Зависимости задач в /task: как блокировки защищают сроки

18.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

В управлении проектами часто возникает ситуация, когда одна задача не может быть выполнена, пока не завершена другая. Например, тестирование новой функции невозможно начать, пока разработчик не закончит код. В /task для таких случаев предусмотрен механизм зависимостей задач. В этой статье разберём, что такое блокирующие задачи, как работают списки «Ждёт завершения» и «Блокирует», зачем нужен замок на карточке и почему нельзя завершить задачу раньше закрытия блокираторов. Вы узнаете, как настроить зависимости и какие преимущества это даёт команде.

Живое демо /task

Ниже представлена интерактивная демонстрация возможностей /task. Вы можете переключать виды, открывать задачи и пробовать различные действия, чтобы лучше понять, как работают зависимости.

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что такое зависимости задач и блокирующие задачи

Зависимость задач — это связь между двумя задачами, при которой одна задача не может быть начата или завершена до выполнения другой. В /task такая связь реализуется через указание блокирующих задач.

Блокирующая задача (или блокиратор) — это задача, которая должна быть завершена, прежде чем станет возможным завершение зависимой от неё задачи. Например, задача «Написать документацию» может блокироваться задачей «Утвердить API».

Блокировки помогают избежать ситуаций, когда сотрудник берётся за задачу, но не может её выполнить из-за отсутствия необходимых результатов от коллег. Это особенно важно в командах, где работа организована по принципу потока (kanban) или в спринтах (scrum).

Списки «Ждёт завершения» и «Блокирует»

В карточке задачи в /task отображаются два списка, связанных с зависимостями:

- **«Ждёт завершения»** — содержит задачи, которые блокируют текущую задачу. Пока хотя бы одна из них не завершена, текущую задачу нельзя закрыть.
- **«Блокирует»** — содержит задачи, которые зависят от текущей. То есть текущая задача является блокиратором для них.

Эти списки позволяют быстро оценить, какие задачи влияют на текущую и на что влияет она сама. Например, если вы открыли задачу «Тестирование», в списке «Ждёт завершения» увидите «Разработка», а в списке «Блокирует» — «Релиз».

Замок на карточке: что это и зачем нужен

Если задача имеет незавершённые блокираторы, на её карточке появляется **замок —** визуальный индикатор, сигнализирующий о том, что задача заблокирована. Замок не позволяет завершить задачу (перевести в статус «Готово») до тех пор, пока все блокирующие задачи не будут закрыты.

Это защищает от случайного или намеренного преждевременного закрытия задачи, которое могло бы привести к нарушению процесса. Например, разработчик может отметить задачу «Написать код» как выполненную, но если код не прошёл ревью (блокиратор «Ревью»), система не позволит это сделать.

Замок также служит напоминанием: задача не может быть завершена, пока не выполнены предварительные условия. Это особенно полезно в распределённых командах, где участники могут не знать о зависимостях.

Запрет завершения до закрытия блокираторов

В /task реализован жёсткий запрет на завершение задачи, если у неё есть хотя бы один незавершённый блокиратор. При попытке перевести такую задачу в статус «Готово» система выдаст предупреждение и не позволит сохранить изменение.

Такой подход гарантирует соблюдение порядка выполнения работ. Однако важно правильно выстраивать зависимости, чтобы не создавать искусственных блокировок. Например, если задача «Согласовать макет» блокируется задачей «Подготовить макет», но макет ещё не готов, это может затормозить весь процесс. Поэтому рекомендуется пересматривать зависимости по мере необходимости.

Как настроить зависимости в /task

Чтобы добавить блокирующую задачу, выполните следующие шаги:

1. Откройте карточку задачи, которую нужно заблокировать.
2. В разделе «Зависимости» нажмите «Добавить блокирующую задачу».
3. Выберите из списка существующую задачу или создайте новую.
4. Сохраните изменения.

После этого задача появится в списке «Ждёт завершения», а на карточке отобразится замок. Аналогично можно добавить задачу, которую блокирует текущая, — она появится в списке «Блокирует».

Для удаления зависимости достаточно нажать на крестик рядом с задачей в соответствующем списке.

Преимущества использования зависимостей

- **Прозрачность —** все участники видят, какие задачи блокируют работу и на что влияет их задача.
- **Соблюдение порядка —** система не позволяет завершить задачу раньше времени, что снижает риск ошибок.
- **Планирование —** менеджеры могут заранее оценить критические пути и избежать простоев.
- **Автоматизация —** не нужно вручную проверять статусы связанных задач; /task делает это автоматически.

Использование зависимостей особенно эффективно в крупных проектах, где много взаимосвязанных задач. Это помогает поддерживать порядок и не забывать о важных этапах.

Заключение

Зависимости задач — мощный инструмент управления проектами в /task. Блокирующие задачи, списки «Ждёт завершения» и «Блокирует», замок на карточке и запрет завершения до закрытия блокираторов обеспечивают контроль над последовательностью работ. Настройка зависимостей проста, а преимущества ощутимы: меньше ошибок, больше прозрачности и лучшее соблюдение сроков. Попробуйте использовать этот механизм в своём следующем проекте.

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