

# Основные подходы к разработке продукта

## Вопросы:

1. Метод водопада (каскадный метод).
2. Метод гибкой разработки.
3. Теория решения изобретательских задач.
4. Теория ограничений.



# Основные подходы к разработке продукта

Основные подходы к разработке продукта различаются не столько сущностью, сколько последовательностью выполнения. Согласно теориям жизненного цикла продукта, его разработка включает в себя несколько основных этапов:

- Осознание потребности и генерация идей: Определение рыночной потребности и поиск новых идей для продукта.
- Планирование и анализ: Сбор и анализ требований, формирование технического задания (ТЗ) и бизнес-анализ.
- Проектирование: Разработка концепции, эскизного и технического проектов, создание 3D-моделей и основной конструкторской документации (КД).
- Создание прототипа (опытного образца): Изготовление опытного экземпляра на основе разработанной КД.
- Тестирование: Испытание опытного образца для проверки его соответствия заданным параметрам и выявление недостатков.
- Запуск и выход на рынок: Вывод готового продукта на рынок, его маркетинг и продажи.
- Непрерывное улучшение: Сбор обратной связи, доработка и внесение изменений в продукт для повышения его качества и конкурентоспособности.

На каждом этапе проводятся свои исследования, расчеты и работы по созданию конструкторской и технологической документации.

# Основные подходы к разработке продукта

Основные подходы к разработке продукта – это «водопад» (Waterfall), который предполагает последовательное выполнение этапов, и гибкие методологии (Agile), такие как Scrum и Kanban, ориентированные на *итеративность* (повторение одного и того же процесса или набора действий, чтобы постепенно достичь желаемого результата), быструю адаптацию и ценность для пользователя.

Кроме того, существует подход, основанный на принципах Lean (бережливое производство), который фокусируется на устраниении потерь и оптимизации процессов.

Выбор подходящего подхода зависит от конкретного проекта, команды и бизнес-целей. Agile-методологии, особенно Scrum, хорошо подходят для проектов с меняющимися требованиями, а «водопад» — для проектов с четко определенными и стабильными требованиями. Lean может быть интегрирован с другими подходами для оптимизации процессов.



# 1. Метод водопада (каскадный метод)

**Модель "водопад" (Waterfall)** — это последовательная (каскадная) методология разработки продукта, где каждый этап строго следует за предыдущим и не может начаться, пока предыдущий не завершен полностью. Этапы включают сбор требований, проектирование, разработку (реализацию), тестирование, развертывание и поддержку. Такая модель подходит для проектов, где требования стабильны и четко определены с самого начала.

## Основные принципы и этапы

- ✓ **Линейность:** Разработка идет строго сверху вниз, как по водопаду, без возврата на предыдущие стадии.
- ✓ **Строгая последовательность:** Следующий этап не начинается, пока предыдущий не завершен и не документирован.
- ✓ **Жесткость:** Невозможность внести изменения в процессе. Если в середине проекта нужны доработки, придется начинать заново, начиная с первого этапа.
- ✓ **Завершенность этапов:** Результаты каждого этапа согласовываются и фиксируются перед переходом к следующему.

## Этапы Waterfall



# 1. Метод водопада (каскадный метод)

Применение модели в работе включает в себя несколько обязательных этапов, пропускать и откатываться к которым после их завершения нельзя.

**Анализ требований.** В начале проекта проводится глубокий анализ всех требований к конечному продукту. Этот этап включает взаимодействие с заказчиком, изучение потребностей пользователей и фиксацию всех спецификаций в документации.

**Проектирование.** Разработчики на основе собранных требований создают детальный план работы. Определяются технологии, инструменты, решения и интерфейсы, а также сроки и бюджет проекта. Все это отражается в проектных документах, которые станут основой для следующих этапов.

**Разработка.** Только после завершения проектирования начинается процесс написания кода, дизайн интерфейса и другие работы. Команда работает строго по техническому заданию и документации, следуя заранее утвержденному плану.

**Тестирование.** Готовый продукт проверяют на соответствие требованиям. В этот момент выявляются ошибки, баги и несоответствия, которые разработчики исправляют перед запуском.

**Внедрение.** Когда продукт проходит все проверки, он передается заказчику или пользователям. Этот процесс может включать установку, интеграцию с другими системами и начальное обучение сотрудников.

**Поддержка.** После выпуска продукта команда занимается его обслуживанием: исправляет ошибки, обновляет компоненты и, при необходимости, дорабатывает функции. Но вносить крупные изменения сложно, ведь водопадная модель не подразумевает итеративного улучшения продукта.

# 1. Метод водопада (каскадный метод)

## Достоинства:

- Очень подробное документирование процесса на каждой стадии.
- Требования к продукту четко определены.
- Снижение требований к квалификации разработчиков.
- Страховка от дефектов разработки благодаря жесткому планированию.
- Легко измеримые результаты каждой стадии.
- Логично «встраивается» в полный жизненный цикл продукта.

## Примеры успешного применения:

1. Модель Waterfall подойдет, если все требования к конечному результату строго установлены заранее и гарантированно не изменятся во время работы:
  - ✓ Строительство зданий и инфраструктуры. Каждый этап — проектирование, закупка материалов, возведение — требует строгого следования плану.
  - ✓ Производство оборудования. При разработке промышленных машин, автомобилей или бытовой техники важно заранее определить все параметры конструкции и требования к конечному изделию.
  - ✓ Госзаказы и тендерные проекты. Часто требуют жесткого соответствия техническому заданию и соблюдения этапов выполнения.

# 1. Метод водопада (каскадный метод)

## Примеры успешного применения (продолжение):

2. Проекты с ограниченными ресурсами и высокой ценой ошибки. В критически важных отраслях модель Waterfall снижает риски за счет тщательного планирования и тестирования перед внедрением:

- ✓ Авиакосмическая отрасль. При разработке самолетов или космических аппаратов каждая ошибка может стоить жизни людей. Все расчеты и тесты должны быть завершены до запуска.
- ✓ Медицина и фармацевтика. При создании медицинского оборудования или лекарств необходимо строгое соблюдение норм безопасности, а тестирование проходит в несколько жестко регламентированных этапов.
- ✓ Военные и оборонные разработки. Требуется надежность и соответствие строгим техническим стандартам.

## Недостатки:

- Отсутствие гибкости. Каскадная модель жизненного цикла проекта отлично работает там, где, требования известны заранее и не меняются. Но на практике заказчики часто пересматривают свои требования к результату прямо в процессе работы. Например, Разработка ПО: клиент хочет добавить новый функционал уже на этапе тестирования, но из-за последовательной структуры Waterfall это приведет к переделке всей архитектуры. Это увеличивает сроки и затраты.
- Долгий срок вывода продукта на рынок. Пока проект проходит все этапы, ситуация на рынке может измениться, и продукт устареет еще до релиза. Например: разработка мобильного приложения по модели Waterfall занимает 1–2 года. За это время появляются новые тренды, меняются пользовательские предпочтения, и конкуренты выпускают аналогичный, но более актуальный продукт.

# 1. Метод водопада (каскадный метод)

## Недостатки (продолжение):

- Высокие риски провала. Waterfall не дает возможности быстро протестировать гипотезы, а значит, если на финальном этапе выяснится, что продукт не соответствует ожиданиям пользователей, исправить ситуацию будет сложно. Например: при разработке онлайн-сервиса по Waterfall после запуска выясняется, что интерфейс неудобен или функции не востребованы, но менять продукт поздно. В Agile же продукт тестируется итерационно, что позволяет избежать подобных ситуаций.

Метод существует с 1970-х годов и соответствует представлению об уровнях готовности продукта или технологии. Для преодоления недостатков имеется ряд улучшающих (доработанных) моделей.

Водопадный метод подходит для коммерческих сделок, в которых договоры подписаны и деньги заплачены. Но при работе на внутренних клиентов труднее трезво относиться к вносимым в последний момент изменениям, когда о них просят люди из вашей собственной организации, имеющие поддержку руководства.



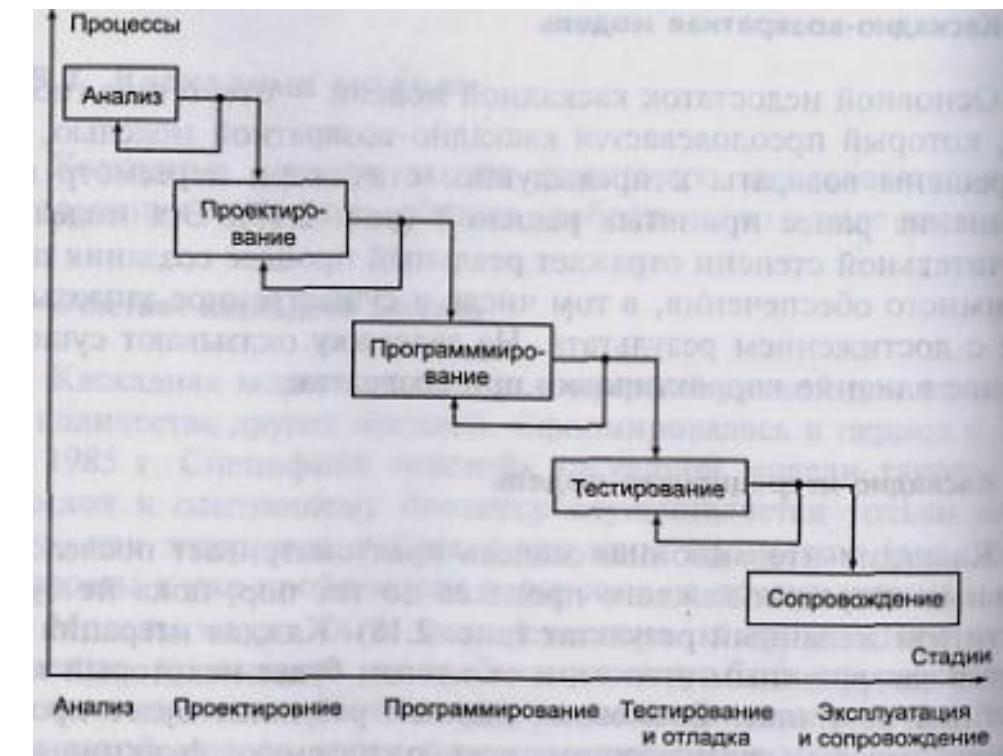
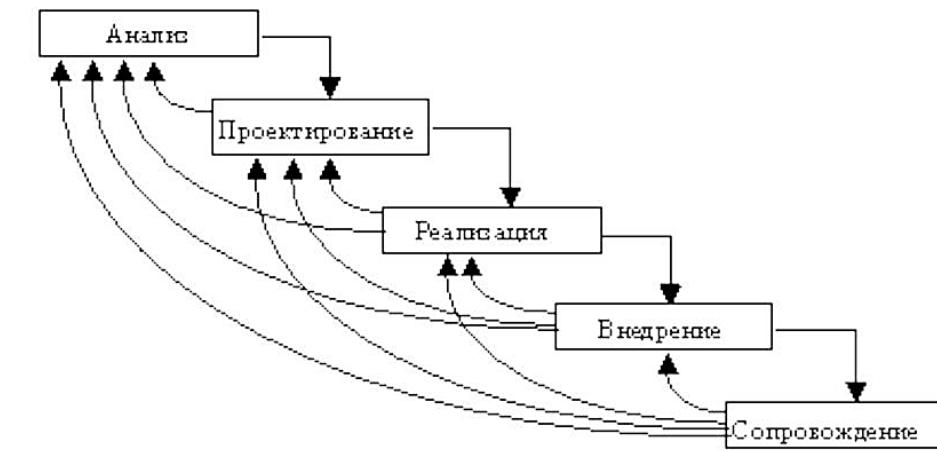
# 1. Метод водопада (каскадный метод)

## Каскадно-возвратная

Основной недостаток каскадной модели — отсутствие гибкости, который преодолевается **каскадно-возвратной моделью**, где разрешены возвраты к предыдущим стадиям и пересмотр или уточнение ранее принятых решений. Эта модель в значительной степени отражает реальный процесс создания программного обеспечения, в том числе и существенное запаздывание с достижением результата. На задержку оказывают существенное влияние корректировки при возвратах.

**Каскадно-итерационная модель** предусматривает последовательные итерации каждого процесса до тех пор, пока не будет достигнут желанный результат. Каждая итерация является завершенным этапом, и ее итогом будет некоторый конкретный результат. Возможно, данный результат будет промежуточным, не реализующим всю ожидаемую функциональность.

**Каскадная модель с перекрывающимися процессами** предполагает наличие таких специализированных команд, позволяющих до определенной степени сократить передаваемую документацию. Следующий процесс начинается до завершения текущего. Более того, несколько процессов могут выполняться параллельно.



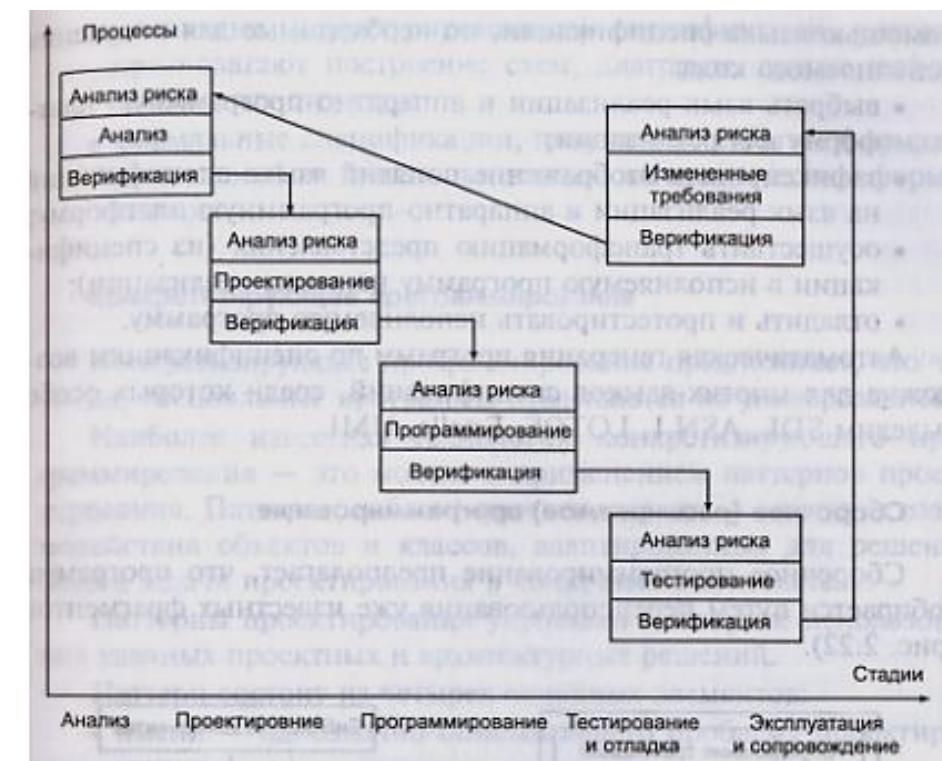
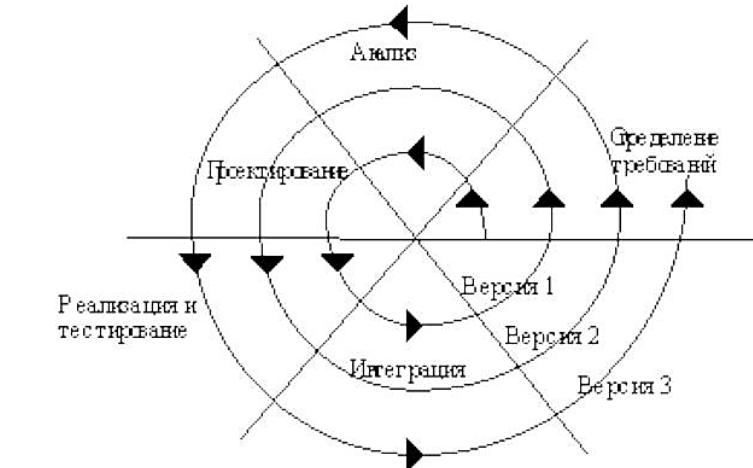
# 1. Метод водопада (каскадный метод)

# Сpirальная модель

**Сpirальная модель** была предложена в середине 80-х годов XX в. с целью сократить возможный риск разработки. Сpirальная модель использует понятие прототипа — программы, реализующей частичную функциональность создаваемого продукта. Создание прототипов осуществляется за несколько витков спирали, каждый из которых состоит из «анализа риска», «некоторого процесса» и «верификации».

Обращение к каждому процессу предваряет «анализ риска», причем если риск превышения сроков и стоимости проекта оказывается существенным, то разработка заканчивается. Это позволяет предотвратить более крупные денежные потери в будущем.

Особенность спиральной модели — в разработке итерациями. Причем каждый следующий итерационный прототип будет обладать большей функциональностью.



## 2. Метод гибкой разработки

**Гибкая методология разработки (agile-методы)** — методология, основанная на коротких итерациях с динамическим переопределением требований на каждом этапе и производимая самоорганизующимися рабочими группами из специалистов различного профиля.

Этот метод существенно больше соответствует современной концепции жизненного цикла продукта на рынке, описанного выше. Метод основан на том, что:

- разработка разбивается на короткие итерации;
- общий «образ» продукта (MVP) возникает достаточно рано, и в процессе доработки и общения с клиентом он уточняется и улучшается.

Вместо отдельных групп, отвечающих за каждую стадию разработки, работа ведется в кросс-дисциплинарных группах с участием маркетологов, специалистов по работе с потребителями, системных архитекторов, инженеров, программистов, тестировщиков и специалистов по User Experience.

**MVP, Minimal Viable Product, или Minimal Valuable Product**, — минимальная версия продукта, которая уже имеет для потребителя ценность.

**Customer development** — методология итеративного развития стартап-компаний, состоящая в том, что нужно максимально быстро создать MVP-продукт для проверки его восприятия клиентами, тестирования бизнес-модели и каналов продаж.

Проверка приводит к циклическому усовершенствованию продукта и бизнес-модели.

**User Experience (UX)** — это восприятие и ответные действия пользователя, возникающие в результате использования и/или предстоящего использования продукции, системы или услуги (ISO 9241-210).

Как правило, для гибкого подхода Agile характерна работа короткими итерациями по две-три недели. Внутри каждой итерации собрана серия задач: анализ, проектирование, непосредственно работа и тестирование. После каждой итерации команда анализирует результаты и меняет приоритеты для следующего цикла.

## 2. Метод гибкой разработки

### Основные принципы и ценности Agile:

- ✓ Люди и взаимодействие важнее процессов и инструментов: Команды работают вместе, а не следуют жестким инструкциям.
- ✓ Работоспособный продукт важнее исчерпывающей документации: Приоритет отдается созданию рабочего продукта, а не подготовке объемных документов.
- ✓ Сотрудничество с заказчиком важнее согласования условий договора: Постоянная обратная связь с клиентом помогает лучше понять его потребности.
- ✓ Готовность к изменениям важнее следования первоначальному плану: Команды готовы адаптироваться к новым требованиям и изменениям на рынке, вместо того чтобы жестко придерживаться первоначального плана.

### Популярные Agile методологии

- ✓ Scrum: Фреймворк, в котором работа ведется короткими, фиксированными по времени итерациями (спринтами), включающими все этапы от анализа до тестирования.
- ✓ Kanban: Метод, использующий визуальную доску для отслеживания прогресса задач. Он фокусируется на непрерывном потоке работы и ограничении незавершенной работы.
- ✓ Lean: Применение принципов бережливого производства для минимизации потерь и максимизации ценности для клиента. Lean часто используется как дополнение к другим Agile-подходам.
- ✓ Экстремальное программирование (Extreme Programming, XP): Методология, уделяющая особое внимание инженерным практикам, таким как парное программирование и разработка через тестирование (TDD).



## 2. Метод гибкой разработки

### Достоинства:

- Быстрое возникновение «нулевого» приближения к продукту.
- Гибкий учет изменяющихся требований клиента на каждой фазе итераций.
- Нужны разработчики высокой квалификации.
- Требует меньше доработок из-за вовлеченности клиента.

### Недостатки:

- Не выглядит так «солидно», как жесткая каскадная схема.
- Некоторые клиенты не готовы идти на высокую вовлеченность в процесс разработки.
- Нет долгосрочного подробного плана.
- Менее подробная документация и стандартизация продукта.

# Где используют гибкие методологии

Agile — идеальный подход для стартапов и небольших проектов на заказ. Тогда большинство минусов сходят на нет — отсутствие структуры не мешает, заказчик сам заинтересован в тесном общении, команда редко меняется, а внедрение занимает меньше времени.

А вот если проект масштабный и тянется долгие месяцы, минусы уже выходят на первый план и мешают реализовать проект так, как нужно.

Если говорить о сферах бизнеса, то изначально Agile создавали именно для применения в командах разработки ПО, игр и интерфейсов. Сейчас его используют Google, Netflix, Microsoft, Spotify, Ericsson, Dell, Adobe и большинство других IT-компаний, как гигантов индустрии, так и совсем мелких стартапов.

Но потом преимущества Agile оценили по достоинству другие компании. Сейчас отдельные принципы этого семейства применяют практически везде, а иногда и всю работу выстраивают по гибкой проектной методологии Agile.

## Сравнение подходов Waterfall и Agile

| Критерий           | Waterfall  | Agile   |
|--------------------|--|---|
| Структура процесса | Линейная и строгая последовательность этапов.                                      | Итеративная, работа ведется циклами (спринтами).  |
| Гибкость           | Требования фиксируются заранее, изменения на последующих этапах сложны и затратны. | Гибкий процесс, допускаются корректировки на любом этапе.                                 |
| Сроки              | Жестко определены с самого начала.   | Корректируются в зависимости от прогресса.  |
| Документация       | Приоритет на детальную документацию перед началом работы.                          | Минимально необходимая документация, упор на взаимодействие команды.                      |
| Обратная связь     | Пользователь тестирует продукт только на финальном этапе.                          | Регулярная обратная связь от заказчика и пользователей на каждом спринте.                 |
| Риски              | Высокий риск обнаружения ошибок на поздних стадиях.                                | Риски минимизируются за счет постоянного тестирования и корректировки.                    |
| Командная работа   | Четкое разделение ролей, взаимодействие строго регламентировано.                   | Самоорганизующиеся команды, активное сотрудничество, свобода внесения предложений и идей. |
| Клиентское участие | Участвует только на этапе постановки задач и финального тестирования.              | Постоянное взаимодействие с клиентом, возможность корректировки проекта в процессе.       |

# Оценка уровня готовности технологии

TPMRL — Technology, Product, Manufacturing Readiness Levels — уровни готовности технологии, продукта, производства.

В промышленности доминирует каскадный метод, который позволяет существенно уменьшить риски дефектов ранних стадий разработки, которые приводят к очень дорогостоящим последствиям на дальнейших стадиях жизненного цикла продукта. Ошибка, не исправленная на предыдущей стадии, требует для исправления на следующей стадии в 10 раз больше затрат. Именно такой подход, однако, приводит к тому, что корпорации внутри себя разрабатывают продукты крайне медленно.

## **Уровни готовности технологии:**

1. Фундаментальные исследования выявили потенциал применения.
2. Определены возможные применения.
3. Получено экспериментальное подтверждение возможности применения на модели.
4. Прототип испытан в лабораторных условиях.
5. Прототип испытан в условиях, близких к реальным условиям.
6. Компоненты системы испытаны в реальных условиях.
7. Прототип всей системы прошел проверку в эксплуатационных условиях.
8. Система испытана и сертифицирована.
9. Штатная эксплуатация и сопровождение.

### 3. Теория решения изобретательских задач (ТРИЗ)

Для ситуаций, в которых требования к продукту содержат противоречия, удобно использовать методологию ТРИЗ.

ТРИЗ как методология изобретательства была предложена Генрихом Сауловичем Альтшуллером. Это советский (а позднее российский) инженер-изобретатель, писатель-фантаст, который разработал ТРИЗ, используя собственный изобретательский опыт и наблюдения за работой других изобретателей.

Интересно, что популярность ТРИЗ в технологически развитых странах существенно выше, чем на родине этой методологии — в России. Это связано, прежде всего, с тем, что высококонкурентная бизнес-среда развитых стран заставляет компании использовать самые эффективные методы ускорения и улучшения качества разработок новых продуктов. Основными понятиями теории решения изобретательских задач являются:

- ✓ Изобретательская задача (проблема).
- ✓ Изобретательская идея.
- ✓ Методология генерации эффективных идей и разрешения проблем на основе моделей противоречий.



Альтшуллер Г.С.  
(1926–1998)

### 3. Теория решения изобретательских задач (ТРИЗ)

Выделяются три основных типа противоречий:

1. Административное противоречие. Нужно что-то сделать, а как это сделать — неизвестно. Такие противоречия лежат на поверхности, их не нужно выявлять, но и их «подсказывательная» сила равна нулю.
2. Техническое противоречие. В глубине административных противоречий лежат технические: если улучшить одну часть системы, недопустимо ухудшится другая часть. Техническое противоречие иногда нужно выявлять, но зато оно может помочь отбросить много пустых вариантов решения. Пример: продукт стал прочнее (лучше), но тяжелее (хуже!).
3. Физическое противоречие. Каждое техническое противоречие обусловлено физическим противоречием: к одной и той же части системы предъявляются взаимно противоположные требования.

Согласно ТРИЗ, ключом к решению проблемы является снятие [системного противоречия](#). При всей внешней простоте данной идеи ее реализация может быть очень сложна. Тем не менее за рубежом ТРИЗ широко используется высокотехнологичными компаниями и даже, по утверждению журнала *Forbes*, стала одним из основных факторов, обеспечивших успех корпорации *Samsung* в конкурентной борьбе с *Apple* на рынке мобильных устройств. (см. О роли ТРИЗ в обеспечении конкурентных преимуществ компании *Samsung* [Электронный ресурс]. URL:

<http://www.forbes.com/sites/haydnshaughnessy/2013/03/07/why-is-samsung-such-an-innovative-company>)

### 3. Теория решения изобретательских задач (ТРИЗ)

#### Суть и принципы метода ТРИЗ

Цель: Находить нестандартные, творческие и эффективные решения проблем с минимальными усилиями.

Основа: Теория основана на изучении закономерностей развития систем и создании на этой основе инструментов для решения задач.

Противоречия: Один из центральных методов — выявление и разрешение противоречий в системе, когда улучшение одной части приводит к ухудшению другой. Для этого используются специальные решающие механизмы.

#### Приёмы решения:

- ✓ Дробление: Разбиение задачи на более мелкие части.
- ✓ Вынесение: Перенос части системы или функции в другое место.
- ✓ Объединение: Соединение нескольких частей или функций в одну.
- ✓ «Наоборот»: Делать всё наоборот, чтобы обойти проблему.
- ✓ «Посредник»: Использование промежуточного элемента для выполнения действия или преодоления препятствия.
- .

### 3. Теория решения изобретательских задач (ТРИЗ)

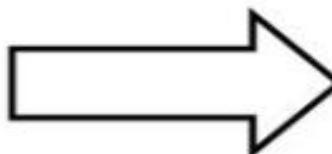
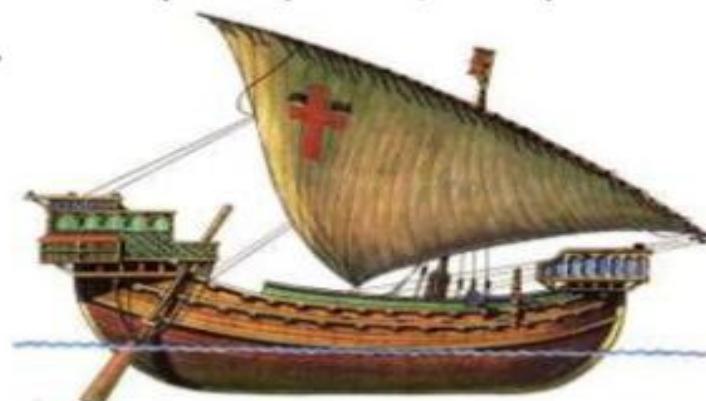
#### Инструменты и этапы решения

- ✓ Формулировка задачи: Чёткое определение проблемы и её формализация.
- ✓ Первичная обработка: Анализ задачи с использованием начальных инструментов.
- ✓ Моделирование противоречий: Поиск и формализация противоречий в системе.
- ✓ Разрешение противоречий: Применение приёмов и решающих механизмов для устранения противоречий.
- ✓ Системная сборка: Проверка и доработка полученного решения.

# Мини-задача

Минимальная задача может быть получена из ситуации по формуле: **то, что есть, минус недостаток**, или **то, что есть, плюс требуемое достоинство** (новое качество). Таким образом, минимальная задача получается из ситуации введением предельных ограничений на изменение исходной технической системы.

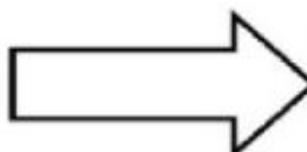
Альтшуллер Г.С., Творчество как точная наука, М.: "Советское радио". 1979 г.,  
с.



# Макси-задача

Максимальная задача, наоборот, получается предельным снятием ограничений: **исходную систему разрешается заменить принципиально новой системой.**

Альтшуллер Г.С., Творчество как точная наука, М., "Советское радио", 1979 г., с. 44-45



# Изобретательская задача

"... обычная задача переходит в разряд изобретательских в тех случаях, когда необходимым условием ее решения является устранение **технического противоречия**.

Нетрудно создать Новую машину, игнорируя технические противоречия. Но тогда машина окажется неработоспособной и нежизненной".

Альтшуллер Г.С., Алгоритм изобретения, М., "Московский рабочий", 1973 г., с.

# Причины трудности изобретательской задачи

- 1) «сначала мы имеем дело не с задачей, а с изобретательской ситуацией - целым клубком задач, и нужно каким-то образом выделить из этого клубка **единственно правильную задачу**;
- 2) пытаясь решить задачу обычными (известными, привычными) путями, мы наталкиваемся на техническое противоречие, и нужно каким-то образом **докопаться до спрятанного в его глубине физического противоречия**;
- 3) чтобы устранить физическое противоречие, нужно каким-то образом **найти подходящий технический прием или физический эффект**.

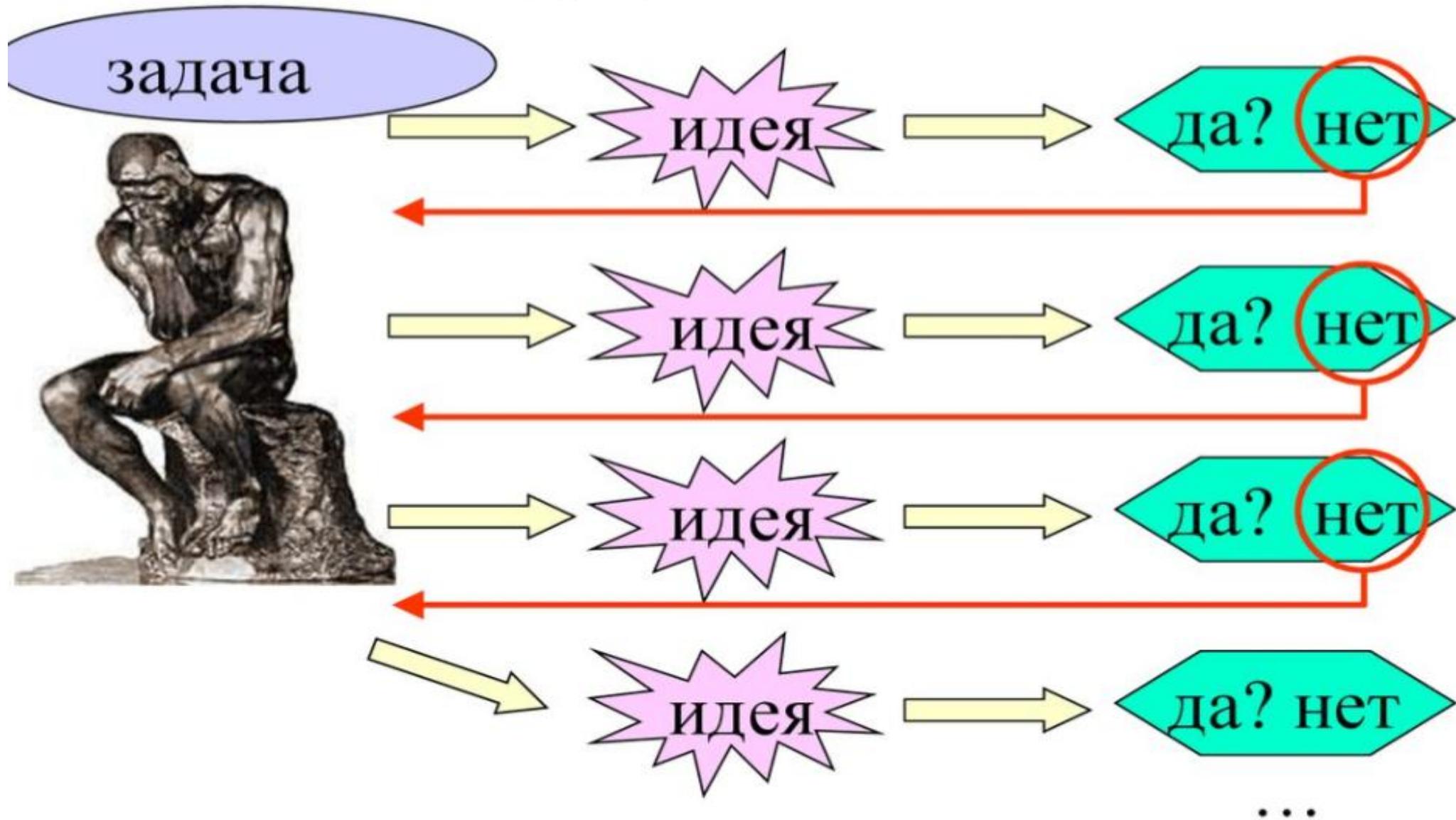
Альтшуллер Г.С., Селюцкий А.Б., Крылья для Икара: Как решать изобретательские задачи, Петрозаводск, "Карелия", 1980 г., с. 48.

## Параметры задачи высшей трудности

1. Предложена не задача, а весьма нечеткая ситуация.
2. Условия задачи содержат неправильные требования и указания.
3. Условия задачи изложены чересчур кратко.
4. Ответ на задачу включает использование физического эффекта, неизвестного изобретателю.

Альтшуллер Г.С., Селюцкий А.Б., Крылья для Икара: Как решать изобретательские задачи, Петрозаводск, "Карелия", 1980 г., с. 93.

# Метод проб и ошибок



## Совершенствование метода проб и ошибок

- увеличение числа "проб", например, с помощью большого числа людей, одновременно работающих над проблемой;
  - замена вещественных проб - мысленными;
- создание т.н. "Неалгоритмических методов".

## Неалгоритмические методы

К подобным методам Г.С. Альтшуллер относил:

- мозговой штурм Алекса Осборна;
- морфологический анализ Фрица Цвикки;
- синектику Уильяма Гордона;
- метод фокальных объектов,
- а также их комбинации.

## The stages of Syneetics

1. Постановка задачи
2. Сделать неизвестное известным.
3. Перевод задачи, "как она поставлена" в задачу, "как она понимается".
- 4 Вывявление вопроса, вызывающего аналогии.
5. Работа по поиску аналогий.
6. Использование аналогий:
  - Прямая аналогия
  - Символическая аналогия
  - Личностная аналогия
  - Фантастическая аналогия
7. Поиск возможностей перевода найденных аналогий и образов в предложения по решению поставленной задачи.

Перевод незнакомого в знакомое:

- Понимание проблемы;
- Большая вариативность решений – из них получение чего-то нового.

Перевод знакомого в незнакомое:

- Изменить привычный взгляд на вещи – посмотреть с другой стороны.

## Алгоритм синектики

1. Найти параметры продукта, важные для потребителя.
2. Найти антоним для одной из них.
3. Сформулировать противоречие объекта

Объект: дверь

1. позволяет войти.
2. не позволяет войти.
3. дыра, не позволяющая войти



# Метод фокальных объектов

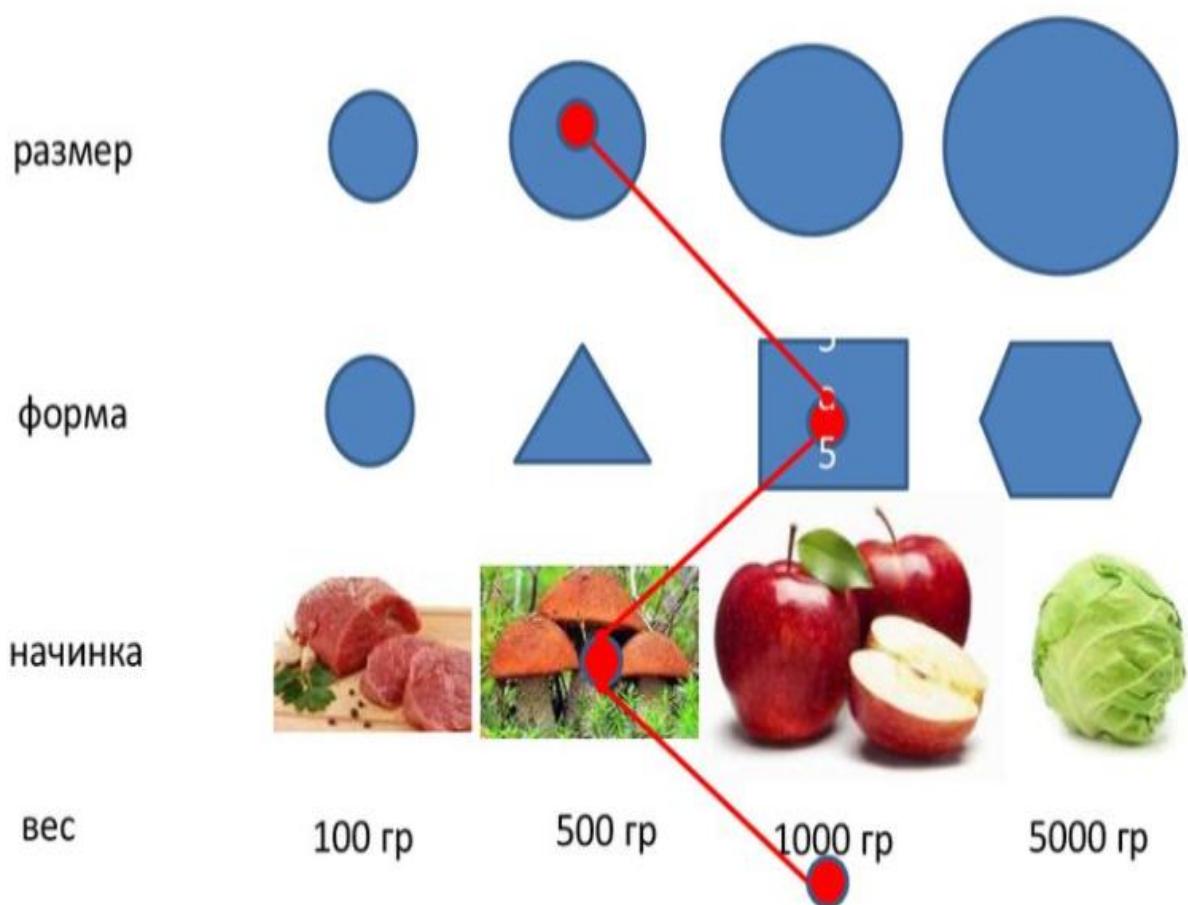
## Алгоритм

- Из условий задачи выделить объект (прототип), подлежащий усовершенствованию (ФО), уточнить цель.
- Выбрать 3-4 случайных объектов
- Выписать для каждого из них несколько характерных признаков (свойств).
- Полученные признаки перенести на прототип (фокальный объект) – получить новые сочетания.
- Новые сочетания развить путем свободных ассоциаций. Зафиксировать все интересные идеи.
- Оценить новые идеи и отобрать наиболее эффективные с точки зрения реализации. Сформулировать задачи на разработку новых модификаций объекта.

# Морфологический анализ

1. Точно сформулировать проблему, подлежащую решению.
2. Выявить и охарактеризовать все параметры, которые могли бы войти в решение заданной проблемы.
3. Сконструировать морфологический ящик или многомерную матрицу, содержащую все решения заданной проблемы.
4. Все решения, содержащиеся в морфологическом ящике внимательно проанализировать и оценить с точки зрения целей, которые должны быть достигнуты.
5. Выбрать и реализовать наилучшие решения (при условии наличия необходимых средств).

## Морфологический анализ. Как создать пирог?



### 3. Теория решения изобретательских задач (ТРИЗ)

Пример формулирования проблем по ТРИЗ:

Проблема: При наклеивании новых обоев из стен удаляют шурупы, на которых крепились ковры, книжные полки и т. д. Возможно, что придется вернуть некоторые предметы на старое место. Как найти отверстия в стене, но так, чтобы на новых обоях метки не были видны?

Административное противоречие: Не найти отверстия в стене.

Техническое противоречие: Если не наносить метки, то потом будет не найти отверстия. Если метки будут нанесены, то ухудшится внешний вид.

Физическое противоречие: Метка на стене должна быть, чтобы знать, куда вставить шуруп, и метки быть не должно, так как она портит внешний вид стены.

Возможное решение: Можно в отверстие поместить железосодержащую деталь (гайку, гвоздь) и использовать намагниченные железные опилки.

.

## 4. Теория ограничений

**Теория ограничений Голдрата (ТОС)** — это методология управления, которая фокусируется на поиске и устранении "узких мест" (ограничений), замедляющих достижение цели системы.

Основная идея заключается в том, что эффективность всей системы определяется её самым слабым звеном, и управление этим ограничением — ключ к повышению производительности.

ТОС предлагает пятишаговый процесс: идентификация ограничения, его использование, подчинение других процессов ограничению, расширение или усиление ограничения и возврат к шагу 1, если ограничение изменилось.

Основные принципы:

- ✓ Ограничение — главный фактор: Успешность всей системы зависит от её самого слабого звена.
- ✓ Фокус на узком месте: Для улучшения работы нужно сосредоточить усилия на проблемах в ограничении, а не на улучшении всех остальных частей системы.
- ✓ Цель системы: Применение методологии для достижения главной цели, например, увеличение прибыли в бизнесе.



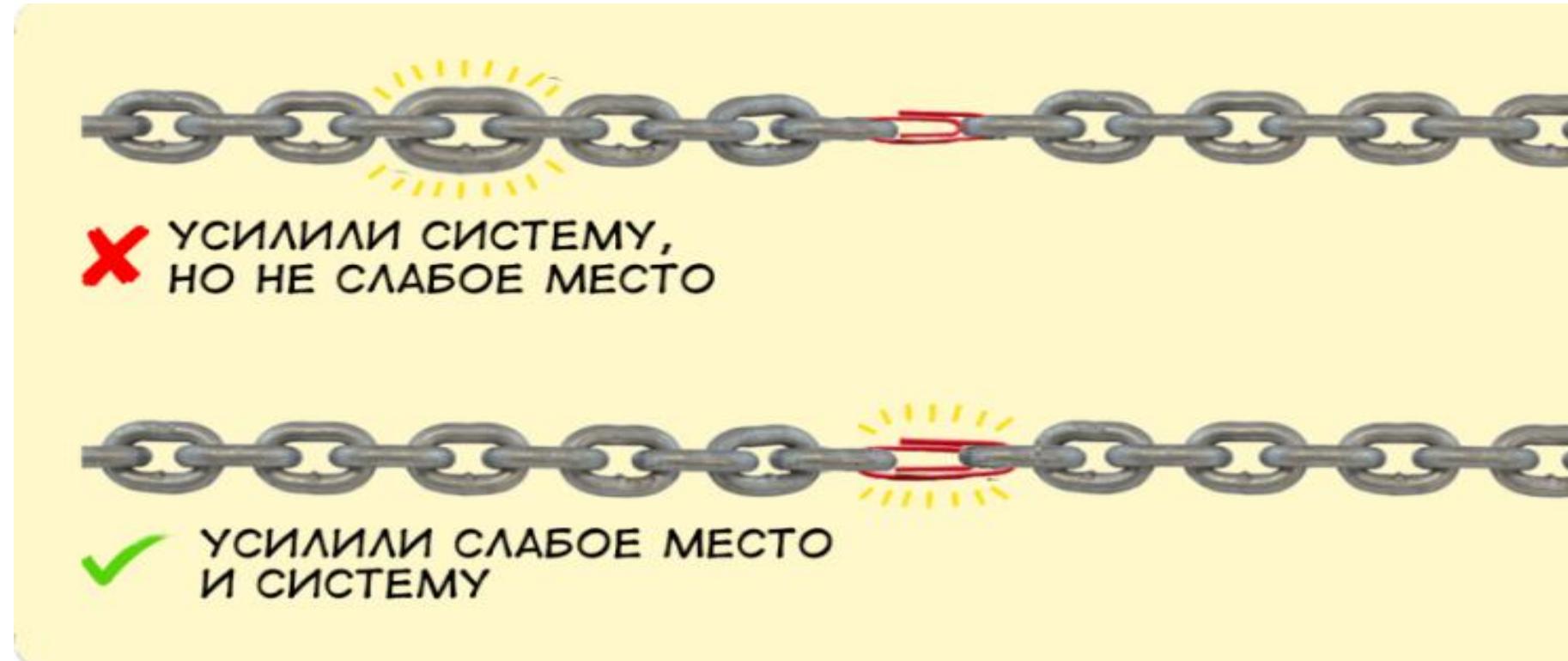
Голдратт Э.  
(1947-2011)

## 4. Теория ограничений

У любой системы, например у производства, есть ограничение. Это так, потому что не существует ничего бесконечного: нет компаний, которые бесконечно продают или всегда получают только прибыль.

Это ограничение и будет слабым звеном цепи, которое снижает возможности всей системы. В каждом периоде в одной цепи может быть только одно слабое место.

Допустим, компания решает усилить производство и кидает все силы на покупку оборудования, а слабое звено — персонал. Скорее всего, компания не только не усилит систему, но и ослабит ее, например поставив плохо обученный персонал работать с новым, более сложным оборудованием.



# **Три вопроса, которые помогут найти ограничение**

В основе теории ограничений лежат мыслительные процессы. Эта теория не дает прямых ответов, а помогает думать. Так, чтобы разобраться со слабым местом в системе, нужно ответить на три вопроса:

**Что мы должны изменить?** Здесь нужно оценить ситуацию и подумать, в чём на самом деле возникла проблема. Например, если компания получает много возврата, проблема может быть не в самой продукции, а в долгих сроках доставки. Можно пообщаться с клиентами и выяснить, что их не устраивает.

**На что нужно поменять?** На этом вопросе нужно подумать, как решить проблему. Нельзя просто взять и изъять слабое звено, например вывести из продаж линейку товаров: его нужно чем-то заменить.

**Как обеспечить изменения?** Здесь нужно понять, какими будут дальнейшие шаги. Допустим, компания узнала, что выпускает большой процент брака. Она понимает, что просто перестать выпускать этот товар нельзя, поэтому решает изменить техкарту и отладить оборудование заново.

# И пять фокусирующих шагов

А еще в теории ограничений есть пять шагов, которые помогают сфокусироваться на слабом звене. Эти шаги в совокупности составляют непрерывную систему улучшений:

- Шаг 1. Выявить ограничение системы.
- Шаг 2. Решить, как эффективно использовать это ограничение.
- Шаг 3. Изменить всю систему в соответствии с принятым решением.
- Шаг 4. Улучшить состояние ограничения.
- Шаг 5. Вернуться к первому шагу и всё повторить.

# Инструменты теории ограничений: барабан, буфер, канат

Вопросы и шаги, которые мы разбирали выше, связаны с основным методом теории ограничений — его называют «барабан-буфер-канат», он же Drum-Buffer-Rope, DBR. Суть метода — не нагружать всю систему на сто процентов, пока не будет ликвидировано ограничение, потому что смысла в этом нет ↓

- Если отдел продаж не справляется, нет смысла нагружать производство: скорее всего, бизнес просто забьет склады и заморозит деньги в запасах.
- Если производство не справляется, нет смысла продавать товар в прежнем объеме: компания просто не сможет его отгрузить.
- Если маркетинг не справляется и не поставляет лидов, нет смысла нагружать продажи и производство: продавать будет некому.

- **Барабан — само ограничение.** Пока оно есть, вся система подстраивается под него.
- **Буфер — запас времени.** Ограничение должно всегда работать, чтобы не тормозить процесс еще больше.
- **Канат — связь с системой.** Нужна, чтобы ограничение получало именно столько работы, сколько успевает выполнить.

Например, в компании, которая выпускает косметику для бани, ограничение — это само производство. Оно не успевает выпускать нужный объем товара.

Пока руководство разбирается с этим, остальные отделы перестраивают работу, чтобы не нагружать производство еще больше и не создавать очередь. При этом само производство работает в полную силу, чтобы компания продолжала поставлять косметику в магазины.

## 4. Теория ограничений

Теория ограничений помогает компаниям найти узкие места, из-за которых бизнес стопорится. Пока эти узкие места остаются, как бы компания ни пыталась развиваться и сколько бы денег ни потратила, именно они будут определять результат в целом.

С помощью теории ограничений руководство компании может сфокусироваться на действительно важных моментах и направить силы и деньги на них. В итоге это поможет, к примеру, снизить процент брака, поднять продажи, расширить бизнес.

