Зависимости задач в /task: блокирующие задачи, списки «Ждёт завершения» и «Блокирует», замок на карт

22.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

В проектах редко бывает так, что все задачи независимы. Обычно одна работа опирается на результат другой: макет — на исследование, релиз — на тестирование, отчёт — на собранные данные. Если эту связь не зафиксировать в системе, команда узнаёт о блокере слишком поздно. В /task механизм зависимостей решает именно эту проблему: он не даёт закрыть задачу, пока не завершены блокирующие, и показывает связи в карточке. В статье разберём, как устроены блокирующие задачи, списки «Ждёт завершения» и «Блокирует», что такое замок на карточке и как всё это использовать без лишней бюрократии.

Живое демо /task

Ниже интерактивная демонстрация /task: можно переключать виды, открывать задачи и пробовать действия. Это поможет увидеть, как зависимости выглядят в интерфейсе до того, как вы настроите их в своём проекте.

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что такое блокирующие задачи и зачем они нужны

Блокирующая задача — это работа, без завершения которой нельзя корректно закрыть другую задачу. В /task такая связь называется зависимостью типа «блокирует»: задача А блокирует задачу В, если В нельзя считать выполненной до завершения А.

Зачем это нужно на практике:

- **Порядок работ.** Зависимости делают последовательность явной: не нужно держать в голове, что отчёт нельзя сдавать до согласования методики.
- **Защита от преждевременного закрытия.** Если исполнитель закроет задачу раньше блокиратора, проект получит ложный сигнал «всё готово». Замок на карточке предотвращает это.
- **Прозрачность для команды.** В списках видно, кто кого ждёт и что уже разблокировано.
- **Планирование.** По зависимостям проще строить таймлайн и понимать критический путь.

Важно не превращать зависимости в тотальный контроль. Блокировка нужна там, где результат одной задачи действительно влияет на другую. Если связи сомнительные, лучше их не добавлять — иначе система начнёт тормозить работу.

Списки «Ждёт завершения» и «Блокирует»: кто кого ждёт

В карточке задачи зависимости отображаются в двух списках. Они симметричны, но отвечают на разные вопросы.

Список «Ждёт завершения»

Здесь перечислены задачи, которые должны быть завершены до текущей. Это блокираторы для выбранной задачи. Например, у задачи «Собрать отчёт» в этом списке будут «Согласовать методику» и «Выгрузить данные». Пока они не закрыты, текущая задача остаётся заблокированной.

Список «Блокирует»

Здесь перечислены задачи, которые зависят от текущей. Это те работы, которые ждут завершения выбранной задачи. Для «Согласовать методику» в этом списке будет «Собрать отчёт». Такой список помогает понять, что произойдёт после закрытия задачи и кого нужно уведомить.

Оба списка доступны в карточке задачи. По ним можно быстро перейти к связанной задаче и проверить её статус. Если блокиратор ещё в работе, текущая задача не завершится — сработает замок.

> Практическое правило: перед тем как брать задачу в работу, откройте список «Ждёт завершения». Если там есть незакрытые блокираторы, сначала разберитесь с ними или пересмотрите зависимость.

Замок на карточке: как работает запрет завершения

Замок на карточке — это визуальный индикатор и техническое ограничение, которое не позволяет завершить задачу, пока не закрыты все блокирующие задачи. В интерфейсе он обычно выглядит как значок замка рядом со статусом или кнопкой завершения.

Что происходит при попытке закрыть заблокированную задачу:

1. Система проверяет список «Ждёт завершения».
2. Если хотя бы один блокиратор не завершён, действие блокируется.
3. Пользователь видит сообщение о том, какие задачи нужно закрыть.
4. После завершения всех блокираторов замок снимается, и задачу можно закрыть.

Замок не мешает работать с задачей: менять описание, добавлять комментарии, вложения, чек-листы. Ограничение касается только перевода в завершающий статус. Это важно: команда может готовить результат заранее, но не может формально закрыть работу раньше времени.

Если зависимость ошибочна, её можно удалить. Но делать это стоит осознанно: возможно, проблема не в замке, а в том, что блокиратор действительно не завершён.

Как настроить зависимости: пошагово

Настройка занимает несколько минут и не требует специальных прав, кроме доступа к редактированию задачи.

1. **Откройте задачу**, которая должна ждать другую.
2. **Найдите блок зависимостей** в карточке — списки «Ждёт завершения» и «Блокирует».
3. **Добавьте блокиратор** в список «Ждёт завершения»: выберите задачу, которая должна быть завершена раньше.
4. **Проверьте обратную связь**: добавленная задача появится в списке «Блокирует» у блокиратора.
5. **Убедитесь, что замок отображается** на карточке зависимой задачи.

6. **Закройте блокиратор** и проверьте, что замок снялся и задача стала доступна для завершения.

Если нужно выстроить цепочку из нескольких задач, добавляйте зависимости последовательно. Не создавайте циклов: задача A не должна ждать B, если B уже ждёт A. /task не даст сохранить такую связь, но лучше продумать последовательность заранее.

Плюсы подхода и типичные ошибки

Блокировки и замок дают команде несколько выгод:

- **Меньше сюрпризов.** Незавершённый блокиратор виден до того, как задача попадёт в релиз.
- **Честный статус.** Закрытая задача действительно означает выполненную работу, а не формальную отметку.
- **Быстрая коммуникация.** Списки показывают, кого ждать и кого разблокирует текущая задача.
- **Управляемость сроков.** Зависимости помогают увидеть критический путь и вовремя перераспределить ресурсы.

Типичные ошибки, которых стоит избегать:

- **Слишком много зависимостей.** Если блокировать всё подряд, команда будет постоянно ждать. Добавляйте только реальные связи.
- **Забытые блокираторы.** Задача может месяцами висеть в «Ждёт завершения» из-за потерянной связи. Периодически просматривайте такие списки.
- **Обход замка через смену статуса.** Если в процессе есть промежуточные статусы, убедитесь, что завершающий статус действительно защищён.
- **Игнорирование обратного списка.** «Блокирует» помогает не забыть уведомить смежников после завершения задачи.

Зависимости в /task — это не бюрократия, а инструмент прозрачности. Они особенно полезны в проектах, где несколько команд работают последовательно, и в релизных циклах, где цена преждевременного закрытия высока. Настройте блокировки там, где это действительно важно, и замок на карточке станет страховкой, а не помехой.

Источник: https://cms.dolinais.ru/zavisimosti-zadach-v-task-blokiruyuschie-zadachi-spiski-zhdet-zaversheniya-i-blokiruet-zam