Зависимости задач в /task: как блокировки защищают сроки

20.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

Живое демо /task

Ниже — интерактивная демонстрация /task: можно переключать виды, открывать задачи и пробовать действия с блокировками.

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что такое зависимости задач и зачем они нужны

В управлении проектами зависимости задач — это связи, при которых одна задача не может быть начата или завершена, пока не выполнена другая. В /task такие связи реализованы через механизм блокировок: задача-блокиратор должна быть закрыта, иначе зависимая задача остаётся заблокированной. Это помогает избежать ситуаций, когда исполнитель берётся за задачу, не имея на руках результатов предыдущего этапа, и в итоге сроки срываются.

Блокировки особенно важны в проектах, где работа идёт последовательно: разработка после дизайна, тестирование после сборки, публикация после согласования. Без формальных зависимостей участники полагаются на устные договорённости, а это источник ошибок и задержек.

Три ключевых элемента: блокирующие задачи, списки и замок

Механизм зависимостей в /task состоит из трёх взаимосвязанных частей. Рассмотрим каждую.

Блокирующие задачи (blockers)

Блокирующая задача — это задача, которая должна быть завершена до того, как зависимая от неё задача может быть закрыта. В /task вы можете указать блокираторы при создании или редактировании задачи. После этого система автоматически отслеживает их статус: пока блокиратор не закрыт, зависимая задача не может быть завершена.

Важно: блокировка не мешает начать работу над задачей (например, подготовить черновик), но запрещает её завершение. Это гибкий подход: исполнитель может планировать время, но не может формально закрыть задачу, пока не получен результат от предыдущего этапа.

Списки «Ждёт завершения» и «Блокирует»

Для удобства отслеживания зависимостей в карточке задачи есть два списка:

- **«Ждёт завершения» —** задачи, которые блокируют текущую. Пока хотя бы одна из них не закрыта,

текущая задача остаётся заблокированной.
- **«Блокирует» —** задачи, которые зависят от текущей. Они не смогут быть завершены, пока вы не закроете эту задачу.

Эти списки дают быстрый обзор: вы сразу видите, что мешает завершить задачу и что от неё зависит. Это помогает расставлять приоритеты и вовремя замечать узкие места.

Замок на карточке

Замок — визуальный индикатор того, что задача заблокирована. Он появляется на карточке задачи, если есть хотя бы один незакрытый блокиратор. Замок сигнализирует: «Эту задачу нельзя завершить, пока не выполнены условия». При наведении или открытии карточки видно, какие именно задачи блокируют завершение.

Замок не блокирует редактирование или комментирование — он влияет только на возможность завершения. Это сделано для того, чтобы команда могла продолжать обсуждение и вносить правки, но не могла обойти зависимости.

Запрет завершения до закрытия блокираторов

Самый важный элемент — это правило, которое не позволяет завершить задачу, пока не закрыты все блокираторы. В /task это реализовано на уровне системы: при попытке завершить задачу с незакрытыми блокираторами вы получите предупреждение и не сможете это сделать.

Такое ограничение кажется жёстким, но оно решает несколько проблем:

- **Предотвращает ложное закрытие.** Исполнитель не может отметить задачу выполненной, если результат предыдущего этапа ещё не получен. Это снижает риск того, что на следующем этапе обнаружатся ошибки из-за недостающих данных.
- **Синхронизирует работу команды.** Все участники видят реальный статус: задача не закрыта, потому что ждёт завершения другой. Это прозрачность.
- **Защищает сроки.** Блокировки не дают «перескочить» через этап и потом вернуться к нему, тратя время на переделки.

  Важно: запрет завершения не означает, что задачу нельзя выполнять. Можно вести работу, но формальное закрытие произойдёт только после снятия блокировки.

Как работать с зависимостями: пошаговый пример

Рассмотрим типичный сценарий: есть задача «Разработать макет» и задача «Свёрстать страницу». Вторая зависит от первой.

1. Создайте задачу «Свёрстать страницу».
2. В карточке задачи в разделе зависимостей добавьте блокиратор — выберите «Разработать макет».
3. Теперь в списке «Ждёт завершения» отобразится «Разработать макет», а на карточке появится замок.
4. Исполнитель может начать верстать, но не может завершить задачу, пока макет не будет готов и закрыт.

5. Когда задача «Разработать макет» будет завершена, замок исчезнет, и «Свёрстать страницу» можно будет закрыть.

Такой подход гарантирует, что вёрстка не будет сдана без макета, а сроки не поедут из-за переделок.

Преимущества использования блокировок

- **Прозрачность.** Все видят, что задача заблокирована и почему. Нет необходимости выяснять статус в чатах.
- **Дисциплина.** Команда привыкает к порядку: сначала закрываем блокираторы, потом зависимые задачи.
- **Снижение рисков.** Меньше вероятность, что задача будет закрыта с неполными данными.
- **Ускорение.** Благодаря явным зависимостям участники не тратят время на ожидание или повторную работу.

Блокировки в /task — это не бюрократия, а инструмент, который помогает соблюдать сроки и поддерживать качество. Начните с малого: добавьте зависимости в ключевые задачи, и вы увидите, как процесс становится предсказуемее.

Источник: https://cms.dolinais.ru/zavisimosti-zadach-v-task-kak-blokirovki-zaschischayut-sroki-3