять же дизайн сайта. Сразу понятно, что на этапе согласования могут возникнуть сложности, доработки, переделки и бесконечные правки. Поэтому мы на старте проговариваем этот момент с заказчиком, согласовываем, кто будет лицом, принимающим окончательное решение, и письменно фиксируем максимально допустимое количество дней на согласование макета — в среднем, по нашему опыту, это обычно 5. Неуправляемые — возможные риски, появление которых мы не можем предотвратить.

Например, один из неуправляемых рисков — более ранний выпуск конкурентами на рынок похожего продукта.

Другой пример — подключение платёжной системы. В описании API говорится, что в ответ на запрос придёт одна строка, а по факту

приходит три. И платёжная система на своей стороне не готова вносить правку, потому что на это нужно время, при этом тысячи пользователей этой системы об этом не просили. Так как нужна именно одна строка, разработчику приходится самостоятельно решать этот вопрос, тратить больше запланированного времени на разработку и фиксацию данной ошибки.

Управляемые	Неуправляемые
Переход на новую платформу (доступы и обучение)	Выпуск подобного продукта конкурентом
Высокая нагрузка разработчика на другом проекте	Скрытые требования для реализации функционала считывания RFID-меток
Неверно выбранная технология реализации	Обстоятельства непреодолимой силы
Увеличение сроков закупки необходимого серверного оборудования, RFID-меток, лицензий, предоставления доступов (например, к JIRA, Confluence), согласования результатов работ (например, дизайн)	
Увеличение стоимости	
Высокая нагрузка заказчика операционными задачами	
Несоответствие ожиданиям	

Пример классификации рисков по степени их управляемости

## Запланировать мероприятия по предотвращению рисков

Для предотвращения рисков нужно понимать, как от них защищаться.

Грамотно выстроенные коммуникации внутри проекта играют главную роль для защиты от рисков и оказывают огромное влияние на успех проекта. Это означает, что все процессы взаимодействия регламентированы, роли всех членов команды закреплены и всё это зафиксировано в понятном для участников формате.

Важно сразу выстроить коммуникацию с заказчиками проекта, создать большую команду из представителей обеих сторон и все вопросы решать совместно. Для этого фиксируем ответственных для каждого этапа проекта в матрице «Роли и зоны ответственности» (RACI Matrix).



В матрице предусмотрены четыре роли:

- **Responsible** (кратко обозначается R или O на примере ниже) — ответственный за выполнение задач; например, дизайнер, разработчики, контент-менеджер;
- **Accountable** (A или C) — ответственный за весь проект, эту роль может занимать только один человек на одной задаче; например, менеджер проекта или тимлид;
- **Consulted** (C или Y) — консультант — сотрудник или группа, с которыми проводятся консультации по реализации проекта и мнение которых должно учитываться; например, представители заказчика, которые помогают правильно реализовать задачи;
- **Informed** (I или И) — сотрудники, которых информируют о выполнении конкретной задачи, но сами они никакого влияния на проект не оказывают; такую роль может выполнять, например, PR-менеджер, которого информируют о ходе проекта, чтобы он(а) мог(ла) готовить материалы для СМИ.

Этап/Документ	Роль/Сотрудник										
	Основатель стартапа	Отдел маркетинга	Отдел закупок	IT-архитектор	Product manager	Project manager	Бизнес-аналитик	Разработчик	Web-разработчик	UX/UI-дизайнер	QA
Этап 1 (MVP)											
Архитектура	У	И	С	О	И	И	С	И			
Концепция	У	И	С	С	С	О	И	И		И	И
Бюджет	У	И	С	С	С	О	И	И			
Договоры	У	И	О	С	И	И	И	И			
Закупки	У	И	О	И	И	И	И	И			
ТЗ и прототипы	У	И		С	С	С	О	И		И	И
Дизайн макетов	У	И			С	С	И	И		О	И
Разработка	У	И		И	С	С	И	О		И	И
Тестирование / Устранение ошибок	У	И		И	С	С	И	О		И	О
Демонстрация MVP	У	И	И	И	С	О	И	И		И	И
Старт этапа 2	У	С		С	О	С	И	И		И	И
Этап 2 (Лендинг)											
Концепция	У	И		И	С	О	И	И		И	И
ТЗ и прототипы	У	И		И	С	С	О			И	И
Дизайн макетов	У	И			С	С	И		И	О	И
Разработка	У	И		И	С	С	И		О	И	
Тестирование / Устранение ошибок	У	И			С	С	И		О	И	О
Демонстрация лендинга	У	И			С	О	И		И	И	И
Старт этапа 3	У	С		И	О	С	И		И	И	И
Этап 3 (Чатбот)											
Концепция	У	И		И	С	О	И	И		И	И
ТЗ и прототипы	У	И		И	С	С	О			И	И
Дизайн макетов	У	И			С	С	И		И	О	И
Разработка	У	И		И	С	С	И		О	И	
Тестирование / Устранение ошибок	У	И			С	С	И		О	И	О
Демонстрация	У	И			С	О	И		И	И	И
Завершение проекта	У	С		И	О	С	И		И	И	И

Пример RACI Matrix.

Однако просто зафиксировать ответственных мало — нужно проанализировать заполненную матрицу, чтобы не было перекоса в одну роль:

- много ячеек задач у ответственного — правильно ли распределены обязанности?
- много ячеек у исполнителей, ответственных за результат, — не много ли ответственности висит на одной роли?
- отсутствие пустых ячеек в матрице — действительно ли эти роли нужны на каких-то этапах?
- более одного ответственного — по условиям матрицы эта роль должна быть только у одного человека;
- нет ответственного — нужно найти сотрудника на эту роль;
- много исполнителей, ответственных за результат, — проверить, действительно ли это необходимо на этом участке, так как при большом количестве есть риск, что задача вообще не будет сделана;
- много консультантов — оценить целесообразность такого количества консультантов, будет ли это эффективно?
- нет консультантов и информируемых — правильно ли выстроены коммуникации?

После того, как разобрались с матрицей ролей, проактивно управляем рисками — фиксируем их и то, что нужно сделать для предотвращения.

Вместе с этим определяем план и способ коммуникаций на протяжении всего проекта: сколько раз в неделю и месяц, каким составом и при наступлении каких событий будем проводить созвоны или очные встречи, какой формат отчётности примем для фиксации договоренностей на этих встречах.

Встречи	Частота и тип встречи	Формат отчётности
Собрания с проектной командой	Ежедневные митапы (общение команды по проекту в неформальном ключе) Еженедельные планёрки (корректировка планов на текущую неделю, отчет по проделанной работе)	Письмо с итогами и достигнутыми договорённостями
Собрания с рабочей группой	Созвон или встреча раз в две недели Демонстрация наработанного по итогам каждого спринта	Письмо с итогами и достигнутыми договорённостями, ссылки на документы и регламенты
Встречи с руководством, маркетологом, PM и PR	По датам ключевых вех (фиксируется на старте проекта)	Письмо с итогами и достигнутыми договорённостями, ссылки на документы и регламенты

### Пример планирования коммуникаций на проекте

Также определяем инфраструктуру проекта:

- через какой канал будем взаимодействовать (например, Microsoft Teams, Skype);
- где будем хранить и вести документацию (например, Confluence);
- какие трекеры используем (например, Jira, Bitrix24).

## Предусмотреть действия при наступлении рисков

Стоит предусмотреть резервный план (contingency plan) на случай непредвиденных обстоятельств.

Задача этапа — выполнить его наиболее эффективным образом, а также собрать и проанализировать информацию о наступившем риске на будущее.

Не стоит тратить время на поиск виноватых — лучше предпринять всё необходимое, чтобы двигаться дальше и продолжать реализацию проекта. На этом этапе главная роль принадлежит коммуникации и поиску оптимального решения.

В нашей практике был проект, когда нужно было разделить один сайт, реализованный на Bitrix, на два. Один из них надо было интегрировать со штатной версией 1С.

Заказчик предполагал, что на разработку на своей стороне их специалисту понадобится три недели. По факту оказалось, что ранее созданная интеграция полностью кастомная — комплекты, цвета, размеры, остатки — и на стороне заказчика нет эксперта, который смог бы выполнить задачу. Пришлось подключиться нам и помочь с решением по интеграции. Дополнительно были пересмотрены сроки проекта.

---

## **Мониторить риски**

Ситуация на проекте постоянно меняется — это нужно принять как аксиому.

Например, если работать по Agile, то перед каждым новым спринтом могут меняться параметры проекта, обрисованные на этапе обсуждения. Следовательно нужно отслеживать изменения параметров рисков, корректируя внутрикомандный перечень топ-10 рисков.

На этапе мониторинга — если возвратиться к примеру с Agile, то это нужно делать в начале каждого спринта — важно отслеживать новые риски, изменять статус выявленных рисков, корректировать планы.

На протяжении проекта перечень может значительно измениться.

К примеру, на крупном проекте на финальном этапе в команде заказчика появляется эксперт, который задаёт много вопросов и предъявляет новые требования к проекту, которые подразумевают изменение архитектуры проекта и кода. Если инициировать эти изменения, то есть риск сорвать плановые сроки реализации проекта. Как вариант — провести встречу, обсудить, что беспокоит эксперта в текущей архитектуре, объяснить, почему реализовано именно так и наметить дальнейшие шаги. При этом важно постараться не увеличить объём работ проекта и не выйти за временные и финансовые рамки.

## **Каждый участник команды должен понимать важность работы с рисками**

Ключевой момент управления рисками — их постоянное отслеживание и предотвращение, в идеале согласованное с длительностью циклов разработки проекта.

Оценка рисков и работа с ними зависят от размера и длительности проекта. Рекомендуем возвращаться к работе над рисками один раз в 1–2 недели, в некоторых крупных проектах периодичность может быть увеличена до месяца.

Лучше составить неполный перечень рисков, чем не составить его вовсе.

В процессе общения с разработчиками или заказчиком лучше забыть фразу «все и так всё понимают»:

- не всегда написанное и услышанное понимается одинаково: в этом случае помогут уточняющие вопросы и прозрачная процедура взаимодействия, о которой говорили выше;
- часто заказчики не понимают, что короткая правка в пределах 10 минут может потянуть за собой сдвиг проекта на месяц: придётся тратить время на внесение правок в документацию, макеты, код, корректировку стоимости проекта.

Главное в работе с рисками — донести до всех участников проектной команды необходимость работы с рисками. Прежде всего нужно стараться не допускать наступления рисков, анализировать прошлый опыт и причины наступления для составления перечня возможных рисков в новых проектах, предотвращать непредусмотренные риски.