Зависимости задач в /task: как блокировки защищают сроки

19.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

Зависимости задач — это механизм, который формально фиксирует, что одна работа не может быть выполнена, пока не завершена другая. В **/task** он реализован через блокирующие задачи, два связанных списка на карточке («Ждёт завершения» и «Блокирует»), а также замок, который запрещает завершение, пока блокираторы открыты. Разберём, как это устроено, зачем нужно и как выстроить работу, чтобы не сорвать сроки.

Живое демо /task

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

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что такое блокирующая задача и два списка

Блокирующая задача (блокиратор) — это задача, которая должна быть завершена до того, как текущая задача может считаться выполненной. В /task связь двусторонняя и отображается на карточке в двух списках:

- **«Ждёт завершения» —** задачи, которые блокируют текущую. Пока хотя бы одна из них не закрыта, текущая не может быть завершена.
- **«Блокирует» —** задачи, которые зависят от текущей. Их завершение тоже ждёт, пока вы не закроете эту задачу.

По сути, это граф зависимостей, видимый с обеих сторон. Если задача А блокирует В, то на карточке А в списке «Блокирует» появится В, а на карточке В в списке «Ждёт завершения» — А. Такая симметрия помогает не потерять связь при работе с любой из задач.

Замок на карточке: как выглядит запрет

Когда у задачи есть хотя бы один незакрытый блокиратор, на её карточке появляется **замок** — визуальный индикатор и одновременно ограничение. Он сигнализирует: «Эту задачу нельзя завершить, пока не закрыты зависимости». При попытке завершить такую задачу система не даст сохранить статус «Готово» и покажет, какие именно блокираторы мешают.

> Замок — это не наказание, а защита от преждевременного закрытия. Он не мешает работать над задачей, но фиксирует, что результат ещё не полный.

Важно: замок не блокирует редактирование, комментарии или смену исполнителя. Он влияет только на завершение. Это позволяет команде продолжать работу, но не даёт формально закрыть задачу, пока не

выполнены предпосылки.

Зачем это нужно: три практических сценария

1. **Защита от «ложного готово».** Разработчик закрыл задачу «Верстка страницы», но макет от дизайнера ещё не утверждён. Без блокировки это выглядит как выполненная работа, а на деле результат может быть переделан. Замок заставляет сначала закрыть блокиратор «Утвердить макет».
2. **Прозрачность сроков.** Руководитель видит в списке «Ждёт завершения», что задача ждёт три других. Это честная картина: срок сдвинется, если блокираторы не будут закрыты вовремя. Без таких связей план выглядит оптимистичнее, чем есть.
3. **Согласованность между командами.** Задача маркетинга «Запустить лендинг» может блокироваться задачей разработки «Выложить сборку». Список «Блокирует» на стороне разработки показывает, кого они держат, и помогает приоритизировать.

Как настроить блокировки: пошагово

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

1. Откройте карточку задачи, которую нужно заблокировать (или которая будет блокировать).
2. В блоке зависимостей выберите добавление связи. В зависимости от интерфейса это может быть кнопка «Добавить блокировку» или выбор из списка.
3. Укажите вторую задачу. Система сама определит направление: если вы добавляете блокиратор к текущей задаче, он попадёт в «Ждёт завершения», а на карточке блокиратора — в «Блокирует».
4. Проверьте, что на карточке появился замок (если блокиратор ещё открыт).
5. Повторите для всех необходимых связей. Не создавайте циклических зависимостей: A блокирует B, B блокирует A — это тупик, который не даст закрыть ни одну из задач.

Если задача уже была завершена, а потом выяснилось, что она должна ждать блокиратор, система может потребовать переоткрыть её. Это нормально: зависимости — инструмент, который отражает реальность, а не статичный список.

Как работать с блокировками в повседневном процессе

- **Смотрите на список «Ждёт завершения» перед планированием.** Если задача стоит в спринте, а её блокираторы не в этом спринте, риск срыва высок. Либо переносите задачу, либо добавляйте блокираторы в текущую работу.
- **Используйте «Блокирует» для приоритизации.** Если задача блокирует три других, её стоит делать раньше. Это простой способ увидеть узкие места.
- **Не бойтесь снимать блокировку, если она устарела.** Зависимости меняются. Если блокиратор больше не нужен, удалите связь, чтобы не тормозить работу.
- **Разбивайте крупные блокираторы.** Если задача ждёт «Согласовать бюджет», а это долго, создайте подзадачу «Подготовить смету» и заблокируйте её — так работа пойдёт параллельно.

Плюсы и ограничения

Плюсы: меньше переделок, честные сроки, прозрачные связи между командами, автоматический контроль готовности. Замок не даёт закрыть задачу «на бумаге», что особенно полезно в распределённых командах.

Ограничения: механизм требует дисциплины — связи нужно поддерживать в актуальном состоянии. Если блокировки не обновлять, они превращаются в формальность. Также стоит избегать избыточных зависимостей: если блокировать всё подряд, команда будет простаивать. Зависимость оправдана, когда без результата блокиратора текущая задача действительно не может считаться выполненной.

Итог

Блокирующие задачи, списки «Ждёт завершения» и «Блокирует», а также замок на карточке — это не бюрократия, а инструмент управления рисками. Он делает зависимости явными и не даёт закрыть задачу раньше времени. Настройка проста, но эффект ощутим: меньше сюрпризов в конце спринта, понятные приоритеты и честный статус проекта.

Источник: https://cms.dolinais.ru/zavisimosti-zadach-v-task-kak-blokirovki-zaschischayut-sroki-2