

Чеклист IT Project Manager

Практическая база для ведения проекта: цели, требования, план, команда, риски, коммуникации, релизы и контроль результата.

Инициация

На старте менеджер переводит идею в управляемую задачу: зачем проект нужен, какой результат считается ценным, кто принимает решения и какие ограничения уже известны.

Сформулировать цель проекта

Зафиксировать, какую проблему решает проект и какой измеримый результат нужен бизнесу.

Что это дает: Цель защищает команду от выполнения набора задач без понятного эффекта. Она помогает принимать решения по scope, срокам и приоритетам.

Как выполнить:

1. Проведите интервью с заказчиком и владельцем продукта.
2. Сформулируйте цель в формате результата, а не активности.
3. Привяжите цель к метрике: деньги, скорость, качество, удержание, снижение риска.

Критерии приемки:

- Цель записана в Project Charter.
- Есть метрика успеха и период оценки.
- Stakeholders подтвердили формулировку.

Определить stakeholders и роли

Понять, кто влияет на проект, кто принимает решения и кому нужно регулярно давать статус.

Что это дает: Карта stakeholders снижает риск поздних правок, конфликтов и решений без ответственного владельца.

Как выполнить:

1. Составьте список участников: бизнес, продукт, дизайн, разработка, QA, аналитика, поддержка.
2. Разделите роли по RACI: responsible, accountable, consulted, informed.
3. Согласуйте канал и частоту коммуникации для каждой группы.

Критерии приемки:

- Есть карта stakeholders.
- У каждого ключевого решения есть accountable.
- Команда знает, кого подключать при споре.

Зафиксировать ограничения проекта

Собрать ограничения по срокам, бюджету, команде, технологиям, безопасности и внешним зависимостям.

Что это дает: Ограничения помогают не строить нереалистичный план и заранее объяснить trade-off между сроком, объемом и качеством.

Как выполнить:

1. Проверьте дедлайны, бюджет, доступность команды и юридические требования.
2. Запишите технические ограничения: legacy, API, инфраструктура, релизные окна.
3. Отметьте, какие ограничения жесткие, а какие можно пересматривать.

Критерии приемки:

- Ограничения внесены в паспорт проекта.
- По каждому ограничению понятен риск.
- Команда согласовала, какие компромиссы допустимы.

Требования

Требования должны быть понятны бизнесу, дизайну, разработке и QA. Хорошая формулировка снижает количество переделок и спорных трактовок.

Разложить требования по типам

Отделить бизнес-требования, пользовательские сценарии, функциональные и нефункциональные требования.

Что это дает: Структура требований помогает не смешивать цель, решение, ограничения и критерии качества в одну неуправляемую формулировку.

Как выполнить:

1. Соберите исходные ожидания из брифа, интервью и текущих проблем.
2. Разделите требования по типам и уберите дубли.
3. Свяжите каждое требование с целью проекта или риском.

Критерии приемки:

- Требования сгруппированы.
- Дубли и противоречия отмечены.
- Есть связь с целью или пользовательским сценарием.

Написать критерии приемки

Для каждой значимой задачи указать условия, при которых работа считается выполненной.

Что это дает: Acceptance criteria уменьшают споры на приемке и дают QA основу для тест-кейсов.

Как выполнить:

1. Опишите ожидаемое поведение системы в конкретных условиях.
2. Добавьте негативные сценарии, ограничения и edge cases.
3. Проверьте критерии с разработчиком и QA до старта работы.

Критерии приемки:

- У ключевых задач есть acceptance criteria.
- Критерии проверяемы.
- QA может построить тесты без догадок.

Управлять score и изменениями

Согласовать, что входит в проект, что не входит и как команда обрабатывает новые запросы.

Что это дает: Scope management защищает сроки и качество от незаметного расширения объема работ.

Как выполнить:

1. Зафиксируйте in scope и out of scope.
2. Создайте правило change request: влияние на срок, бюджет, риск и ценность.
3. Регулярно показывайте stakeholders последствия новых запросов.

Критерии приемки:

- Есть список in/out of scope.
- Новые запросы проходят оценку влияния.
- Изменения согласуются до попадания в работу.

Планирование

План показывает последовательность поставки, ключевые зависимости, буферы, владельцев работ и точки контроля. Это не обещание без изменений, а инструмент управления.

Собрать roadmap поставки

Разложить проект на этапы, milestones и понятные поставки результата.

Что это дает: Roadmap помогает видеть, когда команда получит проверяемую ценность, а не просто завершит внутренние задачи.

Как выполнить:

1. Разбейте проект на логические поставки.
2. Для каждого этапа укажите результат, владельца и критерий готовности.
3. Покажите stakeholders, какие решения нужны на каждом milestone.

Критерии приемки:

- Roadmap опубликован.
- Milestones имеют результат и дату.
- Команда понимает ближайший фокус.

Оценить трудоемкость и риски оценок

Получить оценки команды и явно показать неопределенность по сложным задачам.

Что это дает: Оценки становятся полезными, когда рядом с ними есть допущения, риски и неизвестные зоны.

Как выполнить:

1. Попросите команду оценивать не только время, но и неопределенность.
2. Разделите крупные задачи на исследование и реализацию.
3. Добавьте буфер для интеграций, согласований и исправлений.

Критерии приемки:

- Оценки согласованы исполнителями.
- Высокорисковые задачи выделены.
- Есть план уточнения неизвестных зон.

Построить карту зависимостей

Понять, какие задачи, команды, сервисы и решения блокируют друг друга.

Что это дает: Карта зависимостей помогает заранее увидеть критический путь и не обнаружить блокеры в день релиза.

Как выполнить:

1. Отметьте зависимости между backend, frontend, design, QA, DevOps и внешними системами.
2. Выделите критический путь.
3. Назначьте владельца для каждой внешней зависимости.

Критерии приемки:

- Зависимости видны в плане.
- По каждой зависимости есть владелец.
- Критический путь обсужден с командой.

Поставка

В поставке важно видеть реальный прогресс, быстро поднимать блокеры, защищать фокус команды и не терять связь между задачами и целью проекта.

Вести регулярный статус проекта

Давать stakeholders короткую и честную картину прогресса, блокеров и следующих решений.

Что это дает: Регулярный статус снижает тревожность, ускоряет решения и делает проблемы видимыми до того, как они станут кризисом.

Как выполнить:

1. Обновляйте статус в одном формате каждую неделю.
2. Показывайте прогресс, риски, блокеры и решения, которые нужны от stakeholders.
3. Отделяйте факты от прогнозов.

Критерии приемки:

- Статус публикуется по расписанию.
- Есть список блокеров и владельцев.
- Stakeholders понимают состояние проекта без созвона.

Выявлять и снимать блокеры

Регулярно проверять, что мешает команде двигаться, и добиваться решения блокеров.

Что это дает: Менеджер создает скорость не контролем ради контроля, а устранением препятствий и ясностью приоритетов.

Как выполнить:

1. На daily и async-статусах спрашивайте не только прогресс, но и препятствия.
2. Назначайте владельца блокера и срок следующего шага.
3. Эскалируйте блокер, если команда не может решить его сама.

Критерии приемки:

- Блокеры видны в общем списке.
- У каждого блокера есть владелец.
- Старые блокеры не висят без действия.

Вести журнал решений

Фиксировать важные решения, причины, альтернативы и последствия.

Что это дает: Decision log защищает команду от повторных споров и помогает новым участникам понять, почему проект устроен именно так.

Как выполнить:

1. Записывайте решение сразу после обсуждения.
2. Указывайте контекст, варианты, выбранный путь и owner.
3. Связывайте решение с задачами и требованиями.

Критерии приемки:

- Ключевые решения сохранены.
 - Понятно, кто принял решение.
 - Команда может восстановить контекст через месяц.
-

Релиз и развитие

Релиз завершает не только разработку, но и цикл проверки: что запущено, кто принял результат, какие метрики изменились и что нужно улучшить дальше.

Подготовить релизный план

Согласовать, что именно выходит, кто проверяет, как откатываться и как команда узнает об успехе.

Что это дает: Release plan снижает риск хаотичного запуска и помогает быстро реагировать на проблемы после выкладки.

Как выполнить:

1. Соберите release scope и чеклист готовности.
2. Определите владельцев проверки, коммуникации и rollback.
3. Подготовьте план мониторинга после релиза.

Критерии приемки:

- Release plan согласован.
- Есть rollback plan.
- Команда знает окно релиза и критерии stop/go.

Провести ретроспективу

Собрать выводы после этапа или релиза и превратить их в конкретные улучшения процесса.

Что это дает: Ретроспектива полезна только тогда, когда из нее выходят действия, владельцы и сроки.

Как выполнить:

1. Соберите факты: что получилось, что мешало, где были потери.
2. Выберите 1-3 улучшения, которые реально внедрить.
3. Назначьте владельцев и дату проверки изменений.

Критерии приемки:

- Есть список action items.
- У каждого улучшения есть владелец.
- На следующем цикле проверяется выполнение.

Измерить результат проекта

Сравнить фактический эффект с исходной целью и baseline.

Что это дает: Измерение результата показывает, была ли поставка полезной, и помогает отличить завершение задач от достижения эффекта.

Как выполнить:

1. Вернитесь к метрикам из Project Charter.
2. Сравните baseline, план и факт.
3. Объясните отклонения и предложите следующий шаг.

Критерии приемки:

- Есть post-release отчет.
- Метрики связаны с целью.
- Выводы понятны stakeholders.