

я внутри завода

Роль Project Manager
в системе потока



Я не управляю людьми.

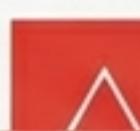
**Я управляю прохождением
заказа через ограничение.**





Что такое ТОС?

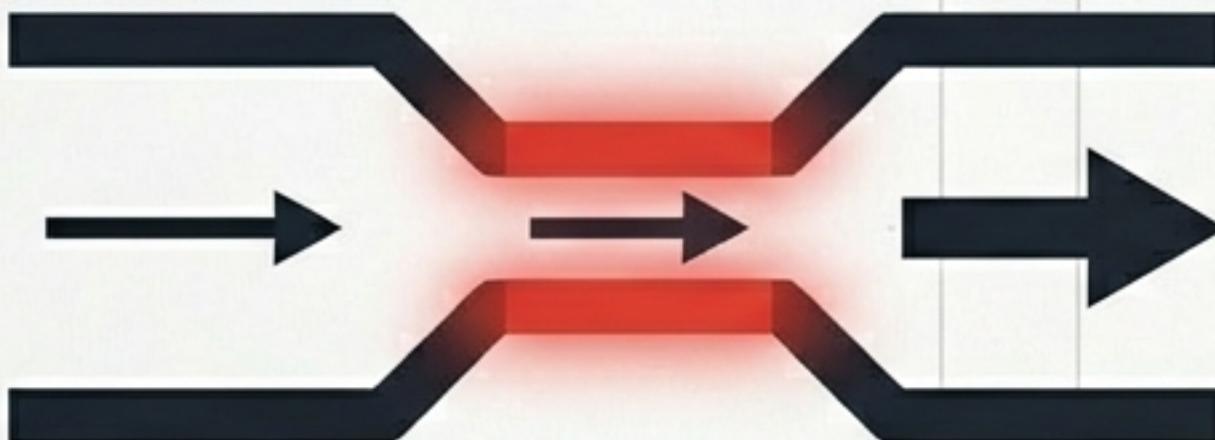
ТОС – Теория ограничений
(Theory of Constraints)



Главная идея:

В любой системе есть одно ограничение, которое определяет её скорость.

Если усиливать ограничение – ускоряется вся система.



Ключевые понятия

Throughput – скорость превращения работы в деньги

WIP незавершённая работа

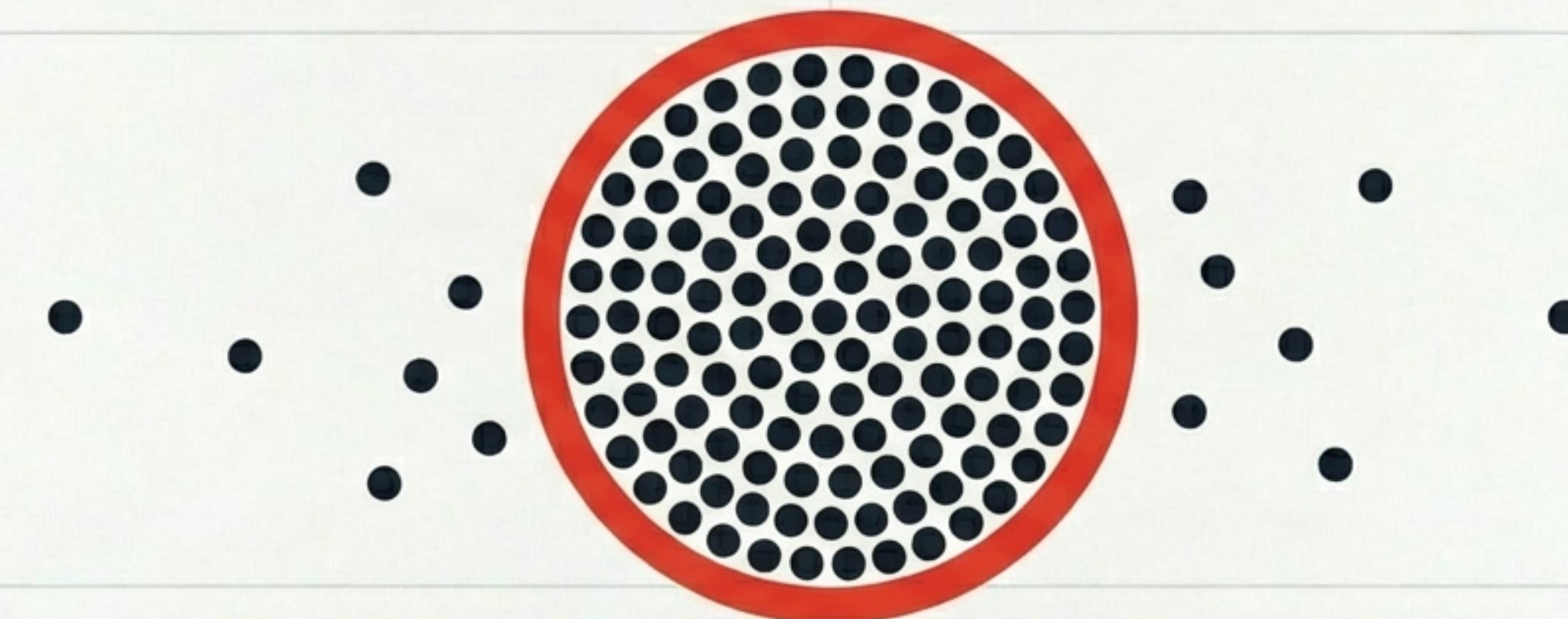
Ограничение самое узкое место системы, которое определяет ритм работы всего потока

**Throughput падает,
когда WIP растёт,
а ограничение
не управляется.**



Что такое ограничение

Ограничение – это ресурс или этап, который ограничивает скорость всей системы.



Примеры ограничений



На заводе:
станок,
который
работает
медленнее всех



В ИТ:
ключевой
разработчик,
инженер,
интегратор



В проекте:
решение
о приоритете,
которое
тормозит поток

Почему это важно:

Ограничение определяет ритм всей системы

- ➔ Если оно простоявает → теряется throughput
- ➔ Если оно перегружено → растёт WIP, хаос и задержки



Я – фильтр между заказом и производством.

Моя задача – защита узкого места и предсказуемый поток.

**Моя главная цель:
Не «чтобы все
были заняты».
А чтобы заказ
стал деньгами.**

Я обеспечиваю:
– Ритм,
– Синхронизацию,
– Защиту ограничения.

Что я обязана проверить и что должно быть на выходе

Проверка на входе:

- Готовность материалов
- Понятное ТЗ / спецификация
- Выделены ключевые ресурсы (разработчики, инженеры, QA, дизайнеры) и подтверждена их загрузка
- Доступ к инструментам и инфраструктуре
- Ясность результата
- Что считается «готово»
- Критерии приёмки у заказчика
- Цель проекта подтверждена
- Приоритет
- Почему заказ запускается сейчас
- Не перегрузит ли ограничение
- Реальная пропускная способность узкого места

Хороший выход:

- Заказ завершён
- Клиент принял результат (акт подписан)
- Ограничение стабильно работало
- WIP не вырос
- Команда готова к следующему циклу
- Нет накопленного технического долга



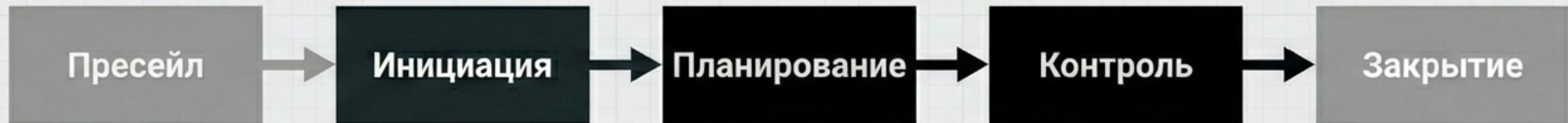
На каждом этапе я отвечаю не за действия, а за результат.

Пресейл (Результат)

- Заказ реально выполнить
- Учтена реальная мощность
- Видны узкие места
- Обещания клиенту синхронизированы с ресурсами
- Будущая просрочка не создаётся

Инициация проекта (Результат)

- Поток запускается осознанно
- Ключевые ресурсы подтвердили доступность
- Приоритет установлен
- Заказ встроен в очередь перед ограничением
- Приняты обязательства системой

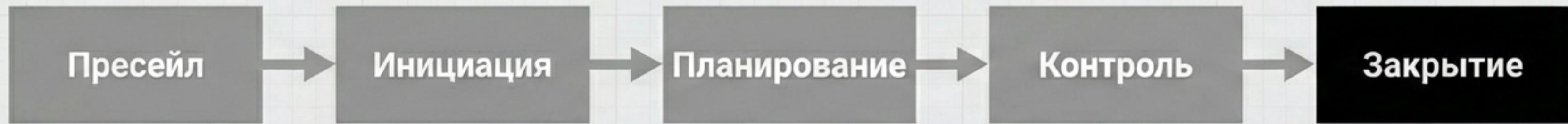


Планирование вокруг ограничения (Результат)

- Найдено узкое место
- Поток подчинён ритму ограничения
- Создан буфер (резерв времени/ресурсов перед ограничением)
- Убран лишний WIP
- Буфер = «подушка безопасности» для ограничения
- План защищает слабое звено и предотвращает хаос

Контроль исполнения (Результат)

- Ограничение не простирается
- Ограничение не захлебывается
- Блокеры снимаются оперативно
- Приоритеты ясны
- WIP контролируется
- Моя работа здесь – не тушить пожары, а не допускать их



Закрытие / передача (Результат)

- Работа превращена в деньги
- Мощность освобождена
- Люди переключены на новые проекты
- Выводы зафиксированы
- Поддержка структурирована

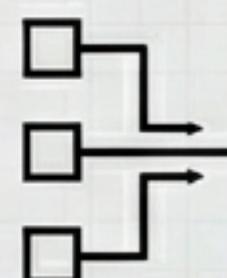
Хорошее закрытие = здоровая система, плохое = скрытая перегрузка будущего



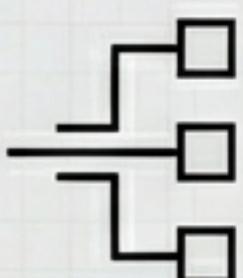
Скрытое ограничение моей роли

Возникает на пересечении приоритетов, решений и когнитивной нагрузки.

Причина: веду несколько потоков одновременно.



**Узким местом становится не станок или разработчик,
а я – РМ, принимающий решения.**



Если убрать меня на 2 недели:

- WIP вырастет
- Приоритеты станут хаотичными
- Блокеры зависнут
- Throughput упадёт
- Завод превратится в склад незавершённой работы
- Деньги перестанут поступать вовремя

Главный вывод: управление потоком = управление ограничением, а не людьми