

Международная Ассоциация Бизнес-ТРИЗ (ИВТА)

Андрей Г. Курьян
Специалист ТРИЗ, 4 уровень

Расширенный системный оператор для продукта

Квалификационная работа на соискание звания Мастер Бизнес ТРИЗ (ИВТА)

Рецензенты:
Алексей Фоменко, Специалист ТРИЗ, 4 уровень
Антон Кожемяко, Мастер Бизнес ТРИЗ

Благодарности:

В период сотрудничества с командой Георгия Марциновского в 2016 - 2024 годах идеи микро-стадий жизненного цикла продукта прошли проверку на практике. Автор выражает благодарность Георгию Марциновскому и его коллегам за сотрудничество.

Автор выражает благодарность Мастеру Бизнес ТРИЗ Антону Кожемяко за ценные замечания и предложения в ходе рецензирования данной работы.

Автор выражает благодарность специалисту 4-го уровня ТРИЗ Алексею Фоменко за глубокое погружение в содержание данной квалификационной работы и большое количество ценных замечаний предложений по ее улучшению.

Дисклеймер. При создании текста квалификационной работы использовался ChatGPT 5.2.

Оглавление

| | |
|--|-----------|
| Оглавление..... | 2 |
| Введение..... | 5 |
| Проблемы классического системного оператора ТРИЗ..... | 5 |
| Научная новизна РСО для ТРИЗ и Бизнес ТРИЗ..... | 7 |
| Методологическая строгость исследования..... | 8 |
| Практическая ценность и тиражируемость методики..... | 9 |
| Глава 1. Обзор существующих работ..... | 11 |
| 1.1. Понятие «система» в классической ТРИЗ..... | 11 |
| 1.1.1. Заимствование из системного подхода..... | 12 |
| 1.1.2. Процесс как система..... | 13 |
| 1.2. Понятие “продукт”..... | 14 |
| 1.2.1. Современные трактовки продукта..... | 14 |
| 1.2.2. Продукт как объект потребления vs объект управления..... | 15 |
| 1.3. Понятие “жизненный цикл системы” (“ЖЦ системы”)..... | 17 |
| 1.3.1. Понятие “ЖЦ системы” в системной инженерии..... | 17 |
| 1.4. Понятие “жизненный цикл продукта”..... | 20 |
| 1.4.1. Маркетинговое понимание ЖЦ продукта..... | 20 |
| 1.4.2. Инженерное понимание жизненного цикла продукта..... | 22 |
| 1.4.3. Сравнение маркетингового и инженерного подходов..... | 23 |
| 1.4.4. Примеры жизненного цикла продукта..... | 25 |
| 1.5. Системный оператор в классической ТРИЗ..... | 25 |
| 1.5.1. Системные переходы..... | 26 |
| 1.5.2. Примеры применения..... | 27 |
| 1.6. Онтогенез и филогенез в системном операторе..... | 28 |
| 1.6.1. Ось жизненного цикла системы в системном операторе..... | 28 |
| 1.6.2. Онтология системного оператора и вариативность экранов..... | 29 |
| 1.7. Инструменты ТРИЗ, использующие ЖЦ системы..... | 29 |
| 1.7.1. Метод анализа главных параметров ценности (MPV)..... | 29 |
| 1.7.2. Обратный функциональный анализ..... | 33 |
| 1.7.3. Анализ ключевых параметров успешности..... | 36 |
| 1.8. Заключение..... | 38 |
| Глава 2. Расширенный СО для продукта..... | 40 |
| 2.1. Модель минимального продукта..... | 40 |
| 2.1.1. Минимальная система и продукт..... | 40 |
| 2.1.2. Определение минимального продукта..... | 41 |
| 2.1.3. Концепция Jobs to Be Done..... | 41 |
| 2.1.4. Представление продукта в системном операторе..... | 42 |
| 2.1.5. Вывод по разделу 2.1..... | 45 |
| 2.2. Модель продукта (не минимального)..... | 45 |

| | |
|--|-----------|
| 2.2.1. Множественность сценариев использования..... | 45 |
| 2.2.2. Микро-стадия использования..... | 46 |
| 2.2.3. Потребители и их роли на микро-стадиях..... | 50 |
| 2.3. Модель жизненного цикла продукта..... | 52 |
| 2.3.1. Понятие сценария взаимодействия..... | 52 |
| 2.3.2. Микро-стадии на разных стадиях ЖЦ продукта..... | 53 |
| 2.3.3. Перенос микро-стадий между разными стадиями ЖЦ..... | 55 |
| 2.3.4. Ось жизненного цикла в системном операторе продукта..... | 57 |
| 2.3.5. Стейкхолдеры на разных этапах жизненного цикла продукта..... | 62 |
| 2.4. Связь модели ЖЦ продукта с моделью бизнес-системы..... | 65 |
| 2.4.1. Модель бизнес-системы и ЖЦ продукта..... | 66 |
| 2.4.2. Значение этапов ЖЦ продукта для компаний..... | 68 |
| 2.4.3. Учет задач продукта при развитии Бизнес ТРИЗ..... | 71 |
| 2.4.4. Определение продукта в РСО..... | 73 |
| 2.5. Модель подсистемы..... | 73 |
| 2.5.1. Подходы к декомпозиции системы на подсистемы..... | 73 |
| 2.5.2. Декомпозиция в расширенном системном операторе..... | 74 |
| 2.5.3. Вариативная декомпозиция системы..... | 76 |
| 2.5.4. Примеры: декомпозиция продукта на разных стадиях ЖЦ..... | 76 |
| 2.6. Модель эволюции продукта..... | 79 |
| 2.6.1. Ось эволюции в классическом системном операторе ТРИЗ..... | 79 |
| 2.6.2. Роль оси жизненного цикла продукта..... | 80 |
| 2.6.3. Изменение микро-стадий при эволюции продукта..... | 81 |
| 2.6.4. Примеры эволюции с учетом оси эволюции и оси ЖЦ..... | 82 |
| 2.6.5. Выводы о полезности совмещения двух осей..... | 85 |
| 2.7. Заключение к главе 2..... | 86 |
| Глава 3. Методика использования..... | 88 |
| 3.1. Методика использования СО в классической ТРИЗ..... | 88 |
| 3.2. Отличия методики от классической..... | 89 |
| 3.3. Этап сбора исходных данных..... | 93 |
| 3.4. Этап построения моделей..... | 99 |
| 3.5. Особенности применения расширенного системного оператора..... | 104 |
| 3.5.1. Поиск проблем и противоречий в существующем продукте..... | 105 |
| 3.5.3. Анализ стейкхолдеров и ролей..... | 109 |
| 3.5.4. Анализ доступных ресурсов..... | 113 |
| 3.6. Область применения и ограничения расширенного системного оператора продукта..... | 118 |
| 3.6.1. Предпосылки и условия эффективности применения РСО..... | 118 |
| 3.6.2. Типы систем для работы с РСО..... | 119 |
| 3.6.3. Типы систем, для которых применение РСО ограничено..... | 119 |
| 3.6.4. Классы задач, решаемых с помощью РСО..... | 120 |
| 3.6.5. Практические ограничения использования..... | 121 |

| | |
|--|------------|
| 3.7. Выводы по Главе 3..... | 122 |
| Заключение..... | 123 |
| Основные результаты исследования..... | 123 |
| Значение результатов для развития Бизнес ТРИЗ..... | 125 |
| Направления для будущих исследований..... | 126 |
| Литература..... | 129 |
| Приложение 1. Сравнительный анализ понятия “система” в СИ и ТРИЗ..... | 133 |
| Приложение 2. Сравнительный анализ понятий “система” и “процесс”..... | 138 |
| Приложение 3. Сравнительный анализ понятий “система” и “продукт”..... | 140 |
| Приложение 4. Методика PCO для продукта..... | 147 |
| Приложение 5. Пример..... | 149 |
| 1. Постановка задачи и границы целевой системы («Система / Настоящее»)..... | 149 |
| 2. Надсистема и контекст («Надсистема / Настоящее»)..... | 149 |
| 3. Подсистемы («Подсистема / Настоящее»)..... | 150 |
| 4. Временная ось + жизненный цикл как дополнительная ось PCO..... | 151 |
| 4.1. Прошрое (до подключения)..... | 151 |
| 4.2. Настоящее (подключение и ввод в эксплуатацию)..... | 151 |
| 4.3. Будущее (эксплуатация и масштабирование)..... | 151 |
| 5. Разбор этапов подключения через PCO (уровни + жизненный цикл)..... | 152 |
| Этап 1. Подключение агента (маркетплейс → HubSpot)..... | 152 |
| Этап 2. Тестирование на существующих лидах..... | 152 |
| Этап 3. Настройка: связи свойств HubSpot ↔ входы/выходы агента..... | 153 |
| Этап 4. Настройка оплаты агента..... | 155 |
| 6. Результат применения PCO: требования, риски, артефакты..... | 155 |
| 7. Вывод по примеру..... | 156 |

Введение

Проблемы классического системного оператора ТРИЗ

Классический системный оператор ТРИЗ (так называемая *многоэкранная* схема Г.С. Альтшуллера и инструментарий работы с ней, далее “многоэкранка”) рассматривает целевую систему во взаимосвязи с ее подсистемами и надсистемой по шкале эволюции (прошрое, настоящее, будущее). Этот инструмент развивает системное мышление изобретателя, позволяя выйти за рамки одной системы и учесть более широкий контекст во времени и пространстве. Однако практика применения классического системного оператора выявила ряд фундаментальных вопросов, на которые он не даёт чётких ответов. В частности, остаются неясными следующие моменты:

- **Сколько надсистем может быть определено для системы?** В классическом подходе обычно указывается одна надсистема (ближайшее внешнее окружение), но на деле система может одновременно входить в несколько более крупных систем. Не определено, следует ли ограничиваться одной надсистемой или учитывать сразу несколько контекстов окружающей среды системы.
- **Каковы критерии правильного выбора надсистемы для анализа?** Поскольку любая система может являться элементом многих более обширных систем, выбор надсистемы повлияет на результаты анализа. Методика не предлагает формализованных критериев, как определить “правильную” надсистему - ближайшую или более далёкую. На практике аналитик опирается на интуицию и цели исследования, что вносит субъективность.
- **Как правильно декомпозировать систему на подсистемы и сколько вариантов декомпозиции рассматривать?** Классический системный оператор фиксирует одну иерархию подсистем (обычно - основные компоненты системы). Но сложный продукт можно разбить на подсистемы разными способами (функционально, по компонентам, по технологическим модулям и т.д.). ТРИЗ не регламентирует, какая декомпозиция “правильная” и нужно ли исследовать несколько вариантов. Это затрудняет полноту анализа, ведь разные разбиения могут выявлять разные проблемы и ресурсы.
- **Как корректно определять систему “в прошлом” и “в будущем”?** Предлагается анализировать предысторию системы (прототипы, предыдущие поколения) и прогнозировать ее будущее (эволюцию системы). Однако не ясно, какие именно прошлые состояния выбирать

(самый первый прототип? предыдущую версию?) и как обоснованно описать будущую форму системы. Без чётких критериев выбор прошлой и будущей версии системы остается субъективным, что снижает достоверность анализа и прогноза.

- **Сколько стадий жизненного цикла учитывать у системы?**
Классическая 9-экранная модель явно не описывает стадий жизненного цикла - обычно ограничиваются общим рассмотрением “прошлого-настоящего-будущего”. Не определено, нужно ли детализировать жизненный цикл на отдельные этапы (разработка, производство, эксплуатация, утилизация и т.п.) при анализе, особенно для современных продуктов с быстрыми итерациями.
- **Сколько стейкхолдеров (заинтересованных сторон) имеет система?** В традиционном подходе часто не акцентируется внимание на разнообразии стейкхолдеров. Фокус обычно на пользователе и разработчике, тогда как реальный продукт взаимодействует с множеством групп: потребители разных типов, заказчики, производственные подразделения, сервисные службы, регуляторы и др. Классический оператор не дает указаний, как выявлять и учитывать всех значимых стейкхолдеров системы.

Перечисленные вопросы остаются без полного ответа в рамках классической ТРИЗ. Существующие материалы либо не упоминают эти аспекты, либо дают лишь общие рекомендации без строгой методик.

Кроме того, **классический системный оператор имеет существенные ограничения при анализе современных продуктов:** в классическом системном операторе в настоящем обычно рассматривается только одна фиксированная структура системы (подсистемы) и одна надсистема. При этом не определено:

- Как система, лежащая в основе продукта, решает проблемы пользователей.
- Как должно учитываться разнообразие стейкхолдеров продукта и их требований на разных стадиях его жизненного цикла, в том числе, на ранних стадиях зарождения продукта (problem - solution fit), и последующих стадиях жизненного цикла продукта, когда адаптация продукта к рынку (product - market fit) и последующее тиражирование продукта (scale).

Перечисленные проблемы свидетельствуют о необходимости развития модели системного оператора.

Научная новизна РСО для ТРИЗ и Бизнес ТРИЗ

Модель расширенного системного оператора (РСО) для продукта, предлагаемая в данной работе, нацелена на преодоление перечисленных выше недостатков классического системного оператора. Новый подход вводит в модель системного оператора дополнительные измерения анализа системы, позволяющие учесть *полный жизненный цикл продукта*, различные условия и сценарии его использования, а также множественность представлений структуры системы. Вместо модели плоской многоэкранки в классическом системном операторе вводится 3-х мерная модель (по сути, “системный оператор в кубе”), охватывающая разные контексты и этапы жизненного цикла и развития системы, лежащей в основе продукта.

Научная новизна данной работы заключается в развитии методологии как классического ТРИЗ, так и Бизнес ТРИЗ посредством новой модели системного оператора и методики ее применения совместно с различными инструментами ТРИЗ и Бизнес ТРИЗ.

В работе предложен оригинальный подход к моделированию и анализу *продукта* на основе расширенного системного оператора ТРИЗ и его совместного использования с известными инструментами ТРИЗ и Бизнес ТРИЗ.

Введены новые понятия, дополняющие понятийный аппарат ТРИЗ и Бизнес ТРИЗ:

- «**минимальный продукт**»,
- «**микро-стадия жизненного цикла системы**» и
- «**сценарий взаимодействия стейкхолдера с продуктом**».

Они позволяют детально описывать концепцию продукта, учитывая минимально достаточную конфигурацию для решения задачи пользователя, более мелкие фазы внутри основных стадий развития, и различные роли пользователей при взаимодействии с продуктом. Также обоснована трёхмерная структура системного оператора продукта - по трём осям: (1) уровни системы (подсистема-система-надсистема), (2) эволюция во времени (прошлое-настоящее-будущее), (3) **стадии жизненного цикла продукта**. Показано, что такая модель применима для анализа продуктов любого типа.

В части жизненного цикла предложенная модель предлагает универсальный подход к описанию и анализу стадий жизненного цикла систем в системной инженерии и ТРИЗ, и стадий жизненного цикла продукта в маркетинге и

менеджменте продукта. Тем самым расширяется область применения классической ТРИЗ и Бизнес ТРИЗ: **3-мерный оператор** охватывает не только техническую эволюцию системы, но и контекст ее рыночного применения, пользовательский опыт и всю цепочку создания ценности в продукте.

Отдельно в работе показана связь между моделью жизненного цикла продукта и моделью бизнес-системы. Проанализирована существующая модель бизнес-системы (модель В. Сушкова в Бизнес-ТРИЗ) и выявлены ее ограничения: модель бизнес-системы фокусируется на стадиях адаптации продукта к рынку (*product-market fit*) и последующего масштабирования (*scale*) продукта, когда ценность продукта уже доказана; она слабо отражает самые ранние этапы жизненного цикла продукта - этапы зарождения идеи и проверки концепции (*problem-solution fit*). В квалификационной работе подчеркивается необходимость разработки специальной модели бизнес-системы для ранних стадий жизненного цикла продукта.

Методологическая строгость исследования

Методологическая основа исследования опирается на **подход**, используемый в *design science*: в работе последовательно выполнены анализ существующих моделей и практик, синтез расширения модели и разработка воспроизводимой методики её применения. В качестве базового инструмента принят **классический системный оператор ТРИЗ** (многоэкранная схема Г. С. Альтшуллера), который в рамках исследования расширен за счёт введения дополнительного измерения, отражающего **жизненный цикл продукта**, а также за счет включения сценарного разнообразия и многообразия стейкхолдеров.

Для проведения исследования и построения моделей использована следующая методика:

1. **Обзор и сравнительный анализ источников и моделей.** Выполнено сопоставление трактовок ключевых понятий («система», «продукт», «жизненный цикл системы/продукта») в смежных дисциплинах (ТРИЗ, системная инженерия, продуктовый менеджмент/маркетинг) и анализ существующих расширений многоэкранных моделей. Это позволило выявить несоответствия терминологии, ограничения применяемых подходов и области, где классический системный оператор оказывается недостаточным.
2. **Онтологически ориентированное моделирование.** Для обеспечения однозначности и проверяемости построений выполнена формализация элементов РСО: зафиксированы понятия, их роли, атрибуты и взаимосвязи,

а также правила переходов между «экранами» модели. Онтологическое оформление использовано как средство устранения терминологической неоднозначности и как способ обеспечить целостность модели при добавлении новых измерений.

3. **Построение модели как алгоритмизируемой процедуры.** Разработан и описан пошаговый процесс построения расширенной модели продукта: от определения границ целевой системы и выбора надсистем/подсистем до выделения стадий и микро-стадий жизненного цикла, описания сценариев взаимодействия и привязки стейкхолдеров и значимых подсистем к соответствующим стадиям. Для каждого шага заданы критерии и правила, что обеспечивает воспроизводимость результата и прозрачность оснований принимаемых решений.
4. **Верификация применимости через практическую апробацию.** Корректность и полезность предложенной модели и методики подтверждаются многократным использованием подхода в профессиональной практике и обучении инженеров и студентов, что позволяет рассматривать методику как отчуждаемую и воспроизводимую, а не как набор эвристик отдельного эксперта.

Тем самым методологическая строгость работы обеспечивается (а) явным определением понятий и структур, (б) онтологической согласованностью модели, (в) алгоритмизацией процедуры построения, (г) проверяемостью и документируемостью шагов анализа, а также (д) междисциплинарной опорой на ТРИЗ, системную инженерию и продуктовый менеджмент как взаимодополняющие источники требований к модели продукта.

Практическая ценность и тиражируемость методики

Практическая ценность работы определяется тем, что предложенный расширенный системный оператор продукта (PCO) и методика его применения ориентированы не на абстрактное описание, а на **решение типовых задач управления и развития продукта**: уточнение границ системы и контекста эксплуатации, согласование требований и критериев качества, выявление конфликтов и точек риска, генерация и сравнение вариантов изменений, а также формирование обоснованной логики продуктовой эволюции по стадиям жизненного цикла. Прикладной результат выражается в возможности получать **структурированные, проверяемые и пригодные для коммуникации модели продукта**, которые поддерживают принятие решений в межфункциональных командах и снижают потери на несогласованность терминов, неполноту рассмотрения и повторное «переоткрытие» уже известных ограничений.

Тиражируемость методики обеспечивается ее оформлением как **воспроизводимой процедуры** построения модели: фиксируются входные данные (описание продукта/контекста, стейкхолдеры, ограничения, наблюдаемые проблемы), шаги моделирования (переходы между уровнями системы и временными срезами, привязка стадий жизненного цикла, выделение сценариев и критичных взаимодействий), а также правила фиксации результатов (единый словарь понятий, требования к полноте описания, логика связывания решений с экранами модели). За счёт этого методика может быть использована не только автором, но и другими специалистами (аналитиками, инженерами, продакт-менеджерами) при наличии исходной информации о продукте и соблюдении указанного регламента работы с моделью.

Практическая применимость подхода подтверждается его использованием в профессиональной и образовательной практике, что демонстрирует возможность переноса методики на разные предметные области при сохранении ее ключевых шагов и принципов.

Глава 1. Обзор существующих работ

В данной главе представлен обзор и критика понятий, моделей и методик, имеющих отношение к системному оператору. Для анализа были взяты модели и методики из следующих дисциплин знаний:

- Системная инженерия (System Engineering);
- Инженерия программных систем (Software Engineering);
- Маркетинг и менеджмент продуктов (Product Marketing & Management);
- Классическая ТРИЗ;
- Бизнес ТРИЗ.

1.1. Понятие «система» в классической ТРИЗ

Первая публикация Г.С. Альтшуллера и Р.Б. Шапиро обозначила системный взгляд на изобретение. Авторы рассматривали технический объект (машину или процесс) как систему взаимосвязанных частей, развитие которой тормозится возникновением внутренних противоречий. В частности, было показано, что «устранение противоречия приводит к изобретению». Уже в этой работе явно применён системный подход: поиск решения предлагалось вести не только в пределах самой системы, но и во внешней среде и смежных системах. Таким образом, возникло понимание, что изобретательская задача должна рассматриваться во всеобъемлющем системном контексте [1].

В начале 1960-х Альтшуллер развивал методику решения изобретательских задач, опираясь на систематизацию опыта изобретателей. Первая книга по ТРИЗ - «Как научиться изобретать» - представила рациональную систему шагов решения вместо бессистемного поиска. Хотя термин «техническая система» ещё не был строго определён, подход подразумевал целостный анализ устройства и его функций в рамках общей системы. Уже намечалось, что каждый изобретательский объект выполняет главную функцию и состоит из взаимодействующих компонентов, улучшение которого требует анализа его структуры и связей [2]. Эта методика заложила основу для дальнейшего развития системной терминологии в ТРИЗ.

К концу 1960-х понятие системы прочно вошло в арсенал ТРИЗ. В книге «Алгоритм изобретения» Альтшуллер представил уже отработанную методику (АРИЗ) решения задач, где одной из ключевых составляющих был анализ технической системы и ее элементов [3]. Изобретательская задача к этому времени понималась как противоречие в конкретной технической системе, требующее разрешения. Были сформулированы первые законы развития технических систем и введено представление об идеальной системе, к которой

должна стремиться эволюция рассматриваемой системы. Таким образом, на рубеже 1960-х ТРИЗ начала рассматривать эволюцию технических систем как закономерный процесс, поддающийся прогнозированию.

В середине 1970-х система понятий ТРИЗ достигла зрелости. В версии АРИЗ-75 были введены новые ключевые концепции: **физическое противоречие** (противоречие требований к одному элементу системы) и модель **вещественно-полевого анализа** для представления минимальной технической системы. Альтшуллер к этому времени подчеркивал иерархический характер техники - любую техническую систему следовало рассматривать в контексте её надсистемы и подсистем. Появились первые стандарты решения изобретательских задач для типовых преобразований систем. Понимание «**технической системы**» расширилось: это уже не просто совокупность частей, а динамичный объект, развивающийся по законам повышения идеальности. Вводится понятие **идеальной системы**: идеальная система - это такая, «**которой нет, а её функция выполняется**». Таким образом, к 1975 году в ТРИЗ оформился системный подход: изобретение рассматривалось как этап развития технической системы через разрешение противоречий вплоть до идеального состояния [4].

1.1.1. Заимствование из системного подхода

Важно отметить, что понятие «система» и сам системный подход в ТРИЗ не возникли в вакууме. Альтшуллер заимствовал и творчески переработал идеи, сформировавшиеся в смежных областях науки 1950-1960-х годов, особенно в общей теории систем и системной инженерии. В этот период в мировой практике складывалась методология проектирования сложных объектов как целостных систем - примером может служить перевод на русский язык фундаментальной работы Г. Гуда и Р. Макола «Системотехника. Введение в проектирование больших систем» (1962) [5]. Подобные труды заложили принципы представления технического объекта как комплекса взаимосвязанных компонентов, объединенных общей целью. ТРИЗ впитала эти междисциплинарные идеи: уже первая статья Альтшуллера предлагала рассматривать изобретательскую задачу в широком системном контексте, а не в отрыве от окружения. Впоследствии системные понятия (надсистема, подсистема, среда) были интегрированы в алгоритмы ТРИЗ для анализа и решения изобретательских задач. Таким образом, системный подход, изначально разработанный в инженерных науках для упорядочивания разработки сложных технических комплексов, был адаптирован в ТРИЗ для целей изобретательского анализа и прогнозирования развития техники.

В приложении (см. Приложение 1) представлен сравнительный анализ понятия система в системной инженерии и ТРИЗ. Он показывает, что хотя

исторически эти понятия происходят из общих работ по системному подходу 50-х годов XX столетия, дальнейшая эволюция понятий в СИ и ТРИЗ происходила по разным траекториям. Расхождения не являются критичными. Более того, подходы взаимодополняют друг друга: в СИ упор сделан на полном жизненном цикле системы с упором на стадии создания и разработки системы; ТРИЗ привносит в систему эволюционное измерение системы из прошлого через настоящее в будущее и фокусируется на выявлении и решении проблем в системе. На основании анализа можно сделать вывод, что интеграция понятия “система” на основании обоих подходов (системная инженерия и ТРИЗ) улучшает понимание систем, позволит обогатить оба подхода и позволит облегчить совместное применение обоих подходов при создании и совершенствовании систем на практике.

1.1.2. Процесс как система

Процесс в терминологии системной инженерии определяется как «набор взаимосвязанных или взаимодействующих действий, преобразующих входы в выходы» [6]. По сути, процесс - это **последовательность операций**, выполняемых во времени для получения определённого результата или продукта. Иными словами, когда мы наблюдаем последовательность шагов, приводящую к выходному результату, мы имеем дело с процессом; процесс преобразует входные ресурсы в выходы целенаправленной работой на каждом этапе [7].

Ключевое отличие между системой и процессом заключается в характере их организации во **времени и пространстве**. Система имеет преимущественно **пространственную структуру**: её элементы существуют одновременно и взаимодействуют в рамках некоторой конфигурации (структуры) в данный момент времени. Напротив, процесс обладает **временной структурой**: его составляющие - операции или действия - разворачиваются последовательно во времени, образуя упорядоченный поток работ [8]. Еще одна трактовка: процесс - это динамическая система, существующая в пространстве-времени, тогда как статическая система - это проекция такой динамической системы на пространство текущего момента [7].

В приложении (см. Приложение 2) представлен сравнительный анализ понятий “система” и “процесс”.

Таким образом, **процесс можно рассматривать как разновидность системы**, обладающей *временной* организацией. Фактически, процесс представляет собой систему действий, связанная структура которых ориентирована на достижение результата. Это отражено и в стандартах: например, ISO 15288 указывает, что элементы системы могут включать *процессы* наряду с оборудованием, программным обеспечением, людьми и

другими компонентами [8]. В системной инженерии процессы нередко моделируются как часть системы - благодаря этому можно анализировать, как последовательность операций (процесс) способствует функционированию целого. В итоге, система и процесс дополняют друг друга: первая описывает объект и его структуру, второй - деятельность (поведение) во времени, направленную на достижение результатов [7].

1.2. Понятие “продукт”

В классическом маркетинге **продукт** определяется очень широко и универсально. Так, Филип Котлер и Кевин Л. Келлер указывают, что продукт - это **«всё, что может быть предложено рынку для удовлетворения какой-либо потребности или желания»**, включая не только физические товары, но и услуги, события, идеи, места и даже личности [9]. Иными словами, продукт - это не просто материальный объект, а **набор свойств и выгод**, способных удовлетворить запросы потребителей. Маркетологи подчеркивают, что потребитель покупает **не сам товар как таковой, а пользу или решение своей проблемы**, которое этот товар предлагает. Ещё в 1960 году Теодор Левитт образно заметил: *«людям не нужен четвертьдюймовый сверло, им нужна четвертьдюймовая дырка»* [10], подчеркивая, что с точки зрения потребителя ценность продукта состоит в результате его использования.

Классические авторы также рассматривали продукт как **совокупность атрибутов**. Например, Уильям Стэнтон описывал продукт как **набор материальных и нематериальных характеристик**, включая упаковку, цену, имидж производителя и сервис, которые покупатель воспринимает целостно, ожидая удовлетворения своих нужд [11]. Таким образом, **продукт как объект потребления** в традиционном понимании - это комплексное предложение, дающее определенную **пользу потребителю**. Питер Друкер отмечал, что покупатель, фактически, **приобретает не сам товар, а ценность, или полезность, которую этот товар ему приносит** [12]. Это означает, что в фокусе внимания находится удовлетворение потребностей: продукт ценен тогда, когда он решает проблему клиента или выполняет для него **«работу»**.

1.2.1. Современные трактовки продукта

Современные подходы в **маркетинге и менеджменте продукта** развивают и дополняют классическое определение. В индустрии высоких технологий и интернет-сервисов продукт всё чаще рассматривается не как отдельный физический товар, а как **целостный пользовательский опыт**. Марти Каган, один из ведущих экспертов в продуктовом менеджменте, подчеркивает, что **великий продукт - это способ решения проблем таким**

образом, чтобы одновременно удовлетворять потребности пользователя и цели бизнеса [13]. В его понимании продукт существует на пересечении ценности для клиента, технической реализуемости и бизнес-целесообразности. Аналогично, в среде цифровых продуктов под продуктом понимают комбинацию функциональности, дизайна, контента и монетизации, формирующую цельный опыт для пользователя [14]. Такой подход расширяет понятие продукта: это не только то, что предлагается, но и как это воспринимается и используется клиентом.

Ещё один современный взгляд предложил Клейтон Кристенсен в теории «работы, для которой нанимается продукт» (Jobs to Be Done). Он предлагает компаниям фокусироваться на том, **какую задачу потребитель “нанимает” продукт выполнить**, то есть какую потребность в конечном счёте закрывает предложение [15]. С этой точки зрения, успешный продукт - это то, что наиболее точно и удобно решает задачу клиента. Если товар перестаёт справляться с «работой», потребитель «увольняет» его и находит замену. Такой подход подчеркивает, что **существо продукта определяется ценностью его использования**, а не только характеристиками. Например, Кристенсен показывал, что люди «нанимают» молочный коктейль не просто как напиток, а чтобы скрасить длительную поездку и утолить голод - то есть коктейль конкурирует не с другими напитками, а с любым способом сделать поездку менее скучной и утолить голод [16]. Современные трактовки, таким образом, укрепляют идею, что продукт - это **решение для пользователя**, и его успех зависит от того, насколько хорошо он выполняет нужную работу.

1.2.2. Продукт как объект потребления vs объект управления

Важно различать **продукт как объект потребления** и **продукт как объект управления**.

Продукт как объект потребления - это взгляд со стороны клиента. Здесь акцент на том, **какую пользу, ценность и опыт** получает потребитель. Продукт выступает носителем **свойств и выгод**, ради которых его приобретают. Как отмечалось выше, для покупателя главными являются удовлетворение потребности, решение проблемы или получение определённого опыта. В маркетинге это находит отражение в концепциях уровня продукта: ядро (базовая выгода), реальный продукт (конкретные характеристики и качество) и расширенный продукт (сопутствующие сервисы, гарантия, бренд и др.) - всё это формирует ценность для потребителя [9]. Проще говоря, **с точки зрения потребления продукт - это то, что обещает и приносит удовлетворение**. Если ожидания не оправдываются, продукт не будет востребован, каким бы совершенным ни был его дизайн или технология.

Продукт как объект управления - это взгляд со стороны производителя, фирмы и особенно продакт-менеджера. Здесь продукт рассматривается как **единица бизнеса**, которую нужно планировать, развивать и поддерживать. Продукт обладает жизненным циклом - от идеи и разработки через запуск на рынок, рост, зрелость и спад - и на каждой стадии им необходимо грамотно управлять [9]. Для компании продукт - это объект инвестиций, источник прибыли и конкурентное предложение на рынке, требующий стратегических решений. Питер Друкер подчеркивает, что цель бизнеса - **создать потребителя**, а это значит, что управление продуктом должно исходить из понимания потребностей клиента и постоянного их удовлетворения лучше конкурентов [17]. Продакт-менеджмент объединяет различные функции (маркетинг, разработка, дизайн, продажи) вокруг продукта, чтобы обеспечить его успех. Марти Каган отмечает, что продуктовая команда должна одновременно решать три задачи: **полезность для пользователя, техническая осуществимость и бизнес-ценность** продукта [13]. Иначе говоря, **продукт как объект управления** - это фокус всех внутренних процессов компании (стратегии, разработки, маркетинга) на том, чтобы создать и поддерживать такое предложение, которое будет конкурентоспособным и востребованным.

В результате, хотя **продукт один и тот же**, угол зрения разнится. **Как объект потребления** - это носитель ценности для клиента, набор свойств и выгод, формирующих удовлетворённость. **Как объект управления** - это комплекс решений и действий компании по созданию, улучшению и доставке этой ценности. Успешный продакт-менеджмент соединяет оба аспекта: глубокое понимание того, что ценят потребители, с эффективным управлением продуктом внутри организации. Именно такой целостный подход обеспечивает, что продукт одновременно желанен на рынке и управляем с точки зрения бизнеса.

В приложении (см. Приложение 3) представлен сравнительный анализ понятий “система” и “продукт”.

Система и продукт представляют собой целостные комплексы элементов, создаваемые для удовлетворения определенных потребностей и обладающие жизненным циклом развития. Общее между ними заключается в целенаправленности создания, наличии структуры, управляемой эволюции и использовании системных методов разработки и анализа.

Различие состоит в том, что система рассматривается преимущественно как технический объект с внутренней структурой и акцентом на функциональность и устранение противоречий, тогда как продукт рассматривается как рыночное предложение ценности, ориентированное на удовлетворение потребителя и коммерческий успех. Иными словами, система

фокусируется на технической состоятельности решения, а продукт - на его ценности для пользователя и конкурентоспособности на рынке.

1.3. Понятие “жизненный цикл системы” (“ЖЦ системы”)

Конкретное историческое введение термина *жизненный цикл* в ТРИЗ не фиксируется ранними первоисточниками Альтшуллера — он появился позднее, когда в ТРИЗ были интегрированы концепции системной инженерии в части жизненного цикла системы.

Идеи, отражающие изменение системы во времени, появились в ТРИЗ через законы и тренды развития технических систем, которые формировались в 1970-х годах в работах Генриха Альтшуллера. Однако, изменение системы рассматривалось с эволюционной точки зрения, но не с точки зрения жизненного цикла системы.

Конкретное понятие “*жизненный цикл системы*” как термин входит в современные ТРИЗ-гlossарии и стандарты ТРИЗ [18, 19].

1.3.1. Понятие “ЖЦ системы” в системной инженерии

Жизненный цикл системы - это совокупность последовательных стадий (фаз) развития системы во времени, через которые проходит система от момента возникновения замысла или потребности в ней до её окончательного вывода из эксплуатации. Иначе говоря, жизненный цикл описывает путь системы начиная с концепции и определения требований, через её разработку и внедрение, последующую эксплуатацию с поддержкой, и вплоть до прекращения использования и утилизации. Важно отметить, что жизненный цикл рассматривается не просто как временной отрезок существования системы, но как **процесс смены состояний системы**, обусловленный выполнением определенных работ на каждой стадии и влиянием внутренних/внешних факторов на систему во время ее жизненного цикла.

В стандарте ISO/IEC/IEEE 15288 [20] и в руководстве INCOSE выделяется ряд **ключевых фаз жизненного цикла** системы, охватывающих весь процесс её создания и использования. Обычно к таким фазам относят:

- **Концепция (Concept)** - формирование замысла системы и обоснование ее необходимости. На этом этапе выполняется анализ потребностей и целей, которые должна удовлетворить будущая система, собираются и уточняются начальные требования заинтересованных сторон. Результатом стадии концепции является общая **концептуальная модель системы** или **техническое задание**, где определено, *что* необходимо создать и *для чего* (целевое назначение).

- **Разработка (Development)** - этап детального проектирования и разработки системы. В ходе разработки система **специфицируется и конструируется**: разрабатывается архитектура и дизайн, создаются необходимые компоненты (аппаратные, программные и др.), проводятся интеграция компонентов и первоначальная проверка решений. К концу стадии разработки получают спроектированную систему, готовую к изготовлению и внедрению.

- **Производство (Production)** - этап реализации системы в физическом или полном программном воплощении. Сюда входит **изготовление, сборка** всех элементов системы согласно проектной документации, а также **интеграция и испытания** системы. На этой фазе система доводится до состояния, пригодного для передачи в эксплуатацию: выполняются сборочные работы, заводские испытания, проверка соответствия системы требованиям.

- **Внедрение и эксплуатация (Utilization)** - этап ввода системы в действие и ее функционирования в целевой среде. На этой фазе система **внедряется** в рабочую (операционную) среду заказчика и **начинает использоваться по назначению**. Проводятся работы по установке/развертыванию системы, обучению персонала, после чего система эксплуатируется пользователями для достижения поставленных целей. В период эксплуатации осуществляется мониторинг работы системы в реальных условиях, накопление опыта ее использования, оценка эффективности и удовлетворенности пользователей.

- **Поддержка (Support)** - сопровождение системы в ходе её эксплуатации. Эта фаза включает в себя **техническое обслуживание, поддержку, модернизацию** и обновление системы для обеспечения требуемого уровня ее работоспособности и эффективности. Выполняются регламентные обслуживания, ремонт, устранение возникающих неполадок, выпуск новых версий программного обеспечения или модификаций, а также логистическая поддержка (обеспечение запасными частями, ресурсами и т.п.) для продления срока службы системы.

- **Выведение из эксплуатации (Retirement/Disposal)** - заключительная фаза жизненного цикла, когда происходит прекращение использования системы. Система **выводится из эксплуатации** по завершении её жизненной программы или при утрате актуальности. На этой стадии осуществляются работы по отключению или демонтажу системы, её утилизации или сохранению (консервации) отдельных компонентов, а также анализ накопленного опыта эксплуатации. Завершающие мероприятия включают безопасное удаление или переработку отходов, архивирование данных и документации, подведение итогов проекта.

Следует подчеркнуть, что набор и названия фаз могут незначительно различаться в разных организациях или проектах. Тем не менее, описанные этапы представляют общий структурированный подход, рекомендованный

стандартом ISO/IEC/IEEE 15288:2015 [20] для полного охвата процесса создания и эксплуатации систем.

Использование жизненного цикла для управления разработкой, внедрением и эксплуатацией систем заключается в том, что каждая фаза сопровождается соответствующими процессами и контрольными мероприятиями, позволяющими направленно и прозрачно управлять прогрессом проекта. В частности, в системной инженерии принято вводить **контрольные точки принятия решений** (*decision gates*) на границах фаз жизненного цикла. На каждой такой контрольной точке проводится всесторонняя оценка результатов текущей стадии и проверяется готовность системы к переходу на следующую стадию. Например, по завершении стадии разработки должен быть проведен анализ проектных решений и тестирование прототипов, чтобы убедиться, что система удовлетворяет исходным требованиям, прежде чем передать ее в производство; аналогично, перед началом эксплуатации оценивается полнота испытаний и готовность инфраструктуры для внедрения системы. Такой подход с «воротами решений» обеспечивает, что продвижение по жизненному циклу происходит только при достижении целей текущей фазы и приемлемом уровне риска. Решения о переходе (или непереходе) на следующий этап принимаются ответственными лицами на основе критериев входа/выхода, заранее определенных для каждой фазы, и документируются в рамках проектного управления. Это позволяет **снизить риски** и избежать преждевременного шага вперед, если система ещё не готова или не соответствует необходимым критериям качества.

Таким образом, концепция жизненного цикла системы служит фундаментом для **планирования и управления разработкой сложных систем**. Разбиение проекта на этапы с определёнными целями и результатами способствует координации действий множества специалистов и подсистем. Стандарт ISO/IEC/IEEE 15288 [20] прямо опирается на модель жизненного цикла, определяя набор процессов инженерии системы, выполняемых на разных стадиях ее жизни. Руководство INCOSE также подчёркивает, что использование этапов жизненного цикла и связанных с ними процессов обеспечивает целостный и управляемый подход к созданию систем [7]. В результате, применение концепции жизненного цикла на практике помогает последовательно провести систему через все стадии - от первоначальной идеи, через разработку и внедрение, к длительной эксплуатации - и успешно завершить её жизнь организованным выводом из эксплуатации, достигнув поставленных целей проекта с контролем качества на каждом шаге.

1.4. Понятие “жизненный цикл продукта”

Понятие **жизненного цикла продукта (ЖЦ продукта)** является фундаментальным как в маркетинге, так и в инженерии, при этом интерпретация данного понятия в этих областях различается. В маркетинговом менеджменте **жизненный цикл** описывает рыночную судьбу продукта – от вывода на рынок до снятия с продажи – и связывается с динамикой спроса, продаж и прибыли на различных этапах [9, 21].

В инженерно-техническом контексте **жизненный цикл** охватывает техническое существование продукта – от замысла и разработки через производство и эксплуатацию до вывода из эксплуатации и утилизации [8].

1.4.1. Маркетинговое понимание ЖЦ продукта

В маркетинге **ЖЦ продукта** представляет собой модель, описывающую изменения объема продаж и прибыли продукта во времени. Классическое определение указывает, что **жизненный цикл** отражает «ход продаж и прибыли продукта на протяжении его рыночной жизни» [9].

Модель **жизненного цикла** была систематизирована Теодором Левиттом, который выделил стадии разработки, внедрения, роста, зрелости и спада, подчеркнув ограниченность коммерческой жизни любого продукта [21].

На стадии **внедрения** продукт только выводится на рынок, объем продаж невелик, а прибыль, как правило, отсутствует из-за высоких издержек на маркетинг и информирование потребителей [9].

На стадии **роста** наблюдается ускоренное увеличение продаж, формирование устойчивого спроса и появление конкурентов, что требует активного управления позиционированием и каналами распределения [22].

Стадия **зрелости** характеризуется насыщением рынка, стабилизацией объемов продаж и усилением конкурентной борьбы, что приводит к необходимости дифференциации продукта и оптимизации затрат [9].

На стадии **спада** спрос на продукт сокращается вследствие технологического устаревания, изменения предпочтений потребителей или появления заменителей, что приводит к снижению продаж и решению о снятии продукта с рынка [21].

Маркетинговая модель **жизненного цикла** используется как инструмент стратегического управления продуктовым портфелем, позволяя соотносить инвестиции и ожидаемые доходы с этапами развития продукта [12].

Problem-Solution Fit на ранних стадиях. На начальных стадиях **жизненного цикла (ЖЦ)** новый продукт сталкивается прежде всего с задачей соответствия

решаемой им проблеме пользователя. Иными словами, необходимо подтвердить, что разрабатываемый продукт действительно решает значимую проблему целевого пользователя лучше или эффективнее существующих альтернатив. Данное соответствие продукта потребности пользователя часто называют *problem-solution fit* - соответствие «продукт-проблема» или «проблема-решение» [50].

С точки зрения бизнеса, достижение *problem-solution fit* означает, что продукт приносит реальную пользу хотя бы небольшому числу пользователей: выявлена существенная проблема и предложено работоспособное решение, которое эти пользователи готовы принять. Исследования в области инноваций показывают, что многие стартапы терпят неудачу именно из-за несформированного соответствия между продуктом и реальной проблемой клиента - компании зачастую слишком рано фокусируются на разработке и масштабировании продукта, минуя стадию глубокого изучения проблем пользователей и проверки гипотез ценности [51].

Поэтому на этапе зарождения идеи и разработки концепции ключевым является поиск и верификация гипотезы ценности: идентификация конкретной проблемы пользователя, формирование ценностного предложения и создание минимальной версии продукта (прототипа, MVP), способного подтвердить, что решение работает и приносит пользу [52]. По сути, *problem-solution fit* - это ранний показатель жизнеспособности идеи: наличие доказательства, что продукт решает важную проблему клиента.

Product-Market Fit на стадиях роста и масштабирования. Если на первом этапе достигнуто подтверждение ценности продукта для пользователя, следующая стратегическая задача - масштабировать эту ценность, доставлять продукт широкой аудитории. Эту стадию соответствия продукта рынку называют *product-market fit*, что подразумевает гармоничное попадание продукта в потребности массового рынка, когда созданное решение удовлетворяет значимому числу клиентов и спрос начинает быстро расти. Термин *product-market fit* («соответствие продукта рынку») был введен М. Андресеном в 2007 г. [55], и именно достижение этого состояния он считал определяющим фактором жизнеспособности стартапа.

В состоянии *product-market fit* продукт обладает признанной ценностью: клиенты будут разочарованы, если продукт исчезнет, наблюдается вирусный эффект распространения - довольные пользователи рекомендуют его другим, и компания переходит к фазе бурного роста.

В отличие от *problem-solution fit*, который фокусируется на концепции и первых пользователях, *product-market fit* характеризует более позднюю стадию ЖЦ - когда продукт выверен и готов удовлетворять потребности широкого

сегмента рынка. Достижение product-market fit означает, что **продукт полезен на масштабе**: налажен устойчивый поток новых пользователей, высока их удерживаемость и удовлетворенность, и бизнес-модель обеспечивает рост [54]. Как отмечают практики, прежде чем переходить к стратегическому расширению бизнеса и наращиванию производства, необходимо убедиться, что продукт действительно соответствует требованиям рынка. Иными словами, продукт, не нужный широкой аудитории, не взлетит даже при значительных маркетинговых затратах, поэтому сначала закладывается фундамент ценности для пользователя, а уже потом - масштабируется доставка этой ценности на рынок.

Таким образом, задачи problem-solution fit и product-market fit привязаны к различным фазам жизненного цикла продукта. На **этапах зарождения и разработки** (идея, предпроектное исследование, опытный образец) бизнес-система вокруг продукта решает задачу **“problem-solution fit”** - проверяет гипотезу о том, что продукт решает проблему пользователя. Здесь происходят генерация концепции, создание MVP, пилотное тестирование на ранних клиентах, итерации по обратной связи. Успешное прохождение этой фазы завершается появлением продукта, приносящего реальную пользу хотя бы небольшой группе потребителей.

Далее, на **этапах внедрения, роста и зрелости** продукта, фокус смещается на задачу **“product-market fit”** - экспансия на рынок, адаптация продукта и бизнес-модели для масштабирования охвата. Бизнес-система на этих стадиях наращивает производство и сбыт, выстраивает каналы дистрибуции, маркетинг и поддержку, обеспечивая доставку ценности уже широкому кругу пользователей с сохранением качества решения их проблемы. Происходит тиражирование продукта, выход на новые сегменты и рынки, усиление конкурентных преимуществ - всё для того, чтобы продукт занял устойчивое место на рынке.

Графически эти этапы иногда представляют как последовательность: **Problem-Solution Fit → Product-Market Fit**, через которую проходит большинство успешных инноваций. Иными словами, **сначала** подтверждается соответствие продукта индивидуальной проблеме (ценность для пользователя), **затем** - соответствие продукта рынку (масштабируемая ценность), и **после** этого продукт переходит в режим экспоненциального роста (масштабирование бизнеса).

1.4.2. Инженерное понимание жизненного цикла продукта

В инженерии **ЖЦ продукта** трактуется как **совокупность процессов, охватывающих полное существование продукта**. Согласно стандарту ISO/IEC/IEEE 15288, жизненный цикл системы (продукта) включает стадии

концепции, разработки, производства, эксплуатации, поддержки и вывода из эксплуатации [8].

Для программных продуктов аналогичный подход закреплен в стандарте ISO/IEC/IEEE 12207, который описывает процессы жизненного цикла программного обеспечения от формирования требований до прекращения поддержки [23].

Инженерный подход рассматривает жизненный цикл как управляемый процесс, в рамках которого принимаются решения о технической архитектуре, производственных технологиях, сервисном обслуживании и утилизации продукта [Blanchard, *Systems Engineering and Analysis*, 2008]. В отличие от маркетинга, инженерный жизненный цикл начинается до появления продукта на рынке и продолжается после прекращения его продаж.

1.4.3. Сравнение маркетингового и инженерного подходов

Маркетинговый и инженерный подходы к жизненному циклу продукта фокусируются на разных аспектах его существования.

Таблица 1.1. Сравнение маркетингового и инженерного подхода

| Аспект | Маркетинговый подход | Инженерный подход |
|-----------------|---|---|
| Цель и фокус | Управление жизненным циклом продукта: спрос, продажи, доля рынка, конкурентные стратегии на разных стадиях. Продукт рассматривается с точки зрения коммерческого успеха . | Управление технической жизнью продукта: процессы разработки, производства, эксплуатации и утилизации. Продукт рассматривается как техническая система , требующая поддержки на всех этапах. |
| Типичные стадии | Внедрение → Рост → Зрелость → Спад (плюс возможная предшествующая стадия разработки). Каждая стадия характеризуется особенностями спроса и конкуренции. | Концепция → Разработка → Производство → Эксплуатация → Поддержка → Вывод из эксплуатации . Возможна группировка в фазы BOL/MOL/EOL для управления жизненным циклом. |

| | | |
|--------------------------------------|--|--|
| <p>Критерии успеха</p> | <p>Маркетинговые показатели: темпы роста продаж, выручка, прибыль по продукту, рыночная доля, узнаваемость бренда. ЖЦП служит для прогнозирования продаж и планирования маркетинговых инвестиций [Kotler, 2016].</p> | <p>Технические и экономические показатели: выполнение требований по функциональности и качеству, сроки и бюджет разработки, производственная себестоимость, надежность в эксплуатации, затраты на обслуживание, срок службы. ЖЦП служит для минимизации совокупных затрат и рисков на протяжении жизни продукта.</p> |
| <p>Управленческие решения</p> | <p>В разрезе стадий ЖЦ продукта принимаются решения о маркетинговой стратегии: запуск новых рекламных кампаний, изменение цены (например, стратегия «снятия сливок» на внедрении vs. снижения цены на спаде), расширение ассортимента или прекращение продаж. Инструменты планирования - матрица БКГ, модели диффузии инноваций, прогнозирование продаж по этапам ЖЦП.</p> | <p>В разрезе фаз ЖЦ продукта принимаются решения о технической и производственной стратегии: выбор архитектуры и технологий, организация производства (вплоть до аутсорсинга и массовости), планирование ТОиР (техобслуживания и ремонтов) во время эксплуатации, стратегия окончания поддержки. Инструменты - стандарты и регламенты (ISO 15288 и др.), системы PLM, методики как Stage-Gate для R&D.</p> |
| <p>Окончание цикла</p> | <p>Спад и вывод с рынка: снижение продаж ведет к решению о снятии продукта с продажи. Возможны варианты продления жизни через модификации или нахождение новых рынков, но в конечном итоге продукт “умирает” на рынке, уступая место новым товарам. Маркетинговый цикл, как правило, заканчивается раньше, чем продукт полностью исчезнет физически (например, модель автомобиля может продолжать эксплуатироваться клиентами после прекращения продаж).</p> | <p>Вывод из эксплуатации и утилизация: после окончания службы продукт демонтируется, утилизируется или заменяется новой версией. Инженерный цикл включает мероприятия по безопасному удалению продукта из эксплуатации: например, переработка материалов, переназначение компонентов, миграция данных пользователей на новое ПО. Цель - минимизировать потери и воздействие при завершении жизни изделия (принципы устойчивого развития и « cradle-to-cradle »).</p> |

Маркетинговый подход ориентирован на отдельные стадии ЖЦ продукта, на которых с продуктом взаимодействуют рынок и потребитель, отражая эти взаимодействия, а также динамику экономических показателей [9].

Инженерный подход ориентирован на все стадии ЖЦ продукта, особо фокусируясь на стадии разработки, производства, внедрения, технической поддержки и ремонта продукта. Инженерный подход в первую очередь рассматривает продукт как объект проектирования, производства и эксплуатации, требующий системного управления на протяжении всего срока службы [8].

Эти подходы не противоречат друг другу: понимание рыночной стадии продукта важно для планирования инженерных изменений, а технические ограничения жизненного цикла определяют возможности его коммерческого продления [25].

1.4.4. Примеры жизненного цикла продукта

Для **физического продукта** (например, бытовой техники) маркетинговый жизненный цикл может завершиться значительно раньше, чем физическое использование изделия, которое продолжается в фазе эксплуатации и обслуживания [9, 24].

Для **цифровых продуктов** жизненный цикл часто продлевается за счет обновлений и версионности, однако с инженерной точки зрения он завершается прекращением поддержки и эксплуатации программного кода [26].

Для **сервисов** жизненный цикл проявляется в эволюции бизнес-процессов и цифровых платформ, при этом завершение жизненного цикла сервиса обычно связано не с его исчезновением, а с трансформацией в новый сервис или продукт [27].

1.5. Системный оператор в классической ТРИЗ

Развитие системного мышления было провозглашено Г. С. Альтшуллером в качестве конечной цели обучения АРИЗ (Алгоритму решения изобретательских задач) [28]. Для формирования такого мышления в ТРИЗ используется «системный оператор», представляющий собой 9-экранную (позднее, многоэкранную) модель представления системы. Данная схема предполагает рассмотрение технической системы не изолированно, а в более широком контексте - на разных уровнях и в разные моменты времени [28].

1.5.1. Системные переходы

От системы к надсистеме. Этот переход означает смену фокуса с самой рассматриваемой системы на более крупную систему, частью которой она является, то есть на её надсистему. Нередко задача может быть решена эффективнее именно на уровне надсистемы - посредством изменения окружающей или высшей системы вместо модификации самой исходной системы [28]. Иными словами, важно видеть более широкий контекст: «когда речь идет о дереве, надо видеть и лес» - понимать окружение, в котором существует данная система [28].

От системы к подсистемам. Этот переход фокусирует внимание на внутренних элементах и компонентах исходной системы. Подход «снизу-вверх» позволяет искать решение, изменяя или усовершенствуя отдельные части системы, а не всю систему целиком [28]. Талантливый изобретатель учитывает структуру системы и при анализе «видит отдельные клетки древесины, а не только само дерево», то есть детально представляет устройство системы и свойства ее частей [28].

В классическом системном операторе переход от системы к подсистемам предполагает единственный вид декомпозиции системы на подсистемы.

От системы в настоящем к системе в прошлом. Этот переход заключается в анализе предшествующих этапов развития системы, её истории. Необходимо рассмотреть, как система выглядела и функционировала раньше, какие изменения она уже претерпела со временем [28]. Такой исторический взгляд позволяет выявить тенденции развития и понять, какие подходы уже пробовались или могли бы возникнуть в прошлом, что дает ценные подсказки для нынешней проблемы. В методологии ТРИЗ предусмотрен анализ эволюции системы, т.е. изучение тенденций развития техники, благодаря чему изобретатель осознаёт, в каком направлении система развивалась до настоящего момента [28]. Умение видеть прошлое системы - редкое, но именно оно характеризует более высокий уровень системного мышления [28].

От системы в настоящем к системе в будущем. Этот переход означает прогнозирование и учёт потенциальных будущих состояний и развития системы. Изобретателю важно уметь представить, как система может выглядеть или какие функции приобрести в перспективе [28]. Этот взгляд вперёд расширяет поиск решений, стимулируя не ограничиваться локальными улучшениями, а стремиться к опережающему развитию системы в соответствии с её возможной эволюцией. Альтшуллер подчеркивает, что видение будущего системы наряду с её прошлым - характерный признак выдающегося творческого мышления [28].

Переход от системы к анти-системе. Помимо рассмотрения над- и подсистем, в системном операторе ТРИЗ учитывается также и **переход от системы к анти-системе**, то есть анализ системы-«противоположности». Анти-системой в ТРИЗ называется система, выполняющая функцию, противоположную функции исходной системы [28]. Подобный переход особого значения приобретает в ситуациях, когда имеющаяся система исчерпала возможности своего развития и требуется принципиально новый подход к решению задачи [28]. В целом, анализ антисистемы расширяет рамки рассмотрения проблемы и способствует более полному учёту противоречивых факторов, влияющих на функционирование исходной системы.

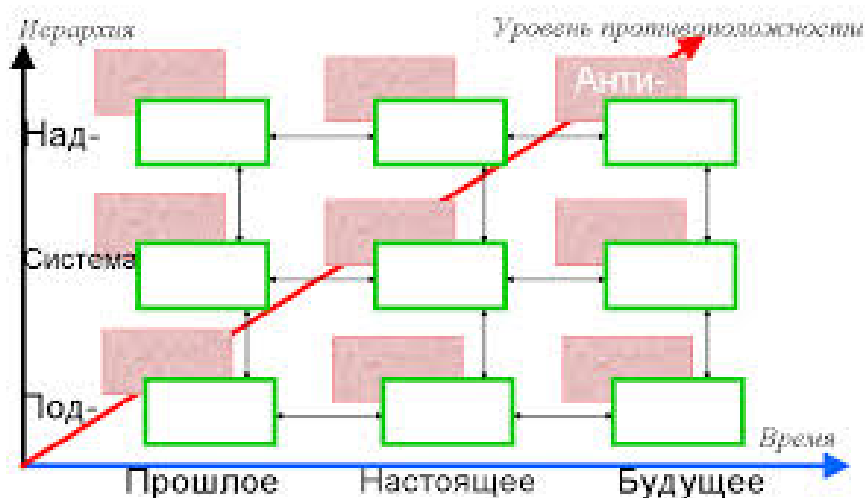


Рисунок 1.1. 9-экранная схема мышления (системный оператор) классической ТРИЗ [Хоменко, 2013]

Г.С. Альтшуллер подчеркивал, что ограничение анализа только самой системой («картинкой в одном экране») является признаком узкого, «слабого» мышления, тогда как использование всех экранов многоэкранной схемы, включая рассмотрение антисистем, соответствует «сильному» системному мышлению [28]. Именно такой всесторонний подход, охватывающий системы, надсистемы, подсистемы, временные изменения и антисистемы, лежит в основе развития системного мышления - конечной цели обучения АРИЗу [28].

1.5.2. Примеры применения

В работе 1975 года приведены наглядные примеры применения принципов системного оператора. Один из них связан с упомянутой выше задачей перемещения чрезвычайно тяжёлой трубы с помощью крана. Традиционный подход заключался бы в поиске способа усилить кран (например, увеличить его грузоподъёмность). Однако, применяя системный оператор, изобретатель рассматривает альтернативные экраны: изменяет надсистему (организацию всего строительного процесса) или затрагивает подсистему (использует иной

материал или конструкцию для крана), обходясь без тривиального усиления исходной системы [28]. Таким образом, задача решается не лобовым усилением имеющегося оборудования, а системным преобразованием условий задачи, что приводит к более эффективному и порой неожиданному решению. Данный пример демонстрирует применение вертикального анализа (по уровням) в сочетании с поиском нетривиальных изменений для достижения цели.

Другой пример касается принципа антисистем. Рассматривая проблему улучшения ледоколов, системный оператор предлагает взглянуть на противоположную постановку вопроса - придумать «*антиледокол*». Антиледокол - это гипотетическая система, которая должна обеспечить проводку судов через льды, но принципиально без ломки льда. Такой подход на практике может привести, например, к идеям использования новых физических принципов воздействия на лёд (расплавление, раздвигание льдин, обход ледовых полей и т.д.) вместо механического разрушения. Приведение этого примера Альтшуллером иллюстрирует, как поиск антисистемы стимулирует инновации, заставляя изобретателя отказаться от эволюционного доведения существующей техники до предела и подумать о радикально ином решении [28].

Применение динамического изменения масштаба можно проиллюстрировать ситуациями, когда решение появляется при рассмотрении объекта в экстремальных размерных вариантах. Хотя в тексте 1975 года отдельно не приводится конкретный пример по этому аспекту, сам метод мысленного «сжатия и расширения» системы широко используется в изобретательских задачах. Например, проблемы, неразрешимые в привычном масштабе, могут получить решение при микро- или макро-рассмотрении: миниатюризация устройства открывает новые физические эффекты, а укрупнение системы позволяет задействовать ранее незначимые факторы. Такой подход согласуется с требованием Альтшуллера видеть элементы «пульсирующими» - меняющими размер и свои свойства в представлении решателя [28].

1.6. Онтогенез и филогенез в системном операторе

1.6.1. Ось жизненного цикла системы в системном операторе

Существенное развитие концепции системного оператора было предложено М. С. Рубиным, который ввел в методологию ТРИЗ понятия *онтогенеза* и *филогенеза* технической системы. Эти понятия, заимствованные из биологии, отражают два типа изменения системы во времени: индивидуальное развитие конкретного объекта (онтогенез) и историческую эволюцию рода однотипных систем (филогенез) [45]. Рубин обосновал необходимость учета обеих шкал

времени в анализе: системный оператор должен рассматриваться не только в контексте эволюции класса подобных систем во времени, но и в контексте жизненного цикла отдельной системы. Соответственно, в расширенной модели системного оператора предлагается вводить две временные оси - онтогенетическую (отражающую изменения конкретного продукта или технической системы на протяжении её индивидуального существования) и филогенетическую (отражающую последовательность поколений систем данного типа в историческом развитии) [45]. Такое дополнение обеспечивает более полный охват различных типов развития: от единичной эволюции продукта до “родовой” эволюции технических решений.

1.6.2. Онтология системного оператора и вариативность экранов

В 2019-2020 гг. группа исследователей (А. Курьян, М. Рубин, О. Эккардт, Н. Щедрин, Н. Рубина) осуществила формализацию знаний ТРИЗ в виде онтологий, включая онтологию системного оператора [46]. В рамках этой работы были интегрированы обе шкалы времени, предложенные Рубиным: в онтологической модели системного оператора зафиксировано наличие двух временных осей - для онтогенеза и филогенеза системы. Более того, исследователи обратили внимание на **вариативность переходов между экранами** классической многоэкранки. В онтологии отмечено, что системный оператор обладает свойством неоднозначности траекторий анализа: последовательность переходов по экранам может различаться, и в зависимости от выбранной траектории состав и содержание экранов формируются по-разному [46]. Иными словами, модель не жестко фиксирует порядок рассмотрения уровней и временных срезов - аналитик волен перемещаться по экранной сетке различными путями, что приводит к вариативным аспектам анализа системы. Фиксация этого свойства в онтологии позволила формализовать ранее неявные возможности метода и подчеркнула гибкость системного оператора как инструмента исследования эволюции систем.

Таким образом, обосновано расширение классической модели: наряду с базовыми девятью экранами (подсистема-система-надсистема × прошлое-настоящее-будущее) официально учитываются две разновидности временного анализа и допускаются различные сценарии переходов между экранами при проведении системного анализа.

1.7. Инструменты ТРИЗ, использующие ЖЦ системы

1.7.1. Метод анализа главных параметров ценности (MPV)

В данном разделе представлен обзор и анализ инструмента ТРИЗ MVP-анализ. И хотя данный инструмент непосредственно не связан с

системным оператором, в инструменте применены понятия, такие как, продукт, стекхолдер, польза (ценность) продукта для стейкхолдера, параметр ценности, и используются связи между этими понятиями и понятиями ТРИЗ: система, полезная функция, подсистемы.

Метод анализа главных параметров ценности (MPV) - инструмент ТРИЗ, применяемый для выявления ключевых потребительских ценностей продукта или системы с позиции разных участников (стейкхолдеров) и последующей постановки задач по улучшению системы [56]. Ниже рассмотрено: какие стейкхолдеры учитываются во время MPV-анализа, как они выявляются, пошаговая схема определения их требований и как результаты анализа переводятся в конкретные инженерно-проектные задачи.

Типы учитываемых стейкхолдеров (внешние и внутренние). При MPV-анализе рассматривают широкий круг стейкхолдеров системы - как внешних, так и внутренних [57]. К внешним стейкхолдерам относят прежде всего конечных пользователей или клиентов продукта, а также заказчиков/покупателей (если они не совпадают с пользователями), регуляторов (надзорные органы), бизнес-партнеров (например, дистрибьюторов), местное сообщество и даже конкурентов на рынке.

К внутренним стейкхолдерам обычно относят тех, кто непосредственно связан с созданием, эксплуатацией и обслуживанием системы: владельцы и руководство компании, разработчики и производственные отделы, сотрудники сервисных/обслуживающих служб, операторы продукта, отделы типа IT, склад, НИОКР и т.д. Таким образом, учитываются все группы лиц или организаций, чьи интересы затрагивает данный продукт или процесс (от собственников и персонала до конечных потребителей и контролирующих организаций).

Выявление стейкхолдеров через MPV-анализ. Идентификация стейкхолдеров - один из первых шагов MPV-анализа. В классических алгоритмах GEN3/TRIZ сначала определяется объект улучшения (продукт или процесс) и формулируется бизнес-задача, а затем явно перечисляются стадии жизненного цикла системы и связанные с ними стейкхолдеры, целевые рыночные ниши и типичные ситуации использования [58]. Другими словами, анализируется окружение системы: на каждом этапе жизненного цикла (разработка, производство, сбыт, эксплуатация, утилизация) выявляются участники, взаимодействующие с системой, и их роли. Для каждой важной стадии и каждого ключевого стейкхолдера могут строиться функциональные модели, показывающие, как данный стейкхолдер взаимодействует с системой и какие потребности/ценности у него возникают [57].

Существует правило: MPV формулируются с позиции пользователя системы. Если пользователь - конечный потребитель, то главные параметры ценности

отражают именно ценности клиента [58]. В случае же, когда анализируется вспомогательная или внутренняя система (например, производственное оборудование, программный комплекс для внутренних нужд), в роли «пользователя» выступает ответственное лицо за данный процесс, и MPV определяются исходя из структуры ценности этого внутреннего стейкхолдера [56]. Проще говоря, MPV конечного продукта ориентированы на конечного клиента, а MPV подсистем или средств производства - на требования владельца процесса (например, отдела производства, логистики, IT и т.п.). Это позволяет через MPV-анализ явно выявить, кто является ключевым стейкхолдером для данной системы и чьи критерии ценности нужно учитывать в первую очередь.

Правило MVP анализа, устанавливающее приоритет требований потребителя над требованиями других стейкхолдеров, хорошо соотносится с продуктовым подходом, но противоречит системной инженерии, в которой требования всех стейкхолдеров (на всех стадиях жизненного цикла системы) должны быть выявлены и реализованы в системе.

Алгоритм определения требований стейкхолдеров. Определение требований стейкхолдеров с помощью MPV-анализа происходит поэтапно. Ниже представлена логическая схема (алгоритм):

1. Сбор потребностей стейкхолдеров. Сначала проводится тщательное исследование потребностей и ожиданий выявленных стейкхолдеров. Применяются методы сбора «голоса клиента/пользователя»: опросы, интервью, изучение отзывов, фокус-группы, анализ данных использования и т.д. [59]. Цель - понять, какие параметры продукта важны для разных участников. На этом этапе формируется полная картина требований различных сторон к системе.
2. Выявление ключевых требований (ценностных параметров). На основе собранной информации составляется перечень основных требований, которые стейкхолдеры считают наиболее значимыми для ценности продукта [56]. Эти требования представляют собой потенциальные параметры ценности (PVs) - характеристики продукта или системы, влияющие на удовлетворенность стейкхолдеров. Каждый параметр можно количественно оценить и ранжировать по степени влияния на удовлетворение потребностей или на решение о покупке/использовании. Таким образом определяется, какие требования являются критически важными (наиболее влияют на ценность с точки зрения каждого стейкхолдера).
3. Анализ соответствия системы требованиям. Далее анализируется, насколько текущий продукт или система удовлетворяет выявленным ключевым требованиям. Для этого сопоставляют существующие свойства/характеристики продукта с ожиданиями стейкхолдеров. Такой

анализ часто выявляет пробелы и проблемные точки - несоответствия, где ожидания не оправдываются (так называемые «болевы точки» клиента) [57]. Например, стейкхолдер может ждать более высокой надежности и удобства, чем обеспечивает текущая система. Выявленные разрывы указывают, какие параметры ценности недостаточно реализованы или требуют улучшения.

4. Определение главных параметров ценности (MPV). Из списка требований выделяются главные параметры ценности - те, которые создают наибольшую ценность для стейкхолдера и критически важны для успеха продукта. Главные параметры обычно отвечают на вопрос: «Что является решающим для того, чтобы стейкхолдер счел продукт ценным?» [58]. В контексте ТРИЗ MPV можно понимать как ключевые потребительские ценности, выраженные через набор ключевых требований. Таким образом, MPV — это обобщенное представление самых важных требований определенного стейкхолдера к системе, например: безопасность использования, экономичность эксплуатации, удобство обслуживания, минимальная стоимость владения, максимальная прибыльность и т.п. для разных групп.
5. Проверка на противоречия требований. Важная особенность MPV-анализа - выявление возможных противоречий между требованиями разных стейкхолдеров либо между требованиями и возможностями системы [56]. На этом шаге анализируются отношения между главными параметрами ценности разных заинтересованных сторон: не конфликтуют ли они друг с другом. Часто проблемы в системе обусловлены именно конфликтом интересов (требований) - например, маркетинг хочет увеличить функциональность для клиента, а производство требует снизить себестоимость, или потребитель желает высокое качество, а владелец бизнеса нацелен на максимальную прибыль при минимальных затратах. MPV-анализ позволяет наглядно демонстрировать такие противоречия требований стейкхолдеров. Суть MPV-анализа, таким образом, сводится к выявлению требований стейкхолдеров по отношению к системе и проверке, не противоречат ли эти требования друг другу. Если обнаружены противоречия, фиксируется необходимость их разрешения - это напрямую переходит в постановку задач (см. следующий раздел).
6. Приоритизация и формализация результатов. На заключительном этапе анализа формируется структура ценности: перечень главных параметров ценности (MPV) с указанием, для каких стейкхолдеров они важны, и описание выявленных несоответствий или конфликтов. Эти результаты затем используются для формулировки конкретных проектных целей и задач улучшения системы [57].

Предложенный в [56] алгоритм не учитывает полный жизненный цикл продукта и многообразие сценариев взаимодействия стейкхолдеров с продуктом на всех стадиях ЖЦ. В этой части MVP-анализ нуждается в улучшении.

Тем не менее, мы отмечаем, что в MVP-анализе прослеживается связь с иерархической осью классического системного оператора. Анализ стейкхолдеров и их потребностей позволяет по-новому рассматривать, что из себя представляет надсистема. Анализ этапов жизненного цикла продукта подтвердил необходимость добавления в системный оператор новой оси - оси жизненного цикла, как было предложено М.С. Рубиным [45].

между что он позволяет перевести язык клиентов и стейкхолдеров на язык инженеров. Выделенные главные параметры ценности и несоответствия трансформируются в четко обозначенные инженерные проблемы, над которыми и предстоит работать команде с инженерной системой.

1.7.2. Обратный функциональный анализ

Обратный функциональный анализ (ОФА) - это метод, предложенный Н. Е. Сауниным (2019) [61] для решения организационно-управленческих задач, который отличается от классического функционального анализа *обратной* логикой проведения анализа. В классическом подходе анализ обычно начинается с рассмотрения имеющейся системы и ее компонентов, определяя какие функции они выполняют. Такой подход в сложных системах приводит к тому, что разные аналитики могут получить разную картину - процесс недостаточно формализован, результаты нестабильны, трудно гарантировать полноту модели. **Обратный** же подход, напротив, предлагает начать не с компонентов, а с целевой функции системы. Сначала выявляются необходимые функции, и лишь затем определяются компоненты для их реализации. Таким образом, ОФА меняет порядок шагов классического анализа, фокусируясь на целях системы прежде, чем на ее устройстве.

Алгоритм ОФА приспособлен для организационно-управленческих задач и развивается *от главной полезной функции (ГПФ) - к функциям - к компонентам системы*. В обобщенном виде он включает следующие шаги:

1. **Формулирование ГПФ и объекта воздействия.** Определяется назначение системы: ее главная полезная функция(и) и объект(ы), на который направлено это воздействие. Иными словами, выясняется, *для чего* создается система и какой результат она должна обеспечить. Результатом шага является перечень главных функций системы и тех параметров объекта, которые эти функции должны изменять ради достижения цели.

2. **Выделение этапов функционирования.** Определяются основные этапы работы системы - последовательность фаз или состояний, через которые она проходит, выполняя свою ГПФ. Например, для автомобиля помимо собственно движения существуют этапы заправки, стоянки, технического обслуживания и т.д.; для отдела продаж - поиск клиентов, заключение сделки, постпродажное сопровождение и т.п.. Важно перечислить все значимые этапы, чтобы охватить целостно процесс функционирования. Итогом является набор этапов, которые система должна успешно пройти для реализации своей главной функции.
3. **Декомпозиция функций по этапам.** Для каждого выделенного этапа выясняется, *какие функции необходимо выполнить*, чтобы данный этап был успешно пройден. Фактически происходит разбиение ГПФ на подфункции, привязанные к этапам. Например, на этапе «поиск клиентов» отдел продаж должен выполнять функции сбора информации, ее анализа, первичного контакта с клиентом и т.д.; на этапе «заключение сделки» - функции подготовки и подписания договора, согласования условий и пр. Результатом становится перечень обязательных функций на каждом этапе функционирования системы.
4. **Определение необходимых компонентов.** Далее для каждой функции на каждом этапе определяются компоненты системы, которые способны эту функцию реализовать. Под компонентами понимаются как элементы конструкции или ресурсы, так и ответственные исполнители, инструменты, регламенты - все, что требуется для выполнения функции. Например, для функции «сбор информации о клиентах» компонентами будут: менеджер (или отдел), база данных клиентов, инструменты аналитики, регламенты отбора информации и т.п. На этом шаге формируется состав системы - набор компонентов, необходимых для реализации всех ранее выявленных функций.
5. **Выстраивание структуры и функциональных связей.** После выявления всех необходимых компонентов полезно проанализировать, как они взаимодействуют между собой, т.е. построить структурную схему системы и при необходимости функциональную модель взаимодействия компонентов. Этот шаг обеспечивает понимание того, каким образом компоненты совместно реализуют главную функцию и подконтрольные функции на каждом этапе.

Ключевым результатом применения ОФА является **образ минимально работоспособной системы**. Поскольку метод фокусируется только на тех функциях, без которых цель системы недостижима, и на тех компонентах, без

которых эти функции не реализовать, по итогам анализа получается структурная схема «*минимально необходимой*» технической или организационной системы. Иными словами, ОФА отфильтровывает все второстепенные элементы, выделяя **только обязательные компоненты для выполнения критически важных функций**. Например, если рассмотреть автомобиль, чья ГПФ - перемещение из точки А в точку Б, то минимально работоспособная версия автомобиля будет включать лишь узлы, необходимые для езды: раму, колеса, двигатель, рулевое управление, тормоза, сиденье водителя и т.п. При этом необязательные для движения элементы - кузов, фары, кондиционер, аудиосистема и др. - в минимальной системе отсутствуют, так как автомобиль способен выполнить свою главную функцию и без них. Такой «скелет» системы задает основу, которую затем можно дополнять вторичными функциями или улучшениями по мере необходимости (например, добавление кондиционера и магнитолы в автомобиль - это надстройки, улучшающие потребительские параметры, которые можно обосновать отдельным анализом параметров ценности после определения ядра системы).

Определение минимально необходимой структуры системы дает возможность целенаправленной оптимизации: **выявления слабых звеньев и точек для улучшения**. Проанализировав каждый обязательный компонент, можно установить, какие из них не справляются в достаточной мере со своей функцией или ограничивают эффективность всей системы. Такие «отстающие» компоненты рассматриваются как слабые звенья системы. Зная их, можно разработать программу усиления или замены этих звеньев, тем самым оптимизируя систему в целом.

Методологически ОФА согласуется с подходом системного оператора (многоэкранкой). Системный оператор в классической ТРИЗ - это средство рассмотрения системы во времени (Прошлое-Настоящее-Будущее) и в иерархическом окружении (Подсистема-Система-Надсистема). ОФА, со своей ориентацией на целевую функцию и многоуровневой декомпозицией, поддерживает те же принципы системного мышления. Во-первых, *ориентация на целевую функцию* соотносится с фокусом системного оператора на назначении системы в надсистеме. ОФА фактически начинает анализ с вопроса: «Для чего существует система, какую главную задачу она выполняет?», что соответствует рассмотрению системы во взаимосвязи с надсистемой (ее роль и ценность в более широком контексте). Во-вторых, *декомпозиция по уровням* в ОФА (разбиение системы на функции и компоненты) отражает тот же принцип, что и вертикальная ось системного оператора: рассмотрение подсистем, из которых состоит система, и надсистемы, частью которой она является. Выявленные в ходе ОФА компоненты и их функции по сути формируют «экран» подсистемного уровня, а понимание главной функции и объекта воздействия выводит нас на

надсистемный уровень (показывая, частью какой более широкой системы является анализируемый объект).

В-третьих, *учет этапов функционирования*, по сути, является прообразом оси жизненного цикла в системном операторе. ОФА специально требует зафиксировать все этапы работы системы во времени (от начала процесса до конца), тем самым гарантируя рассмотрение системы *в динамике*. Таким образом, метод ОФА удерживает целостную модель системы во времени и пространстве - он заставляет видеть систему одновременно по *вертикали* (уровни системы: от компонентов до надсистемы через главную функцию) и по *горизонтали* (этапы и последовательность функционирования во времени). Это полностью соответствует принципу, на котором основывается системный оператор: мыслить о системе комплексно, не упуская ни структурного, ни временного измерения.

1.7.3. Анализ ключевых параметров успешности

Инструмент для анализа ключевых параметров успешности (АКПУ) основывается на выявлении и анализе **ключевых параметров успешности** продукта через призму требований заинтересованных сторон (стейкхолдеров). В основе методики лежит ряд определений. **Стейкхолдер** определяется как индивид или группа, от которых зависит успешное прохождение системой конкретной стадии ее жизненного цикла и которые в случае неудовлетворения своих ожиданий способны отказаться от взаимодействия с системой в пользу альтернативы. Соответственно, вводится понятие **«ключевые требования стейкхолдера» (КТС)** - это диапазоны допустимых значений определенных параметров системы или процесса, нарушение которых ведет к отказу стейкхолдера от использования данной системы. На базе этих требований выделяются **«ключевые параметры успешности» (КПУ)** - параметры самой системы, от значений которых зависит выполнение ключевых требований и, в конечном счёте, достижение системой успешности. Иными словами, КПУ представляют собой те характеристики продукта, на которых фокусируются основные ресурсы разработки, поскольку именно за счёт улучшения этих характеристик планируется обеспечить конкурентное преимущество системы.

С точки зрения целей продукта, акцент на КПУ означает концентрацию на реализации его целевой функции и ценности для заказчика. Авторами методики АКПУ подчеркивается, что **успешная система** - это такая система, которая обеспечивает достижение цели заказчика в требуемой мере и превосходит по этому показателю доступные альтернативы. Следовательно, определяя КПУ, разработчики фактически идентифицируют измеримые аспекты выполнения продуктом своей главной функции, наиболее значимые для стейкхолдера и отличающие продукт от конкурентов. Таким образом, методология нацелена на обеспечение того, чтобы проектируемая система

удовлетворяла ключевые ожидания пользователей и других участников, прямо связанных с ее успешностью, не распыляя усилия на второстепенные параметры.

Принципиальной особенностью подхода АКПУ является **учёт стейкхолдеров на всех стадиях жизненного цикла продукта**. Каждый этап жизненного цикла системы характеризуется собственным набором задач и требований, а потому и своим «набором» стейкхолдеров - от потребителей и пользователей на стадии эксплуатации до обслуживающего персонала на стадии сопровождения или регуляторов на стадии вывода продукта на рынок. Методика АКПУ требует выявления ключевых требований этих стейкхолдеров для всех релевантных стадий, что обеспечивает полноту видения успешности. Работа со стейкхолдерами на каждой стадии жизненного цикла позволяет заранее обнаружить потенциальные разрывы: несоответствие ожиданиям на любом этапе может привести к отказу определённой группы стейкхолдеров от продукта, ставя под угрозу общий успех. Таким образом, вовлечение заинтересованных сторон и анализ их ключевых требований на всём протяжении жизненного цикла - от концепции и разработки до использования и утилизации - выступает необходимым условием создания конкурентоспособной и устойчивой системы.

Методологически подход АКПУ тесно перекликается с инструментом **системного оператора** (классической многоэкранки). В АКПУ рассматривается в контексте подсистем и надсистем. В классической многоэкранке уровень надсистемы отражает окружение системы (рынок, пользователи, внешние условия), а уровень подсистем - внутреннее устройство. Подход АКПУ учитывает надсистемный контекст через анализ альтернатив и внешних факторов успешности: как было отмечено, успешность определяется сравнительно - относительно доступных альтернативных решений. Тем самым, выявляя КПУ, аналитики фактически рассматривают продукт в его надсистеме, т.е. на фоне конкурентной среды и потребностей внешнего мира. Различные стейкхолдеры, действующие на надсистемном уровне, могут предъявлять разнонаправленные ожидания, и АКПУ позволяет их явно зафиксировать. Одновременно методика не игнорирует и **подсистемный уровень**: для выполнения конкретных КПУ могут быть ответственны отдельные компоненты или модули продукта, следовательно, определение ключевых параметров успеха задает направления развития или улучшения подсистем.

В части временного анализа в АКПУ присутствует прообраз новой оси в системном операторе, отражающей стадии ЖЦ системы или продукта. АКПУ органично вписывается в такую временную структуру: проводя анализ, команда оценивает успешность продукта на каждой выделенной стадии жизненного цикла.

В совокупности это создает многомерное видение: требования стейкхолдеров раскладываются как по уровням иерархии системы, так и по времени. В АКПУ для времени используется прообраз временной оси, отражающей стадии ЖЦ системы.

Фокусировка на **целевой функции продукта** является ещё одним общим знаменателем между АКПУ и системным оператором. Системный оператор побуждает рассматривать систему исходя из её предназначения (функции) на разных уровнях и этапах, чтобы не упустить сути проблемы при переходе от одной «экрана» к другой. АКПУ же прямо нацеливает анализ на удовлетворение главной цели продукта: определяемые ключевые параметры успешности фактически выражают, насколько эффективно система выполняет свою главную функцию в глазах стейкхолдера. Тем самым оба подхода взаимно усиливают друг друга: системный оператор задаёт структурированную рамку (время/уровни системы), а АКПУ наполняет её содержанием, связанным с целевой функцией и критериями успеха. Такой синтез обеспечивает методологическую строгость - анализ остаётся целостным и сфокусированным на главном, несмотря на обилие частных требований и условий.

1.8. Заключение

Развитие концепции системного оператора на основе работ Г.С. Альтшуллера [30], М. С. Рубина [45] показывает необходимость эволюционного расширения этой модели за счет введения дополнительной оси - жизненного цикла системы. В отдельных работах [59 - 62] в неявном виде такие попытки уже предпринимались, так в [59, 61, 62] рассматривается жизненный цикл системы или продукта, который является прообразом новой оси в системном операторе, о которой упоминал М.С. Рубин [45] как включение двух типов временных шкал (онтогенеза и филогенеза), что позволяет рассматривать как индивидуальную траекторию развития продукта, так и историю развития целого класса технических решений. Одновременно онтологическое оформление системного оператора фиксирует гибкость работы с экранной моделью, указывая на возможность разнообразных переходов между уровнями и временными срезами.

Однако классическая реализация системного оператора, ориентированная на техническую систему, оказывается недостаточной при переходе к продуктовой плоскости анализа - где на первый план выходят пользователь и рынок. Это противоречие между богатой методологией ТРИЗ и потребностями продуктового инновационного процесса служит мотивацией для разработки **расширенного системного оператора для продукта**. В следующих главах квалификационной работы будет представлена такая расширенная модель. Она нацелена на устранение отмеченных ограничений и на повышение

эффективности применения системного оператора в задачах продуктового анализа и проектирования.

Глава 2. Расширенный СО для продукта

Классический системный оператор ТРИЗ (многоэкранка) представляет собой универсальный инструмент анализа, который позволяет рассматривать объект на разных системных уровнях (подсистема, система, надсистема) и во времени (прошлое, настоящее, будущее). Однако для современного продуктового подхода, ориентированного на пользовательскую ценность и полный жизненный цикл продукта, такая схема оказывается недостаточной.

Специфические аспекты продукта - жизненный цикл со всеми его стадиями, множественность сценариев использования, роли различных пользователей и стейкхолдеров, а также структурные изменения продукта во времени - не находят полного отражения в многоэкранке. В связи с этим возникает необходимость расширить классическую модель системного оператора, дополнив её новыми измерениями, учитывающими особенности продукта.

В данной главе предлагается расширенная модель системного оператора, адаптированная для анализа продукта с учетом перечисленных факторов. В

2.1. Модель минимального продукта

2.1.1. Минимальная система и продукт

В ТРИЗ любой объект рассматривается как *система*, предназначенная для выполнения определенной полезной функции. Минимальная техническая система, согласно ТРИЗ, - это такая система, которая выполняет лишь одну полезную функцию, имея при этом минимально необходимый набор элементов (см раздел 1.1, Понятие «система» в ТРИЗ). Например, Г.С. Альтшуллер отмечал, что минимальная система может быть представлена двумя взаимодействующими компонентами («инструмент - изделие», как в веполе), совместно обеспечивающими выполнение единственной функции. В то же время понятие *продукта* шире воздействие одного элемента на другой.

В контексте маркетинга и продуктового подхода продукт определяется через пользу для пользователя: это *решение конкретной проблемы пользователя* или удовлетворение его потребности (см. раздел 1.2, Понятие «продукт»). Иными словами, потребитель приобретает не просто объект или технологию, а то ценностное решение, которое этот объект ему предоставляет. В классической метафоре Теодора Левитта [21]: «людям нужна не сама дрель, им нужна четвертьдюймовая дырка», - подчеркивается, что ценность продукта состоит в результате, получаемом пользователем, а не в самом по себе техническом функционировании устройства. Таким образом, *минимальный продукт* можно предварительно понимать как минимальную систему, рассматриваемую в контексте ее применения - то есть, это система,

нацеленная на решение одной пользовательской задачи или проблемы. В отличие от абстрактной системы, которая выполняет функцию вне зависимости от конкретного применения, продукт всегда «привязан» к сценарию использования и пользовательской ценности (см. раздел 1.2 Понятие «продукт»).

2.1.2. Определение минимального продукта

Принципиальное отличие между технической системой и продуктом заключается в учете контекста и потребностей пользователя. Система в понимании ТРИЗ - это прежде всего функциональный комплекс, обладающий внутренней структурой и выполняющий заданную функцию. Продукт же - это рыночное предложение, создающее полезный результат для пользователя лишь постольку, поскольку решает проблему пользователя или выполняет для него работу. Если система создается и оценивается по техническим критериям (функциональность, надежность, соответствие спецификациям), то продукт оценивается по тому, насколько полно и удобно он устраняет *внешнюю* проблему пользователя и вписывается в его сценарий применения. Таким образом,

минимальный продукт - система с одной функцией, предназначенная для одного сценария использования, то есть ориентированная на одно конкретное применение.

Данное определение подчеркивает, что мы рассматриваем не просто систему само по себе, а систему + контекст: **одна функция + один сценарий использования**. Такой фокус соответствует современным взглядам на создание продуктов, где главной мерой успешности [62] является решение конкретной задачи клиента лучше альтернативных средств .

2.1.3. Концепция Jobs to Be Done

Для описания минимального продукта удобно воспользоваться концепцией «работы, для которой нанимается продукт» (Jobs to Be Done, JTBD). В рамках этой концепции, предложенной К. Кристенсенем [16], продукт рассматривается через призму той задачи или «работы», которую пользователь пытается выполнить с его помощью. Фактически, *пользователь «нанимает» продукт для выполнения определенной работы* - если продукт справляется хорошо, он продолжает его использовать, если нет - «увольняет» и ищет замену. Кристенсен отмечал, что потребители покупают товары не ради самих по себе характеристик, а ради достижения желаемого результата: *«когда людям нужно выполнить работу, они по сути нанимают для этого продукт»*.

Следовательно, при проектировании продукта следует исходить из понимания той задачи, которую пользователь хочет решить, и конструировать решение строго под эту задачу [15, 16]. Использование подхода JTBD позволяет формализовать сценарий: описать, **в какой ситуации и какой результат** нужен пользователю. Для минимального продукта это означает, что мы явно фиксируем единичный сценарий применения, где продукт приносит пользу. Такой подход согласуется с определением выше: минимальный продукт решает *ровно одну* проблему в заданном контексте.

2.1.4. Представление продукта в системном операторе

В системном операторе (известного как многоэкранка в ТРИЗ) система рассматривается на разных уровнях и в разных контекстах. Применяя этот подход к продукту, можно формально разделить два ключевых уровня рассмотрения минимального продукта: уровень самой системы и уровень надсистемы. **На уровне системы** выступает техническое решение - носитель функции, сам продукт как устройство или технология, выполняющая свою основную функцию. **На уровне надсистемы** рассматривается сценарий использования - та самая пользовательская проблема, ради решения которой продукт создан. Такое представление иллюстрирует связь: продукт (система) получает смысл и ценность только внутри надсистемы (сценария использования). На рисунке показано представление минимального продукта в системном операторе.



Рисунок 2.1. Минимальный продукт в системном операторе

Минимальный продукт в рамках системного оператора - это связка “система-надсистема”, причем система обеспечивает функцию, а надсистема формирует требуемый сценарий и критерий успеха (решение проблемы).

Для наглядности рассмотрим примеры минимальных продуктов, демонстрирующих изложенные принципы.

Пример. Зубная щётка. Пользовательская проблема: необходимость быстро и эффективно удалить налет с поверхности зубов перед выходом из дома. Сценарий использования - утренняя гигиеническая процедура (чистка зубов перед работой), где ключевой «работой» является обеспечение свежести и чистоты полости рта за ограниченное время. **Функция продукта** - механическое удаление загрязнений (налета) с эмали зубов. В терминах системного оператора, система - сама зубная щетка как физический объект (инструмент с щетинками), который при взаимодействии с зубами выполняет очищающую функцию. Надсистема - утренняя гигиена и потребность поддерживать гигиену; сюда входит пользователь (человек, собирающийся на работу) и ситуация (утренние сборы, когда требуется быстро добиться чистоты зубов). **Минимальность продукта** в данном примере выражается в том, что щётка решает одну конкретную задачу (удаление налёта) в одном конкретном сценарии (ежедневная чистка зубов дома), не выходя за рамки этой функции и ситуации. При этом технически щётка является простой системой (ручка + щетина), достаточной для выполнения своей единственной функции.

Пример. Смартфон в качестве фотоаппарата. Смартфон выполняет много полезных функций, но нас в этом примере будет интересовать только одна из них. Пользовательская проблема: необходимость зафиксировать визуальное воспоминание о важном моменте. Сценарий использования (в формате JTBD): «Мне нужно быстро запечатлеть момент, когда мой ребенок делает первый шаг, чтобы потом поделиться этим с семьей». В данной ситуации **работа продукта** - превращение визуального образа в цифровую фотографию, доступную для хранения и передачи. На уровне системы продуктом выступает смартфон (точнее, его камера и встроенное ПО) - техническое средство, выполняющее функцию съемки и сохранения изображения. На уровне надсистемы - жизненная ситуация молодого родителя, желающего сохранить значимый момент развития ребёнка и оперативно поделиться им с близкими; сюда входит социальный контекст (семья), информационная среда (социальные сети и мессенджеры для обмена фото) и критерий успеха (момент зафиксирован вовремя и в приемлемом качестве). **Минимальный продукт** в этом примере - это смартфон-камера, решающий одну задачу: сделать фото первого шага ребенка в момент его совершения. Несмотря на то, что современный смартфон технически многофункционален, в данном рассматриваемом минимальном сценарии мы выделяем из него продуктивную сущность «смартфон как фотоаппарат». Именно в этой узкой роли он соответствует модели минимального продукта: выполняет одну функцию

(фотосъемка) в рамках одного пользовательского кейса (сохранение памятного момента для семьи). Если смартфон справляется с этой задачей (фото сделано быстро, удобно и с нужным качеством), пользователь «нанимает» его на эту работу и остаётся доволен решением своей проблемы; в противном случае он бы обратился к другому решению (например, отдельной фотокамере или попросил бы кого-то сделать снимок).

Пример. Сервис доставки пиццы. Пользовательская проблема: нехватка времени на приём пиццы в рабочее время (в результате возникает риск пропустить обед или снизить продуктивность работы). **Сценарий использования (JTBD):** «Я хочу быстро пообедать, не покидая рабочего места, чтобы не отвлекаться от дел». **Функция продукта:** доставка горячей еды в офис в течение ограниченного времени. **Продукт реализован** как веб-сервис, через который пользователь оформляет заказ на доставку обеда.

С точки зрения модели системного оператора данный сервис можно рассматривать на двух уровнях - системы и надсистемы. На уровне системы продукт представляет собой техническую систему доставки - интегрированный комплекс компонентов (веб-интерфейс для оформления заказа, база данных и алгоритмы обработки заявок, инфраструктура курьеров и кухни), совместно выполняющих функцию своевременной доставки еды пользователю. Этот веб-сервис выступает как технический носитель указанной функции, специально спроектированный для удовлетворения потребности пользователя в питании при дефиците времени [см. раздел 1.1).

На уровне надсистемы учитывается контекст использования продукта: офисная среда и распорядок рабочего дня пользователя. В этой надсистеме ключевая задача пользователя заключается в том, чтобы пообедать быстро и без отрыва от работы; эту задачу сервис доставки пиццы решает, органично вписываясь в рабочий график и инфраструктуру (офис, интернет, мобильные устройства). Иными словами, продукт предоставляет решение проблемы пользователя (нехватки времени на обед) в конкретном контексте (см. раздел 1.2). Согласно подходу Jobs to Be Done, потребитель фактически «нанимает» данный сервис для выполнения работы - организации быстрого обеда без лишних усилий [16]. Продукт будет успешен постольку, поскольку он надёжно и удобно выполняет эту работу и тем самым полностью удовлетворяет потребность пользователя.

2.1.5. Вывод по разделу 2.1

Минимальный продукт - это концепция, объединяющая минимальность системы (одна ключевая функция) и ценностную ориентацию (одно ключевое применение). Модель минимального продукта, описанная через системный оператор, подчеркивает необходимость рассматривать решение не изолированно, а во взаимосвязи с пользователем и сценарием использования. Такой подход обеспечивает фокусирование разработки на сути продукта - на той единственной «работе», ради которой его будут использовать.

Далее (в разделе 2.2) будет модель продукта в случаях множественных сценариев использования продукта.

2.2. Модель продукта (не минимального)

2.2.1. Множественность сценариев использования

Современный продукт, как правило, эксплуатируется не в одном-единственном сценарии, а в нескольких и даже многих. Даже простой предмет, например зубная щетка, имеет более одного сценария использования. Основной сценарий - непосредственное применение для очистки зубов (удаление налета с эмали перед выходом на работу утром). Однако помимо этого, можно выделить и второй сценарий - хранение щетки до следующего применения, когда продукт находится в пассивном состоянии, ожидая очередного использования. В системной инженерии второй сценарий может не рассматриваться вообще, но в рамках маркетинга и менеджмента продукта второй сценарий является источником для создания пользы для потребителя.

Современные теории продуктов подчеркивают, что ценность изделия определяется именно пользой в конкретных сценариях использования [16]. Иными словами, пользователи «нанимают» продукт для выполнения определенных работ, и один и тот же продукт может использоваться для выполнения разных работ в разных условиях [16], (см. также раздел 1.2, Понятие "продукт").

Для сложных изделий, таких как смартфон, многообразие сценариев использования проявляется еще сильнее. **Мультифункциональность** сложного продукта означает, что он предназначен для решения различных проблем пользователя и поэтому применяется в различных обстоятельствах. Например, смартфон может выступать как средство связи, камера, навигатор, проигрыватель медиа и т.д. - каждый из этих вариантов использования представляет отдельный сценарий, хотя все они реализуются одним устройством. Но даже если продукт выполняет по сути одну и ту же главную

функцию, он все равно может использоваться в разных условиях. К примеру, фонарь обеспечивает освещение (единая функция) как в походе, так и дома при отключении электричества - контекст применения отличается, хотя выполняемая функция схожа. **Следовательно, один продукт способен использоваться в нескольких сценариях использования в зависимости от ситуации.**

Ранее, в разделе 2.1, была рассмотрена модель минимального продукта (см. раздел 2.1, Модель минимального продукта). В минимальной модели предполагалось, что система (продукт) действует в рамках единственного фиксированного сценария, соответствующего одной надсистеме. В частности, для минимального продукта устанавливалось отношение между системой и надсистемой, охватывающей один сценарий использования. Однако в общем случае продукт связан с несколькими надсистемами - по одной на каждый сценарий его применения. В терминах системного подхода это означает, что **продукт как система может быть частью нескольких различных надсистем, каждая из которых отражает определенный сценарий использования.**

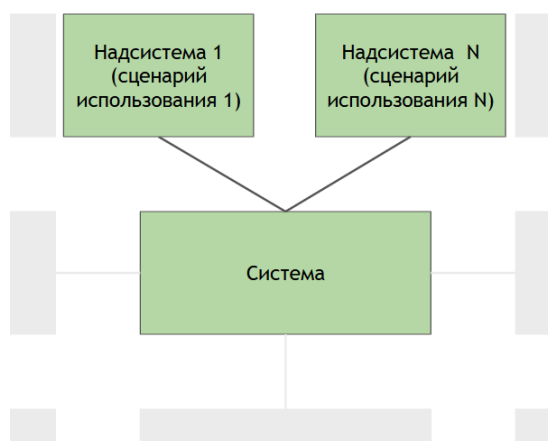


Рисунок 2.2. Продукт - это система и много надсистем

Еще Г.С. Альтшуллер подчеркивал необходимость рассматривать техническую систему в надсистеме, частью которой она является [30]. Соответственно, наличие у продукта множества сценариев использования фактически означает, что для его полного описания требуется учесть несколько надсистем (по числу сценариев использования продукта).

2.2.2. Микро-стадия использования

Классический системный оператор (многоэкранка) обычно фиксирует рассмотрение системы в одной надсистеме на определенном этапе (См. раздел 1.5, Системный оператор в классической ТРИЗ). Для упорядочения связи между системой и несколькими надсистемами (т.е. множеством сценариев

использования) в рамках системного оператора продукта нами **вводится понятие "микро-стадия использования"**.

Под микро-стадией использования продукта подразумевается ограниченный период или эпизод на этапе эксплуатации, в течение которого реализуется определенный отдельный сценарий использования.

Иными словами, каждая микро-стадия использования соответствует конкретному сценарию использования продукта в определенный промежуток времени. В дальнейшем будет показано, что введенная микро-стадия является составной частью общей стадии использования в жизненном цикле продукта. Следует отметить, что стадия эксплуатации (использования) признается одной из ключевых фаз жизненного цикла изделия в инженерном понимании [8] (см. также раздел 1.4, Понятие "жизненный цикл продукта") - именно на этой стадии продукт выполняет свою функцию и приносит пользу потребителю. В данной работе совокупность таких полезных эпизодов эксплуатации продукта разлагается на микро-стадии.

Для каждого выявленного сценария использования определяется своя микро-стадия использования продукта.

Таким образом, **продукт можно рассматривать как совокупность микро-стадий системы, связанных со сценариями использования, соответствующими разным ситуациям, в которых продукт приносит пользу (в рамках сценария использования).**

Эти микро-стадии могут следовать друг за другом или чередоваться в хронологическом порядке в пределах общей стадии эксплуатации. Поскольку разные сценарии реализуются не одновременно, а в разные моменты времени, соответствующие им микро-стадии происходят в разные периоды. Совокупность всех определенных микро-стадий фактически образует полную стадию использования продукта в рамках его жизненного цикла. Другими словами, последовательность микро-стадий (возможно, с паузами между ними) охватывает весь период, пока продукт находится в эксплуатации.

Если же в какие-то периоды времени в пределах стадии использования не осуществляется ни один из предусмотренных сценариев (то есть ни одна микро-стадия не активна), то в эти промежутки продукт не производит никакой полезной функции для пользователя. На примере зубной щетки это означает, что после завершения микро-стадии чистки зубов наступает микро-стадия хранения (пассивное нахождение щетки на полке до следующего сеанса чистки). Если же щетка забыта и вовсе не используется, то все время нахождения без дела не приносит пользы до тех пор, пока снова не наступит момент ее применения.

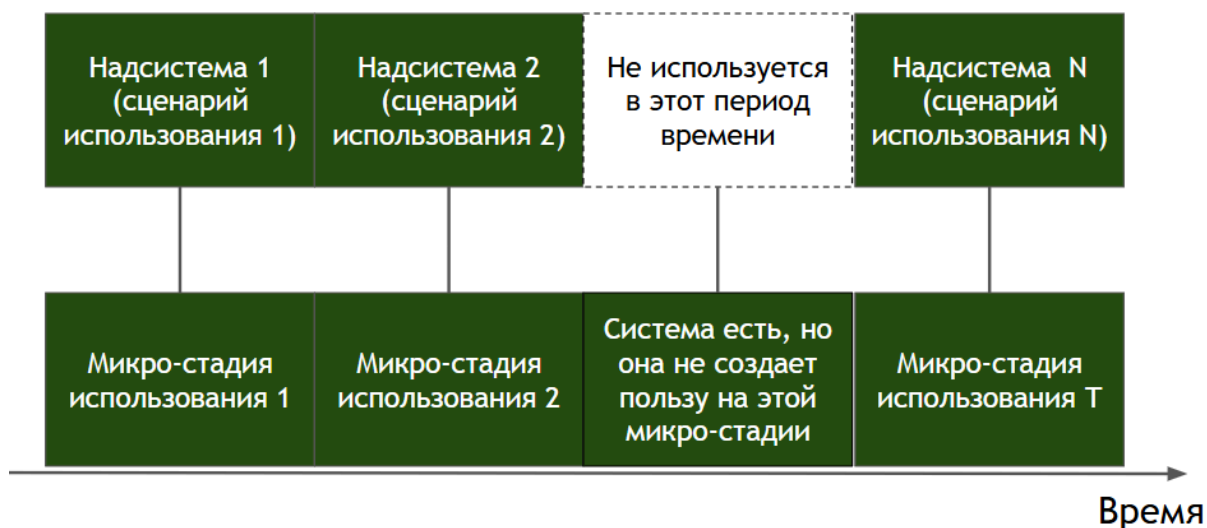


Рисунок 2.3. Микро-стадии и сценарии использования продукта

Пример. В сервисе доставки пиццы можно выделить следующие активные и пассивные микро-стадии использования:

1) *Выбор блюда на сайте: Пользователь открывает веб-сайт доставки пиццы и просматривает меню. Он выбирает подходящую пиццу (например, с учетом предпочтений по ингредиентам, размеру) и дополнительные позиции, если нужно (напитки, салат и т.п.). На этой стадии важны удобная навигация и фильтры меню - они позволяют быстро найти желаемое блюдо. Пользователь активно оценивает варианты и принимает решение, какую пиццу заказать.*

2) *Оформление заказа: Определившись с выбором, пользователь переходит к оформлению заказа. Он вводит адрес доставки (или выбирает заранее сохраненный офисный адрес), указывает способ оплаты (онлайн оплата картой или выбор наличного расчета курьеру) и при необходимости оставляет комментарии к заказу (например, уточняет вход в здание офиса, пожелания по приготовлению). Затем пользователь подтверждает заказ. Эта активная стадия завершается, когда система принимает заказ и отображает подтверждение с ориентировочным временем доставки.*

3) *Ожидание доставки: После оформления заказа наступает пауза, когда пользователь ждет прибытия пиццы. Эта стадия пассивна - пользователь переключается на свои рабочие дела, однако внутренне отсчитывает время. В этот период важны точность прогноза времени доставки и прозрачность сервиса: пользователь может получать уведомления о статусе заказа (например, «заказ готовится», «курьер выехал») - это успокаивает и позволяет планировать свое время. В контексте пассивной*

стадии ожидания должно быть минимальным и предсказуемым, чтобы не вызывать у пользователя беспокойства или раздражения.

4) *Получение пиццы:* После прибытия курьера пользователь активно возобновляет взаимодействие с сервисом - он выходит к точке доставки (например, ресепшен офиса или вход в здание), встречается с курьером и получает заказ. Здесь происходит краткое взаимодействие: проверка заказа (соответствует ли полученное блюдо заказанному), расчет с курьером (если не была произведена онлайн-оплата) и, возможно, обмен любезностями или подпись в подтверждение получения. Эта стадия отмечает материальную реализацию ценности продукта - пользователь физически получает пиццу.

5) *Приём пищи:* Получив пиццу, пользователь приступает к её употреблению - фактически реализуется главная полезная функция продукта. Он открывает коробку, разогревает еду, если нужно (некоторые офисы оснащены микроволновками), и ест свой обед прямо на рабочем месте или в зоне отдыха офиса. Эта стадия активна по отношению к пользованию результатом продукта: пользователь взаимодействует уже не с веб-сайтом, а непосредственно с полученной пищей. В контексте модели продукта прием пищи тоже рассматривается, поскольку это конечная цель использования сервиса - удовлетворение потребности в еде.

6) *Утилизация упаковки:* После окончания еды пользователь должен выбросить пустую коробку из-под пиццы и другую использованную упаковку. Хотя это действие совершается пользователем (активно), с точки зрения главной функции продукта оно является завершающим пассивным этапом после основного использования. Продукт как система уже выполнил свою роль, и дальнейшее взаимодействие - лишь побочный процесс. В академической модели продукта утилизация часто рассматривается как заключительная стадия пользовательского опыта: важно, чтобы упаковка была удобна в утилизации (например, коробку легко сложить и выбросить, она не пачкает окружающее). Эта стадия влияет на общее впечатление: если избавиться от упаковки просто, пользователь быстрее возвращается к работе, завершая цикл использования продукта на позитивной ноте.

После прохождения всех указанных микро-стадий цикл использования сервиса завершается. Пользователь насытился и вернулся к своим обязанностям, а веб-сервис выполнил свою «работу» - избавил пользователя от голода быстро и удобно. Анализ всех стадий - от первого контакта с сайтом до выброса коробки - позволяет целостно оценить пользовательский опыт и выявить возможные улучшения продукта. Например, если ожидание (пассивная стадия) слишком длительное, это указывает на необходимость совершенствовать логистику; если на стадии

выбора блюда возникают затруднения, стоит улучшить интерфейс сайта и рекомендательную систему. Такой подробный разбор соответствует модели продукта, где продукт представляется не статичным объектом, а системой в использовании - в определённом контексте, с определённой целью пользователя и последовательностью этапов взаимодействия.

Если в основе продукта (например, сервиса) находится процесс, то микро-стадии его использования естественным образом совпадают с операциями этого процесса (см. пример сервиса доставки пиццы). Это объясняется тем, что процесс, в отличие от статической системы, обладает временной структурой (его операции последовательно разворачиваются во времени) (см. раздел 1.1.2). Каждая операция процесса может рассматриваться как отдельная микро-стадия, в рамках которой реализуется определенная часть полезной функции продукта. Это справедливо не только для сервисов (например, заказ, доставка, оплата), но, как мы покажем далее, для производственных и бизнес-процессов (резка, сборка, упаковка). Таким образом, в процессах микро-стадии использования выражены более явно и фактически совпадают с функциональной декомпозицией процесса во времени.

2.2.3. Потребители и их роли на микро-стадиях

Продукт может использоваться через последовательность микро-стадий - небольших этапов взаимодействия пользователя с продуктом. Для каждой такой микро-стадии должен быть явно определен соответствующий сценарий использования продукта. В сценарии описывается, как именно применяется продукт на данном этапе, и **обязательно присутствует потребитель** (пользователь), взаимодействующий с системой. При этом, в зависимости от ситуации, потребителем может быть как один и тот же человек на всех стадиях, так и разные люди на разных микро-стадиях взаимодействия.

Даже если во всех сценариях использования продукта участвует один и тот же человек (один потребитель), **его роль может меняться от сценария к сценарию**. Иными словами, один пользователь в разных контекстах применения продукта решает различные задачи. В одном случае он может выступать инициатором действия, в другом - получателем результата, и т.д. Таким образом, при моделировании продукта важно учитывать не только сами микро-стадии, но и роли потребителя на каждой стадии.

Пример. Сервис доставки пиццы: в разных микро-стадиях сервиса могут участвовать различные люди. Например, один сотрудник (коллега) оформляет заказ пиццы, секретарь получает пиццу у курьера, а директор вместе с коллегой едят пиццу. Здесь каждая стадия (заказ, получение, потребление) представляет свой сценарий использования, причем на

каждой стадии свой **участник-потребитель** с определённой ролью (заказчик, получатель, конечный потребитель).

Пример. Смартфон: во всех основных сценариях использования смартфона участвует один человек - его владелец. Однако на разных стадиях использования он выполняет разные роли. Например, при создании фотографий владелец действует в роли **фотографа**, при совершении телефонного звонка - в роли **абонента (звонящего)**, а при публикации поста в блог - в роли **блогера**. Несмотря на то, что это один и тот же пользователь, каждый сценарий использования смартфона заставляет его выступать в новой роли, соответствующей текущей задаче.

Таким образом, для каждой микро-стадии продукта необходимо четко определять сценарий использования с указанием **конкретного потребителя**. Это позволяет понять, **кто** и **как** взаимодействует с продуктом на каждом шаге, и учитывает, что разные стадии могут подразумевать участие разных людей или разные роли одного пользователя. Такой подход обеспечивает целостное описание **модели продукта** с точки зрения реального применения: все ключевые моменты использования продукта связаны с соответствующими сценариями и участниками, что помогает полно и правильно отразить ценность продукта для потребителей на каждом этапе.

На следующем рисунке представлена связь между микро-стадиями и различными пользователями и их ролями.

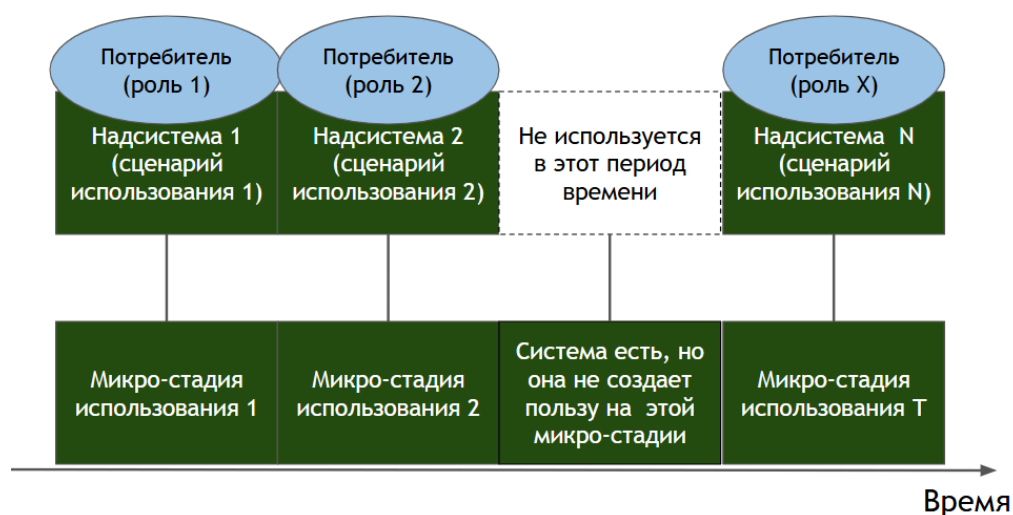


Рисунок 2.4. Связь между микро-стадиями и пользователями и их ролями

Микро-стадии продукта демонстрируют механизм появления разных стейкхолдеров у продукта. В этом разделе показано, как разные стейкхолдеры могут появляться на микро-стадиях использования продукта. В дальнейшем будет показано, как стейкхолдеры могут появляться на других стадиях жизненного цикла продукта.

2.3. Модель жизненного цикла продукта

В данном разделе концепция микро-стадий, введенная в разделе 2.2 для детализации стадии использования, распространяется на весь жизненный цикл продукта. Для этого вводится более общее понятие - **сценарий взаимодействия**, - позволяющее описывать эпизоды взаимодействия стейкхолдеров с системой на любых этапах жизни продукта, а не только на стадии использования.

2.3.1. Понятие сценария взаимодействия

Сценарий взаимодействия - это описание конкретного эпизода взаимодействия стейкхолдера (заинтересованного участника) с системой в определенном контексте для достижения некоторой цели. Данное понятие обобщает используемый ранее термин «сценарий использования» (вариант использования), который обычно относится к взаимодействию конечного пользователя с продуктом на фазе использования (эксплуатации) продукта. В отличие от этого частного случая, сценарий взаимодействия охватывает **любой** случай, когда какое-либо заинтересованное лицо вступает во взаимодействие с системой на той или иной стадии её жизненного цикла. Формально сценарий взаимодействия включает следующие компоненты:

- **Контекст взаимодействия** - условия и среда, в которых происходит эпизод (этап процесса, стадия жизненного цикла, внешние обстоятельства).
- **Участник взаимодействия (стейкхолдер)** - конкретное лицо или организация, выполняющая роль актора в данном эпизоде (например, конечный пользователь, инженер-разработчик, оператор производства, сервисный техник и т.д.).
- **Тип деятельности и цель** - характер выполняемой участником работы и ее целевая направленность. Сценарий фиксирует, *что делает* стейкхолдер и *зачем* (например, тестирует прототип для поиска дефектов, собирает изделие для ввода в эксплуатацию, проводит техническое обслуживание для поддержания работоспособности и пр.).
- **Воздействие на систему** - способ, которым в рамках данного эпизода участник задействует продукт либо влияет на его состояние. Иными словами, указывается, **как именно** система применяется, изменяется или потребляется в рамках данного сценария.

Таким образом, сценарий взаимодействия определяет, **кто, где, когда и с какой целью** взаимодействует с продуктом, а также каким образом при этом используется сама система или оказывается воздействие на нее. Классический

сценарий использования продукта (например, эпизод работы пользователя с устройством в период эксплуатации) является лишь частным случаем сценария взаимодействия - в нём стейкхолдером выступает конечный потребитель, а стадией жизненного цикла является эксплуатация. Однако более широкая концепция сценария взаимодействия позволяет учитывать и иные эпизоды: взаимодействие инженера с системой при ее испытаниях, участие оператора на этапе производства, действия техника при обслуживании, процессы демонтажа при утилизации и т.д.

Например, на стадии разработки можно выделить сценарий **тестирования прототипа инженером**; на стадии производства - сценарий **сборки узла оператором конвейера**; на стадии сопровождения - сценарий **планового технического обслуживания**, выполняемого сервисным инженером; на стадии вывода из эксплуатации - сценарий **демонтажа системы** специализированной бригадой. Во всех этих случаях имеет место взаимодействие стейкхолдера с системой, хотя ни один из эпизодов не является «классическим» пользовательским применением готового продукта.

Следовательно, любое значимое взаимодействие со стороны заинтересованного лица на любой фазе жизненного цикла может быть описано соответствующим сценарием взаимодействия.

Совокупность всех релевантных сценариев взаимодействия охватывает полный жизненный цикл продукта. Последовательность таких сценариев фактически образует цепочки микро-стадий внутри каждой крупной стадии. Другими словами, каждую фазу ЖЦ продукта можно детализировать как набор эпизодов взаимодействия, происходящих последовательно (или итеративно) во времени. Ниже рассмотрено, как идея микро-стадий и сценариев взаимодействия применяется ко всем этапам жизненного цикла.

2.3.2. Микро-стадии на разных стадиях ЖЦ продукта

Логично распространить идею микро-стадий (введенных ранее для стадии эксплуатации) и на остальные стадии жизненного цикла системы/продукта. Каждая крупная стадия ЖЦ, как правило, не является монолитной - она включает в себя ряд шагов, процессов или состояний, последовательно сменяющих друг друга.

Например, стадия разработки обычно состоит из последовательности этапов: концептуальный дизайн, техническое проектирование, изготовление прототипов, тестирование и т.д. - фактически внутри неё можно выделить микро-стадии разработки, соответствующие перечисленным процессам (каждый из этих этапов представляет собой отдельный сценарий взаимодействия команды разработки с системой). Методологии проектного

менеджмента (например, модель Stage-Gate) формализуют такое дробление, устанавливая контрольные вехи между шагами разработки и принимая решения о переходе к следующему этапу после оценки результатов предыдущего. Аналогично, на стадии производства изделия можно увидеть смену микро-стадий: подготовка производства, собственно изготовление, сборка, контроль качества, выпуск пилотной партии и пр. - каждое из этих событий представляет отдельный эпизод взаимодействия в рамках общей фазы производства. Стадия поддержки/обслуживания также состоит из повторяющихся циклов работ: плановое техобслуживание, обновление программного обеспечения, ремонт неисправностей, предоставление запчастей, техническая поддержка пользователей и т.п. Фактически каждое сервисное вмешательство или очередной выпуск обновления можно трактовать как отдельную микро-стадию поддержки (отдельный сценарий сервисного взаимодействия), после которой продукт возвращается в штатный режим эксплуатации. Даже заключительная стадия вывода из эксплуатации и утилизации распадается на ряд шагов: отключение и демонтаж системы, переработка и удаление отходов, архивирование данных, анализ накопленного опыта эксплуатации и т.д. - таким образом, и эта фаза состоит из последовательности эпизодов, завершающих жизненный цикл системы.

Таким образом, каждая стадия жизненного цикла может быть представлена как последовательность более мелких эпизодов - «микро-стадий», соответствующих конкретным работам или событиям (т.е. отдельным сценариям взаимодействия, характерным для данной фазы).

Следует подчеркнуть, что введение микро-стадий для всех этапов ЖЦ направлено не на искусственное усложнение модели, а на более точное отражение реального жизненного процесса продукта. В традиционных схемах жизненного цикла границы между стадиями часто довольно широкие и условные. Разбиение же на микро-стадии позволяет увидеть, как именно протекает переход продукта из одного состояния в другое.

Например, вместо общей абстрактной «эксплуатации» мы описываем череду конкретных сценариев взаимодействия продукта с пользователем; вместо единого блока «разработка» - последовательность этапов создания и проверки опытных образцов; вместо агрегированного «обслуживания» - набор регулярных технических операций, через которые продукт поддерживается в рабочем состоянии.

Такой подход создаёт более «дискретную» модель времени жизни системы, которую при необходимости можно увязать с календарным графиком проекта или с циклами потребления на рынке. Кроме того, детализация стадий облегчает выявление узких мест и проблем на каждом отдельном отрезке жизненного пути.

Например, можно отдельно проанализировать риски, возникающие при переходе между микро-стадиями (переход от создания прототипа к тестированию, от производства пилотной партии к массовому выпуску, от активного использования к фазе простоя и т.д.), и подготовить меры по их сглаживанию. В целом, представление жизненного цикла как последовательности микро-стадий создает основу для более гибкого и точного моделирования поведения продукта во времени.

2.3.3. Перенос микро-стадий между разными стадиями ЖЦ

В процессе эволюции продукта возможен перенос отдельных микро-стадий (конкретных сценариев взаимодействия) между стадиями его жизненного цикла.

Пример. Так, в мебельной отрасли ярким примером стала **самостоятельная сборка**: эта микро-стадия, по сути представляющая сценарий сборки изделия, перемещена со стадии производства на стадию эксплуатации, то есть передана самому потребителю. Розничная модель ИКЕА основана на продаже мебели в виде комплекта деталей, которые покупатель затем собирает собственными руками.

В результате производитель снижает издержки на сборку и логистику (отпадает необходимость крупноузловой сборки на заводе и доставки габаритных изделий) и может предложить более низкую цену [31]. Для потребителя же создаётся дополнительная ценность за счет вовлеченности: помимо экономии средств, пользователь получает опыт DIY («сделай сам») и эмоциональную удовлетворенность от результата своего труда. Исследования показывают, что покупатели, лично участвующие в создании продукта (например, сборке мебели), начинают субъективно ценить его выше, чем если бы получили готовое изделие [32].

Этот феномен, известный как «эффект ИКЕА», демонстрирует, как перенос сценария сборки на стадию использования (т.е. выполнение сборки самим пользователем) повышает ценность продукта в глазах потребителя при одновременном снижении затрат производителя.

Аналогичные сдвиги наблюдаются в цифровой сфере.

Пример. В современном программном обеспечении всё чаще практикуется **открытое бета-тестирование**, когда этап тестирования продукта, ранее относившийся к стадии разработки, частично переносится на стадию эксплуатации и выполняется самими конечными пользователями. Компании выпускают предварительные версии (бета-версии) программ и привлекают конечных пользователей в роли тестировщиков, собирая их отзывы и данные об ошибках.

Таким образом, микро-стадия верификации и отладки продукта смещается от компании во внешнюю среду использования - пользователи фактически участвуют в сценарии тестирования, выступая в роли внешней группы QA. Это позволяет ускорить вывод продукта на рынок и улучшить качество за счёт масштабного покрытия разнообразных сценариев взаимодействия пользователей с продуктом, недостижимого усилиями внутренней команды [33].

Фактически продукт начинает «жить» у потребителей ещё до формального окончания разработки - концепция «вечной беты» подразумевает непрерывное обновление и улучшение уже эксплуатируемого решения [34]. Пользователь при этом становится соучастником жизненного цикла продукта, влияя на его доработку, что повышает лояльность аудитории, хотя и требует от неё определённых усилий (например, сообщать о неполадках и мириться с не полностью сформированной функциональностью на ранних этапах).

Временные границы между стадиями разработки и использования размываются: продукт дольше находится в итеративном улучшении, а его жизненный цикл приобретает более гибкое, эволюционное поведение во времени.

В секторе услуг перенос микро-стадий также стал драйвером преобразования ценности.

Пример. В индустрии туризма стадия планирования и бронирования поездки, ранее обеспечиваемая туроператорами и агентствами, переместилась непосредственно к потребителю на этап пользования сервисом. С распространением онлайн-платформ путешественники сами подбирают рейсы, отели и другие составляющие тура через интернет, выполняя работу, которую раньше делал агент. Такая микро-стадия планирования услуги фактически «переехала» со стороны предложения (этап подготовки услуги профессионалами) на сторону спроса (этап, осуществляемый самим клиентом).

Это привело к существенным изменениям в отрасли: пользователи получили более широкий выбор и контроль над формированием своего тура, зачастую снижаются транзакционные издержки (нет комиссий агентств), однако возрастает нагрузка на самого клиента, которому необходимо разбираться в предложениях. Ценность туристского продукта при этом повышается за счёт высокой персонализации и гибкости - путешественник может сконфигурировать маршрут под свои уникальные предпочтения. Роль потребителя трансформируется в роль со-продюсера услуги [27].

Масштаб этих изменений иллюстрирует статистика: с 2000 по 2021 годы число рабочих мест традиционных турагентов сократилось примерно на 70%, поскольку большинство клиентов предпочли самостоятельное онлайн-бронирование [35]. Одновременно онлайн-сервисы уполномочили потребителей напрямую осуществлять те задачи, что раньше выполняли посредники: клиент самостоятельно находит и резервирует оптимальные варианты поездки через удобный интерфейс, становясь активным участником жизненного цикла туристского продукта [31].

Таким образом, граница между стадией предоставления услуги и стадией ее потребления смещается, а сам продукт-услуга эволюционирует - он предлагает больше возможностей в режиме самообслуживания, меняя модель поведения системы «продукт-потребитель» на протяжении жизненного цикла.

Общие последствия подобных перестановок микро-стадий состоят в перераспределении активности и ценности между производителем и пользователем. Передача части работ потребителю позволяет производителю сократить издержки и повысить эффективность, тогда как потребитель получает более дешёвый, персонализированный или оперативный продукт, но взамен берёт на себя новые функции в его жизненном цикле. В результате возрастает вовлеченность пользователя и его удовлетворенность (через чувство соучастия и контроля), а сам продукт приобретает способность более гибко адаптироваться к потребностям рынка по мере использования. Перенос микро-стадий между фазами жизненного цикла, будучи продуманным стратегически, приводит к появлению новых источников ценности и изменяет траекторию развития продукта во времени, подтверждая системный характер эволюции продуктовых сервисов [9].

2.3.4. Ось жизненного цикла в системном операторе продукта

Представление жизненного цикла продукта в виде цепочки микро-стадий (сценариев взаимодействия) открывает возможность ввести в системный оператор отдельное измерение (ось), отвечающее за динамику продукта на протяжении его жизненного цикла.

Напомним, классический системный оператор ТРИЗ (многоэкранка) Альтшуллера имеет две явные координаты анализа: иерархию системы (подсистема - система - надсистема) и время (прошрое - настоящее - будущее). При рассмотрении конкретного продукта это означает, что традиционно мы оцениваем систему на каком-то фиксированном этапе её жизни (как правило, подразумевается стадия эксплуатации) и можем мысленно смещаться либо во времени (рассматривая её предшественников и возможное будущее), либо по иерархии (спускаясь в подсистемы или поднимаясь к надсистеме).

Как было показано в Главе 1, потребность во введении оси жизненного цикла в системный оператор назрела. Так, Рубин [45] предложил разделять онтогенез и филогенез системы, в ряде инструментов ТРИЗ [56, 61, 62] явно присутствует анализ жизненного цикла системы. Однако явного учёта стадий жизненного цикла в классической схеме системного оператора нет.

Например, типичные вопросы системной инженерии - учёт требований эксплуатации при проектировании, обеспеченность обслуживания или утилизируемость продукта - требуют рассмотрения системы на разных стадиях ее ЖЦ, но классический оператор явно не настраивает мышление на смену фаз разработки, производства, эксплуатации и т.д..

В целях более полного охвата жизненного цикла продукта предлагается ввести ось жизненного цикла в структуру системного оператора. Это означает, что модель продукта рассматривается не в одной условной «середине» его жизни, а прослеживается через последовательность этапов от начала до конца.

На практике такая интеграция реализуется как добавление нового измерения к матрице анализа. Если вообразить классический системный оператор в виде плоской матрицы, то ось ЖЦ превращает его в объёмную структуру (условно - куб мышления или «стопку» матриц), каждая «прослойка» которой соответствует определённой стадии или даже микро-стадии жизненного цикла.

К примеру, можно рассматривать три ключевых уровня иерархии (подсистема, система, надсистема) на стадии концепции, затем те же уровни на стадии разработки, и так далее - через производство, эксплуатацию, обслуживание - вплоть до вывода из эксплуатации. В каждом таком сечении будут свои особенности: на стадии концепции - в надсистеме будет сценарий взаимодействия инженера по созданию абстрактных моделей и исходных требований; на стадии производства - в надсистеме будут различные сценарии взаимодействия с компонентами системы, их сборки; на стадии эксплуатации - в надсистеме будут сценарии использования готового продукта пользователем и т.п. Получается своеобразная 3-мерная схема: две оси заимствованы из классической ТРИЗ (время-эволюция и системная иерархия), а третья - добавлена для прохождения по этапам жизненного цикла. Схематично идея представлена на следующем рисунке (рис. 2.5).

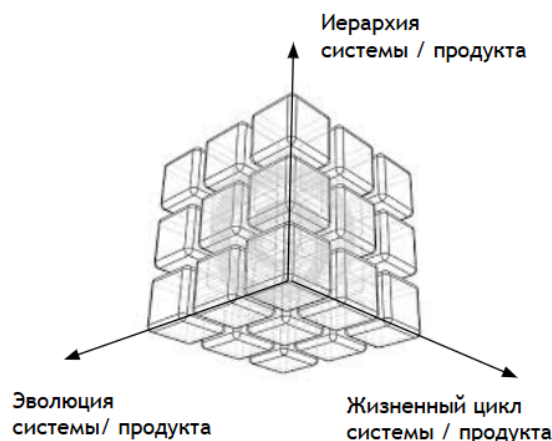


Рисунок 2.5. 3-осевая модель системного оператора для продукта

При решении прикладной задачи мысленное перемещение вдоль оси жизненного цикла продукта означает смену контекста на другую микро-стадию продукта (иной эпизод взаимодействия продукта со стейкхолдером), что помогает выявить новые стороны проблемы или решения. Формируя такой многомерный «куб» знаний о продукте, разработчик получает более объемное видение ситуации.

Конечно, практически заполнить сразу все ячейки этой схемы очень сложно да и не требуется - в каждом конкретном проекте актуальными будут отдельные сочетания измерений. Однако само наличие оси жизненного цикла в системном операторе дисциплинирует эксперта не ограничиваться одной фазой, а проверять решения на разных этапах.

Пример. В качестве иллюстрации применения оси жизненного цикла продукта рассмотрим проект компании, производящей автоматические кофемашины и пакетированные напитки (кофе, чай, капучино и др.). Задача состояла в сборе данных о потреблении напитков, однако изначально кофемашины не были оборудованы средствами телеметрии (пользователь вручную вставляет одноразовый пакетик порошка, и автомат не фиксирует вид и факт приготовления напитка). Первоначальное решение предполагало модернизацию всего парка аппаратов (десятки тысяч кофемашин) - их оснащение камерами и модемами для слежения за использованием пакетиков, что делало проект чрезвычайно сложным и затратным по ресурсам.

В ходе анализа проблемы с помощью оси жизненного цикла была построена модель жизненного цикла пакетика напитка как продукта, включающая последовательность микро-стадий. Были выделены следующие стадии и ключевые микро-стадии жизненного цикла пакетика:

1. **Производство.** Изготовление пакетированных порций напитка на заводе.
2. **Доставка на склад.** Транспортировка партии пакетиков на центральный склад компании.
3. **Доставка к автомату.** Поставка пакетиков в точку эксплуатации, пополнение запасов конкретной кофемашины (например, торгового автомата в офисе).
4. **Загрузка в мерчендайзер.** Заправка пакетиков в специальное внешнее хранилище возле машины (мерчендайзер - ящик для выдачи порций). Эта операция обычно выполняется сервисным сотрудником.
5. **Выбор пользователем.** Момент, когда конечный пользователь достаёт выбранный пакетик из мерчендайзера перед приготовлением.
6. **Загрузка в машину.** Непосредственное вставление пакетика пользователем в кофемашину для приготовления напитка.
7. **Приготовление напитка.** Автоматическое растворение содержимого пакетика водой и выдача готового напитка (происходит внутри машины после шага 6).
8. **Утилизация пакетика.** Выброс использованного пакетика в урну после приготовления напитка (действие выполняется пользователем или обслуживающим персоналом уборки).

Проанализировав эту последовательность, разработчики определили ключевую микро-стадию, на которой целесообразно собирать информацию о потреблении, - этап «**выбор пользователем**». Вместо модернизации самих кофемашин фокус проектирования был смещен именно на эту микро-стадию жизненного цикла продукта. Иными словами, задача была переосмыслена через выбор оптимального сценария сбора данных. В результате был разработан новый узел - мерчендайзер с датчиком, способный фиксировать момент, когда пользователь берет пакетик (то есть фактический выбор напитка), и передавать эти данные на сервер.

Таким образом, использование модели жизненного цикла продукта в рамках системного оператора позволило сместить фокус задачи на оптимальный эпизод сбора данных и существенно упростить инженерную проблему, устранив необходимость доработки самой кофемашины. Итогом стал более экономичный и простой в реализации проект: вместо дорогостоящей модернизации оборудования достаточно внедрить сравнительно недорогой модуль для фиксации факта выбора пакетиков, что радикально снизило стоимость решения.

Введение оси ЖЦ в системный оператор продукта представляется целесообразным по ряду причин.

Во-первых, это обеспечивает полноту системного мышления при работе с продуктом. Специалист получает методический инструмент, заставляющий его последовательно «проверить» продукт на всех этапах - от замысла до окончания эксплуатации. Это снижает риск пропустить важные требования или ограничения, проявляющиеся только на отдельных стадиях.

Например, решение, отлично работающее на этапе опытного прототипа, может оказаться нежизнеспособным в массовом производстве из-за технологических допусков или стоимости материалов; или же нововведение, привлекательное с точки зрения маркетинга, может затруднить обслуживание на этапе эксплуатации. Используя ось ЖЦ продукта, команда разработки систематически анализирует последствия решений на последующие фазы, тем самым приближаясь к принципу «учет жизненного цикла в дизайне» (например, концепции Design for X, охватывающие пригодность к производству - manufacturability, пригодность к поддержке - maintainability, пригодность к утилизации - recyclability и др.).

Во-вторых, ось ЖЦ способствует лучшей координации межфункциональных действий при разработке и сопровождении продукта. Жизненный цикл продукта затрагивает разные подразделения организации: инженеров (разработка, производство, поддержка), маркетологов (вывод на рынок, продвижение, снятие с продаж), сервисные службы (обслуживание, работа с возвратами) и т.д. Каждая функция традиционно сосредоточена на «своём» этапе, что чревато разрывами во взглядах. Инструмент в виде системного оператора с осью ЖЦ позволяет всем участникам проекта работать с единой схематической моделью, где отражены их взаимосвязи.

Например, маркетинговые планы по расширению продуктовой линейки на этапе роста можно совместить с инженерными планами модульной архитектуры, облегчающей последующую модернизацию; требования сервисных инженеров по доступности узлов для ремонта учитываются еще на стадии конструирования и т.д.

Таким образом, системный оператор расширяется из чисто изобретательского метода (как в классической ТРИЗ) в средство стратегического продуктового планирования, отображающее жизненный цикл. Это особенно актуально в современной практике управления инновациями, где широко применяются системы Product Lifecycle Management (PLM) - они тоже основаны на идее целостного видения продукта от колыбели до могилы. Предлагаемая модель фактически интегрирует концепцию PLM в мышление разработки: ось жизненного цикла служит напоминанием, что проектные решения должны оцениваться с точки зрения всей дальнейшей «судьбы» продукта.

В-третьих, добавление оси ЖЦ обогащает аналитические возможности системного оператора и позволяет по-новому формулировать изобретательские задачи. Появляется возможность искать решения через «перенос идей» между стадиями: например, задаться вопросом, нет ли аналогов рассматриваемой проблемы на другой фазе жизненного цикла системы (порой задачи эксплуатации удаётся решить методами, высказанными на этапе производства, и наоборот). Такой перенос близок к приёму «смотри на соседнее окно» классической 9-экранной схемы, но расширяет его диапазон: соседними «окнами» теперь могут быть не только прошлое или надсистема, но и смежные стадии ЖЦ.

Кроме того, ось ЖЦ позволяет явно учитывать, что противоречия могут носить межстадийный характер: например, противоречие между оптимальностью конструкции для производства и удобством обслуживания в эксплуатации. В расширенном операторе эти конфликтующие требования проявляются на разных «слоях» (одно на слое производства, другое на слое эксплуатации), и их можно явно отследить и смоделировать. В классическом же подходе пришлось бы мысленно удерживать оба аспекта сразу, без явной визуальной опоры.

2.3.5. Стейкхолдеры на разных этапах жизненного цикла продукта

В разделе 2.2 рассматривались стейкхолдеры на стадии использования продукта. Однако жизненный цикл продукта включает и другие этапы (от замысла до вывода из эксплуатации), на каждом из которых появляется свой набор заинтересованных лиц. Согласно стандартному подходу системной инженерии жизненный цикл охватывает стадии концепции, разработки, производства, внедрения, эксплуатации и утилизации продукта [8]. Ниже перечислены основные стадии ЖЦ и характерные стейкхолдеры, вовлечённые на каждой из них:

- **Концепция (инициация).** Ключевые участники: заказчики и инвесторы, формулирующие исходные потребности и требования к продукту [8]. Также на этом раннем этапе участвуют аналитики рынка и маркетологи, исследующие спрос, и подразделения, ответственные за определение технических требований и нормативов (например, органы стандартизации) [9]. Формируется команда разработки под руководством менеджера проекта, которую привлекает основной стейкхолдер - инициатор (например, инвестор) [12].

- **Разработка и проектирование.** Ведущую роль играют инженеры-конструкторы и разработчики программно-аппаратных средств. Они создают архитектуру и дизайн продукта, определяют ключевые технические решения. Параллельно участвуют тестировщики (верификаторы), проводящие проверки и валидацию прототипов, а также поставщики ключевых технологий или компонентов (например, производители микросхем, материалов) - их

вклад влияет на проект [Blanchard, Systems Engineering and Analysis, 2008]. На этой стадии могут подключаться регуляторные органы для экспертиз и сертификации проекта (в зависимости от отрасли).

- **Производство.** На производственной стадии в процесс вовлекаются новые акторы: инженеры-технологи, планирующие и оптимизирующие технологический процесс, операционные менеджеры, отвечающие за запуск серийного выпуска, и линейный персонал (например, операторы сборочных линий), непосредственно осуществляющий сборку. Параллельно действует отдел контроля качества, инспектирующий выпускаемую продукцию [8]. Таким образом, выпуск продукта сопровождается сменой фокуса от разработчиков к производственникам - каждый из них является стейкхолдером, влияющим на успех этапа.

- **Внедрение и доставка.** Переход продукта от производства к эксплуатации выделяется в отдельную фазу. Здесь задействованы подразделения логистики и службы внедрения (deployment), которые обеспечивают поставку, установку и ввод продукта в эксплуатацию. Например, для сложных технических систем на этапе внедрения участвуют сторонние монтажные организации и инженеры по наладке оборудования. Кроме того, может привлекаться учебный персонал (инструкторы) для обучения конечных пользователей работе с продуктом [7].

- **Использование (эксплуатация).** На стадии непосредственного использования продукта главными стейкхолдерами становятся конечные пользователи (операторы системы). В эту фазу также входят обслуживающий персонал и технические специалисты, которые поддерживают работу продукта (например, системные администраторы, операторы оборудования) [8]. Именно на этапе эксплуатации проявляются требования пользователей к надёжности, удобству и эффективности продукта, поэтому обратная связь от этой группы стейкхолдеров критически важна для развития системы.

- **Сопровождение и поддержка.** Параллельно с эксплуатацией (или после развёртывания системы) действует фаза сопровождения. Здесь участвуют сервисные службы, отвечающие за ремонт, снабжение запасными частями и техническое обслуживание в период эксплуатации. Система сервисного обслуживания продукта может включать как производителя (обеспечивающего запасные части), так и сеть независимых авторизованных сервис-центров [24]. Например, в индустрии бытовой техники производитель организует поставку деталей, а сторонние сервис-центры выполняют ремонт [9]. Таким образом, на фазе поддержки присутствует несколько групп стейкхолдеров, объединённых общей целью - обеспечение бесперебойной работы продукта на месте эксплуатации.

- **Вывод из эксплуатации (утилизация).** На заключительной стадии жизненного цикла появляются стейкхолдеры, отвечающие за корректное прекращение использования продукта. Сюда относятся организации, занимающиеся утилизацией или переработкой изделия, а также контролирующие органы, надзирающие за безопасностью и соблюдением

нормативов при списании системы [8]. Часто требуется создание отдельной инфраструктуры под этот этап: например, для автомобилей действуют системы сбора и переработки вышедших из употребления машин [25]. В самой организации-владельце продукта ответственность за вывод из эксплуатации несут оперативные службы, которые выполняют работы по отключению, демонтажу и передаче изделия на утилизацию [7].

Из приведенного обзора видно, что состав стейкхолдеров существенно меняется от стадии к стадии жизненного цикла продукта [8]. Группы, влияющие на продукт на этапе концепции и разработки (заказчики, конструкторы, испытатели и др.), отличаются от групп на этапе производства (технологи, сборщики, контролеры качества), а те, в свою очередь, - от стейкхолдеров стадии эксплуатации (конечные пользователи, операторы) или утилизации (специалисты по ликвидации, экологические службы и т.д.). Иными словами, помимо фазы непосредственного использования, на каждой другой фазе жизненного цикла проявляются свои заинтересованные стороны со своими ролями и интересами.

Важно отметить, что внутри каждой крупной стадии могут быть выделены микро-стадии, соответствующие отдельным сценариям взаимодействия, на которых задействованы либо разные акторы, либо один и тот же стейкхолдер выступает в разных ролях. В разделе 2.2 уже было показано, что даже на стадии эксплуатации имеются различные микро-стадии (например, ввод продукта в действие, штатная работа, техническое обслуживание), где участвуют разные стейкхолдеры - от инженеров монтажа до конечных операторов и ремонтных бригад - или происходит смена роли одного и того же актора (пользователь продукта может по совместительству выполнять функции администратора и т.п.) [8]. Подобное разнообразие наблюдается и на других фазах жизненного цикла.

Пример. Стадия производства: в рамках стадии производства продукта можно проследить смену ведущих ролей на последовательных шагах. Сначала в работу включается инженер-конструктор, актуализирующий конструкторскую документацию под серийное изготовление; затем технолог разрабатывает и настраивает технологический процесс; после чего оператор сборочной линии выполняет непосредственную сборку изделия. На каждом из этих шагов - проектирование, технологическая подготовка, сборка - главным действующим лицом выступает отдельный специалист со своей функцией [24]. Таким образом, даже внутри одной фазы происходит последовательная смена сценариев взаимодействия: разные профессионалы поочередно становятся ключевыми стейкхолдерами процесса.

Пример. Сквозной цикл: на протяжении полного жизненного цикла продукта роли стейкхолдеров могут переходить от одних участников к другим. Изначально, на этапе концепции, главную роль играет заказчик или инициатор, формулирующий запрос и финансирующий разработку [17]. На стадии разработки и испытаний ведущим становится верификатор - специалист или орган, проверяющий соответствие создаваемого решения предъявленным требованиям (например, отдел приемочного контроля или независимый аудит) [8]. Далее, в фазе эксплуатации центральным стейкхолдером становится конечный пользователь, от удовлетворенности которого зависит успех продукта в использовании [9]. Таким образом, «заказчик → верификатор → пользователь» - это сменяющаяся цепочка ключевых акторов по мере продвижения продукта через стадии жизненного цикла, отражающая переход ведущих сценариев взаимодействия от одной группы к другой.

На разных стадиях жизненного цикла продукта участвуют различные стейкхолдеры, и в пределах одной стадии могут происходить изменения ролей на микро-стадиях. Учет этой динамики имеет важное значение для системного анализа продукта. Полная идентификация стейкхолдеров на всех этапах и понимание изменения их ролей позволяет учесть множество точек зрения при разработке и эксплуатации системы [8]. Это, в свою очередь, необходимо для построения системного оператора продукта - целостной модели, описывающей продукт как систему во времени.

Включив всех ключевых стейкхолдеров и эволюцию их ролей на микро-стадиях (т.е. охватив все значимые сценарии взаимодействия на различных этапах), такой подход обеспечивает формирование полного набора требований и ожиданий к продукту [24]. Иными словами, анализ ролей заинтересованных сторон (и соответствующих сценариев их взаимодействия с системой) на протяжении всего жизненного цикла служит основой для комплексного понимания и успешного управления развитием продукта в рамках системного оператора.

2.4. Связь модели ЖЦ продукта с моделью бизнес-системы

В данном разделе выделяются две ключевые задачи, связанные с интеграцией продукта в бизнес-систему, и анализируется их отражение в расширенном системном операторе: (1) **обеспечение соответствия продукта проблеме пользователя (problem-solution fit)** на ранних этапах жизненного цикла, и (2) **обеспечение соответствия продукта рынку (product-market fit)** на этапах масштабирования продукта. Кроме того, рассматривается модель бизнес-системы В. Сушкова [48] с точки зрения охвата ею разных фаз развития

продукта, а также обсуждается значение указанных задач для стартапов и крупных компаний. Наконец, формулируются выводы о том, как развитие бизнес-ТРИЗ должно учитывать обе эти задачи и соответствующие стадии жизненного цикла нового продукта.

Модель бизнес-системы В. Сушкова представлена на следующем рисунке.

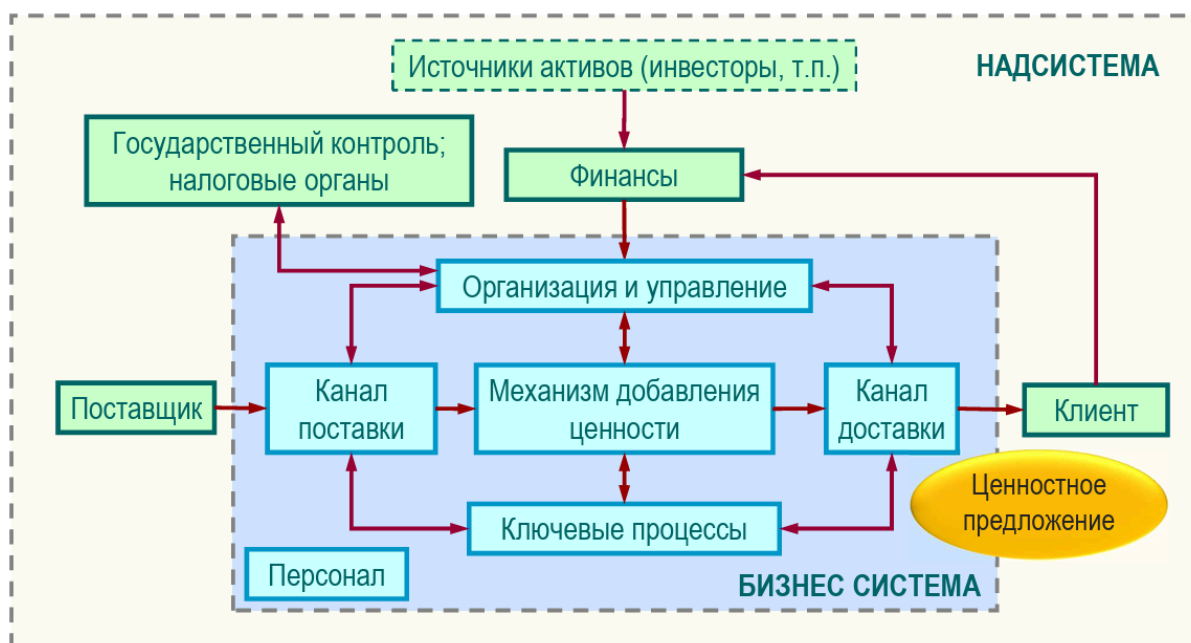


Рисунок 2.6. Модель бизнес-системы В. Сушкова

2.4.1. Модель бизнес-системы и ЖЦ продукта

В рамках Бизнес ТРИЗ одним из подходов к описанию взаимодействия продукта и бизнеса является модель бизнес-системы, предложенная Валерием Сушковым (TRIZ Master, IBTA) [48]. Данная модель структурирует уровни и инструменты бизнес-ТРИЗ для решения бизнес-задач и, по сути, ориентирована на улучшение и развитие уже функционирующей бизнес-системы вокруг продукта.

Так, в описании уровней бизнес-ТРИЗ Сушков выделяет: первый уровень - устранение *ключевых существующих* проблем компании (слабые продажи, сбои в процессах и т.п.), второй - снятие ограничений роста и повышение эффективности *имеющихся* систем и процессов, третий - прогнозирование эволюции системы и предупреждение рисков для дальнейшего развития.

Фактически, акцент сделан на оптимизации и масштабировании продукта и связанных с ним бизнес-процессов, когда основные параметры продукта и его

ценность уже определены. Вторая задача, **product-market fit** (масштабирование ценности), таким образом лежит в ядре модели Сушкова.

Однако данная модель в меньшей степени охватывает самые ранние стадии жизненного цикла продукта, связанные с формированием и проверкой гипотез о продукте (этап **problem-solution fit**). Слабое отражение этих стадий проявляется в том, что модель Сушкова практически не описывает процессы генерации концепции нового продукта, поискового эксперимента с пользователями и начальной настройки ценностного предложения. Например, на этапах **идея** → **прототип** → **пилотное внедрение** перед компанией стоят специфические задачи: выявить, какую **новую** проблему стоит решить; определить минимально необходимый набор свойств продукта, критичных для пользователя; проверить, готов ли пользовательский сегмент изменять своё поведение или платить за предлагаемое решение; установить оптимальные условия поставки или формы использования продукта в небольшом масштабе.

Именно здесь - в стадии зарождения продукта - формируются контуры его будущей бизнес-системы: определяются целевая аудитория, ключевые ценностные свойства, требуемые партнеры и ресурсы, возможные каналы продвижения. Эти элементы затем лягут в основу бизнес-модели при масштабировании. В модели же Сушкова основное внимание уделяется уже *реализованной* ценности и тому, как её масштабировать и улучшать. Таким образом, **первичная фаза поиска соответствия “продукт-проблема” оказывается вне фокуса его модели бизнес-системы.**

Следует подчеркнуть, что отсутствие явного описания ранних этапов не умаляет ценности модели Сушкова для задач оптимизации бизнеса; однако для полноты методологии Бизнес ТРИЗ необходимо дополнить ее инструментами и этапами, учитывающими процесс формирования нового продукта. В противном случае существует разрыв между моментом возникновения инновационной идеи и тем этапом, когда модель бизнес-системы начинает «работать» (фаза масштабирования и совершенствования). Заполнение этого разрыва - за счет интеграции стадий **problem-solution fit** - позволит связать воедино весь жизненный цикл инновационного продукта в рамках единой методологической схемы.

На следующем рисунке показана связь между ЖЦ продукта, моделью бизнес-системы В. Сушкова [49], и задачами продукта. Также показано место для еще одной модели для ранних стадий ЖЦ продукта, которой не хватает в Бизнес ТРИЗ.

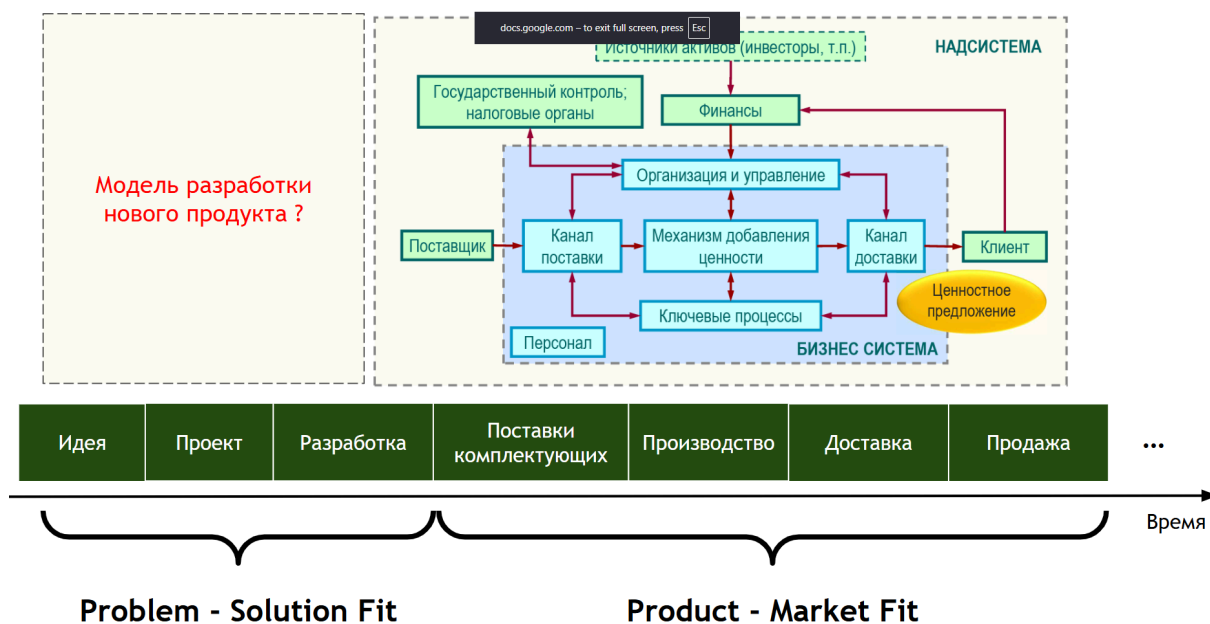


Рисунок 2.7. Модель бизнес-системы на ЖЦ продукта

Кроме того, расширенный оператор охватывает и **динамику эволюции** продукта и бизнес-системы: переход от локальной полезности к масштабной. Это означает, что при анализе продуктовой инновации с помощью системного оператора ТРИЗ необходимо проследивать, **как изменяются противоречия и проблемы на разных стадиях**. На стадии problem-solution fit типичные противоречия могут лежать в плоскости «требования пользователя vs. технические ограничения продукта» или «необходимость создать ценность vs. ограниченность ресурсов стартапа». На стадии product-market fit противоречия смещаются: «необходимость быстро расти vs. ограниченность операционных возможностей», «индивидуальные запросы клиентов vs. стандартизация для массового рынка» и т.п. Расширенный оператор позволяет разместить эти разнородные проблемы по соответствующим клеткам “времени” и “уровня”, тем самым создавая системную карту задач для методического решения средствами ТРИЗ.

Таким образом, интеграция обеих задач (ранней и поздней) в схему оператора обеспечивает целостность анализа: видно, что успех продукта зависит как от **своевременного разрешения начальных противоречий, препятствующих нахождению product-solution fit**, так и от преодоления последующих противоречий роста на пути к product-market fit.

2.4.2. Значение этапов ЖЦ продукта для компаний

Важность рассмотренных задач несколько отличается в контексте стартапов и крупных корпоративных игроков, хотя по сути и те, и другие проходят через оба этапа при создании нового продукта.

Стартапы. Для стартапов *problem-solution fit* и *product-market fit* являются фактически вопросом выживания и роста. На ранней стадии стартап (особенно технологический) обычно ограничен ресурсами и вынужден работать в условиях высокой неопределённости. Методологии бережливого стартапа (*Lean Startup*) и развитие клиента [50] акцентируют, что стартап должен сначала убедиться в наличии проблемы, которую стоит решать, и в ценности предлагаемого решения для первых клиентов, прежде чем масштабировать продукт или привлекать существенные инвестиции. Если такая проверка не произведена, стартап рискует «строить и продавать то, что никому не нужно».

Поэтому успешные молодые компании сначала сфокусированы на тесном взаимодействии с небольшим пулом ранних пользователей, итеративной доработке продукта и *пивотах* (изменениях концепции) до тех пор, пока не будет достигнут явный отклик: пользователи подтверждают готовность пользоваться продуктом и платить за него (стадия *problem-solution fit*).

Лишь после этого стартап направляет усилия на агрессивный рост, масштабирование маркетинга и продаж для захвата рынка - т.е. на достижение *product-market fit* и экспоненциального роста. Практически все известные «единороги» прошли через эту последовательность. Классическим примером является компания **Airbnb**: ее основатели изначально проверили гипотезу, будут ли люди платить за аренду жилья у незнакомцев, с помощью простого эксперимента - предложили участникам конференции в Сан-Франциско переночевать на импровизированных местах в их квартире. Получив первые бронирования и подтверждение интереса (пользователи решили свою проблему с жильем во время мероприятия), стартап добился первичного *problem-solution fit*. Выяснилось, что при крупных событиях городские отели переполнены - возник нишевый спрос, который их платформа могла удовлетворить. Далее, вооружившись этой подтвержденной концепцией, Airbnb поступила в акселератор *Y Combinator*, получила инвестиции и масштабировала модель на другие города, систематически работая над доверием пользователей и качеством сервиса. В течение последующих этапов компания достигла полного *product-market fit* на глобальном рынке: миллионы людей стали регулярно пользоваться её сервисом, что подтолкнуло экспоненциальный рост бизнеса. Этот пример демонстрирует разницу между двумя этапами: первый успех измерялся в штуках бронирований и отзывах первых клиентов, а успех второго этапа - в массовом притоке пользователей, географической экспансии и укреплении бренда.

Корпорации. Крупные компании, выводя на рынок новые продукты или инновации, также обязаны учитывать оба типа соответствия, хотя условия их достижения отличаются. С одной стороны, у корпорации обычно есть достаточные ресурсы, репутация и доступ к существующей клиентской базе, что упрощает ей масштабирование нового решения (т.е. достижение

product-market fit может идти быстрее благодаря готовым каналам сбыта и маркетинговым возможностям). С другой стороны, корпоративные инновации часто сталкиваются с внутренними барьерами и менее гибкими процессами, что затрудняет стадию problem-solution fit.

Внутри большой организации новая продуктовая идея проходит стадии согласований и прототипирования, аналогичные стартапу, но может страдать от недостатка прямого контакта с конечным пользователем и от давления быстро показать отдачу. Тем не менее, лучшие практики корпоративных инноваций перенимают подходы стартапов: создаются *инкубаторы* и *акселераторы* внутри компаний, небольшие кросс-функциональные команды работают по принципу Lean Startup [51], быстро тестируя гипотезы на реальных пользователях и даже выпуская ограниченные пилотные продукты на рынок для проверки.

Примером может служить стратегия компании Google при запуске новых сервисов: многие продукты Google (как, например, Gmail в 2004 г.) сначала запускались в режиме бета-версии с ограниченным доступом. Это позволило компании собрать обратную связь от ранних приверженцев и убедиться, что сервис действительно решает важные пользовательские задачи (problem-solution fit подтвержден) прежде, чем широко распространять его на всех пользователей. После успешного бета-периода, когда метрики удовлетворенности и вовлеченности пользователей показывали ценность продукта, Google масштабировал сервис для массового рынка - тем самым достигая product-market fit с минимальными рисками репутации.

Другой пример - Amazon с продуктом Kindle: перед тем как устройство стало массовым, Amazon провела обширное тестирование прототипов электронного ридера на фокус-группах, проверяя, будет ли целевая аудитория воспринимать чтение с экрана как решение своей проблемы (удобство доступа к книгам) при приемлемых условиях. Только убедившись в положительной реакции (problem-solution fit), компания запустила Kindle на широкий рынок, используя свою мощную платформу e-commerce для быстрого масштабирования продаж (в итоге достигнув product-market fit, что выразилось в доминировании Kindle на рынке электронных книг).

В то же время история знает и обратные случаи, когда корпорации, пропустив стадию глубокого понимания потребности, терпели неудачу с новыми продуктами, несмотря на агрессивный запуск. Например, неудачный запуск устройства Google Glass часто объясняют отсутствием явного problem-solution fit: продукт предлагал технологически продвинутое решение, для которого не оказалось достаточно острой и массовой

потребности на рынке, вследствие чего, несмотря на масштабный пиар, не удалось достичь *product-market fit* и проект был свернут.

Из приведенных примеров видно, что вне зависимости от масштаба организации, **закон жизненного цикла инновационного продукта** требует пройти через обе фазы. *Стартaпы* вынуждены уделять *problem-solution fit* максимум внимания из-за ограниченных ресурсов и отсутствия права на ошибку - один неверный продукт, не решающий реальной проблемы, и компания исчезает. *Корпорации* могут позволить себе больше экспериментов, но каждая неудача стоит значительных средств, а успех нового продукта в портфеле влияет на стратегическое положение фирмы. Поэтому и там, и там залог успеха - **последовательное решение двух задач**: сначала создать продукт, несущий ценность (решающий проблему), затем наладить механизмы доставки этой ценности большому числу клиентов.

2.4.3. Учет задач продукта при развитии Бизнес ТРИЗ

Рассмотренные аспекты ясно показывают, что для эффективного применения Бизнес ТРИЗ необходимо явно учитывать обе ключевые задачи жизненного цикла продукта - как на этапе зарождения идеи, так и на этапе её масштабирования. Расширение системного оператора и общей методики ТРИЗ в сторону **интеграции этапов *problem-solution fit*** обеспечивает методологическую поддержку процессу формирования инновационного продукта.

Это означает, что Бизнес ТРИЗ должна включать инструменты для генерации и проверки гипотез на ранних стадиях, работы с неопределенностями и поиском противоречий в потребностях клиентов. Одновременно, сохранение традиционного фокуса на этапе ***product-market fit*** важно для решательных инструментов Бизнес ТРИЗ, позволяющих убирать ограничения роста, оптимизировать процессы и формировать устойчивую бизнес-модель при выходе на широкий рынок. Развитие Бизнес ТРИЗ, таким образом, должно идти по пути **сшивки двух этапов**: от творческой генерации ценности до её максимально эффективной реализации в масштабе.

Следует отметить, что первоначально ТРИЗ создавалась для технических задач, где этапы жизненного цикла продукта хоть и присутствуют, но аспект рыночного масштабирования не был предметом внимания классической ТРИЗ. В Бизнес ТРИЗ же игнорирование одной из фаз жизненного цикла явно снижает эффективность методики. Если ориентироваться только на вторую задачу (масштабирование), как это отчасти прослеживается в модели В. Сушкова, есть риск получить блестяще оптимизированное решение, которое «решает не ту проблему».

Обратная ситуация - заикленность на творческом поиске идеи без умения ее развернуть в устойчивый бизнес - также не приводит к успеху. Поэтому современный подход требует баланса: **Бизнес ТРИЗ должна помогать решать обе задачи: 1) как породить инновационную идею (и довести её до состояния problem-solution fit), 2) обеспечить ее воплощение в успешный рыночный продукт (product-market fit).**

В контексте квалификационной работы по расширенному системному оператору для продукта это означает, что предлагаемая модель или метод должна охватывать весь путь продукта - от формулирования пользовательской проблемы до глобальной надсистемы, в которой продукт занимает прочное место. Разрабатываемые рекомендации и инструменты должны помогать изобретателю или менеджеру инноваций задавать правильные вопросы и решать противоречия на каждом из этих этапов.

Например: *«Какую существенную проблему пользователя мы решаем и как это проверить?»* (этап соответствия продукт-проблема); *«Какие изменения в надсистеме потребуются для масштабирования решения?»* (этап соответствия продукт-рынок); *«Какие противоречия могут возникнуть при переходе от небольшого решения к массовому продукту и как их преодолеть?»* - и т.д. Включение таких вопросов в расширенный системный оператор фактически превращает его в комплексный **навигационный инструмент по жизненному циклу инновации.**

Подводя итог, расширенный системный оператор ТРИЗ для продукта, обогащённый учётом бизнес-системы, позволяет связать воедино стратегию создания новой ценности и стратегию ее распространения. Он обеспечивает системное видение, в котором **ранняя ценность для пользователя плавно перетекает в масштабируемую ценность для рынка**, а методология ТРИЗ сопровождает продукт на всём этом пути. Такой подход особенно актуален для современных быстро меняющихся рынков, где успешны те организации, которые умеют одновременно быть новаторами (генерируя правильные решения для острых проблем) и эффективными бизнес-архитекторами (выводя решения на большую арену).

Развитие Бизнес ТРИЗ должно отражать эту двуединую задачу, предоставляя инструменты и принципы как для стадии **problem-solution fit**, так и для стадии **product-market fit**, тем самым повышая шансы инновационных продуктов на успех от идеи до зрелого бизнеса.

2.4.4. Определение продукта в РСО

На основании представленных в данном разделе аргументов можно дать следующее определение продукта в расширенном системном операторе:

Продукт - это совокупность микро-стадий системы и связанных с ними сценариев взаимодействия стейкхолдеров, на которых стейкхолдеры получают пользу от взаимодействия с системой (система реализует требования стейкхолдеров).

2.5. Модель подсистемы

2.5.1. Подходы к декомпозиции системы на подсистемы

В теории систем и системной инженерии выделяется несколько ключевых подходов к разбиению сложной системы на подсистемы [7]. Принципиально различаются по крайней мере три стратегии декомпозиции: физическая (структурная), функциональная и процессная (временная).

Физическая (структурная) декомпозиция.

Данный подход предполагает разбиение системы на материальные компоненты и модули, связанные устойчивыми связями. Основанием выделения подсистем служат реальные конструктивные части системы и прочные связи между элементами - информационные, логические, иерархические, энергетические и т.д. [8]. Проще говоря, система рассматривается как совокупность пространственно сосуществующих частей, образующих конфигурацию в данный момент времени [8].

Функциональная декомпозиция.

При функциональном подходе система расчленяется по принципу выполняемых ею функций. Анализируются функции системы, как это принято рассматривать в ТРИЗ: задается вопрос «что делает система», независимо от того, как именно это реализовано физически [24]. Группы элементов, совместно выполняющие определенную задачу, объединяются в функциональные подсистемы. Таким образом, сначала выявляются главные функции, затем - основные, второстепенные и дополнительные функции, после чего определяется, какие компоненты их реализуют [7].

Процессная (временная) декомпозиция.

Этот подход фокусируется на динамических аспектах - разбиении системы по этапам процессов или стадиям во времени. Выделяются последовательные шаги или фазы, через которые проходит система при преобразовании входов в

выходы [6]. По сути, процессная декомпозиция описывает систему как набор операций или состояний, сменяющих друг друга (алгоритм функционирования) [7].

Следует отметить, что указанные стратегии декомпозиции не являются взаимоисключающими. На практике сложную систему часто рассматривают в нескольких измерениях сразу - через призму конструкции, функций и процессов [7]. Каждый из подходов высвечивает разные грани системы: структура показывает устройство, функции - назначение, процессы - поведение во времени. Существенно, что не существует единственно «правильного» способа декомпозиции: выбор разбиения определяется целями и опытом исследователя. Два разных эксперта или коллектива, анализируя одну систему, могут предложить разные декомпозиции системы на подсистемы [36].

2.5.2. Декомпозиция в расширенном системном операторе

В классической многоэкранке («системный оператор» Альтшуллера) декомпозиция системы на подсистемы задается фиксировано один раз и далее подразумевается неизменной по ходу анализа [30]. Иначе говоря, выбрав рассматриваемую систему, изобретатель определяет ее основные подсистемы и надсистему и работает с этой иерархической триадой, исследуя её во времени (прошлое-настоящее-будущее) [30]. Классический подход обычно предполагает анализ системы на одной стадии ее жизни (как правило, на текущей стадии эксплуатации продукта), без явного дифференцирования других фаз жизненного цикла. В результате разбиение «система - подсистемы» в традиционном системном операторе отражает единственный способ декомпозиции, релевантный выбранному этапу.

Как отмечалось ранее (см. раздел 2.3, «Модель жизненного цикла системы и продукта»), стандартный оператор не настраивает мышление на смену фаз разработки, производства, эксплуатации, утилизации и др., из-за чего ряд аспектов может выпасть из поля зрения [8].

В расширенном системном операторе продукта, предложенном в данной работе, вводится дополнительное измерение, связанное с жизненным циклом (ЖЦ) продукта (см. [8]). Новая ось ЖЦ превращает плоскую 3×3 матрицу Альтшуллера в объемную модель (условно - куб мышления) с «прослойками», соответствующими стадиям и микро-стадиями жизненного цикла продукта. При этом на разных стадиях система может рассматриваться через **различные декомпозиции**. Иными словами, разбиение на подсистемы больше не зафиксировано раз и навсегда - оно может меняться от одной стадии к другой в зависимости от контекста.

Такой подход отражает реальность: на каждом этапе жизненного цикла акценты в устройстве и функционировании продукта меняются, и разумно **«переопределять» подсистемы под конкретный контекст стадии и микро-стадии**. К примеру, на стадии **концепции** продукт целесообразно представить через абстрактные функциональные блоки и требования над-системного уровня (больше внимания главной функции и интеграции в окружение); на стадии **проектирования** - через подсистемы, выделенные по главным техническим модулям и функциями (архитектура изделия); на стадии **производства** - через разделение продукта на технологические узлы и сборочные единицы, критичные для изготовления (структурная декомпозиция с упором на технологичность компонентов); на стадии **эксплуатации** - через совокупность компонентов, влияющих на пользовательский опыт (например, интерфейс, основные рабочие агрегаты, расходные материалы), а также через инфраструктуру поддержки продукта (надсистему). Соответственно, на **стадии обслуживания** логично рассматривать разбиение продукта с точки зрения узлов, подлежащих техническому обслуживанию или замене (например, сменные модули, запчасти, программные обновления), то есть выделять подсистемы по признаку обслуживаемости. И наконец, на **стадии утилизации** фокус может сместиться на материалы и компоненты, подлежащие переработке, опасные элементы и т.п., что тоже представляет отдельный вариант декомпозиции системы.

Таким образом, расширенный системный оператор предполагает **динамическую декомпозицию**: на каждой микро-стадии ЖЦ используется наиболее уместный способ разбиения системы на подсистемы - физический, функциональный, процессный или их комбинация. Выбор основывается на том, какие аспекты выходят на первый план в данной фазе. В каком-то смысле, на оси ЖЦ система «пересобирается» для анализа под новый ракурс. Это не противоречит целостности модели, поскольку все эти представления связаны одним продуктом, а лишь дополняют друг друга. На практике реализация оси ЖЦ в системном операторе означает, что эксперт, перемещаясь вдоль этой оси, **сменяет контекст** на другую стадию жизни продукта. При этом он может пересмотреть, какие подсистемы и связи актуальны на новом этапе, и тем самым выявить там специфические противоречия или возможности, которые не были видны при рассмотрении системы только в одном разрезе. Визуально такой подход можно представить как «стопку» матриц системного оператора для ключевых этапов ЖЦ: на каждой из них своя картина системы. На рис. 2.5 в разделе 2.3 схематично показано, как классическая 9-экранная матрица разворачивается в трехмерную модель с учетом оси жизненного цикла. В следующем подразделе рассмотрены преимущества этого решения.

2.5.3. Вариативная декомпозиция системы

Включение оси жизненного цикла и гибкой декомпозиции в системный оператор существенно обогащает аналитический инструментарий разработчика. Прежде всего, это обеспечивает **полноту системного мышления** при работе с продуктом: эксперт вынужденно проверяет решения и гипотезы **на каждой стадии ЖЦ**, не ограничиваясь привычным «срезом» настоящего времени.

Кроме того, ось ЖЦ в системном операторе способствует лучшей **координации межфункционального взаимодействия** при разработке и сопровождении продукта. Наглядное представление продукта во времени позволяет связать воедино точки зрения различных подразделений компании - конструкторов, технологов, маркетологов, сервис-инженеров и др. Каждая функция традиционно фокусируется на “своём” этапе жизненного цикла, что создает риск фрагментации общей картины. Использование же единой трёхмерной модели (куба) облегчает коммуникацию: все участники проекта видят связи между своими областями ответственности на общей схеме.

Таким образом, **динамическая декомпозиция по стадиям ЖЦ расширяет горизонты анализа**: позволяет выявлять конфликты и точки согласования между требованиями разных этапов (разработка vs. эксплуатация, производство vs. утилизация и т.д.), а также заимствовать идеи и решения между стадиями. Последнее особенно важно для инноваций: иногда решение задачи на текущем этапе находится через перенос подхода, применяемого на другой фазе (например, использование принципов, успешно работающих в производстве, для улучшения обслуживания, и т.п. - классические примеры такого переноса рассматривались в разделе 2.3.2).

Суммируя, включение вариативной модели подсистем в системный оператор повышает его эвристическую ценность: эксперт получает более глубокое **«трехмерное» понимание системы во времени**, что облегчает прогнозирование развития продукта и принятие стратегических решений.

2.5.4. Примеры: декомпозиция продукта на разных стадиях ЖЦ

Рассмотрим примеры, иллюстрирующие, как одна и та же система может декомпонироваться по-разному в различных фазах жизненного цикла. В каждом примере прослеживается изменение фокуса разбиения на подсистемы от этапа к этапу - что позволяет увидеть дополнительные грани анализа.

Пример. Физический продукт (стиральная машина). На ранних стадиях разработки стиральной машины естественно применить **функциональную декомпозицию**: выделяются подсистемы по основным функциям - узел

стирки (барабан с приводом), узел нагрева воды, система подачи моющего средства, система управления, корпус и т.д. Такое разбиение отражает архитектуру изделия, удовлетворяющую требуемым функциям. Однако на стадии **производства** фокус смещается: здесь более полезна **структурная декомпозиция** на технологические модули. Например, машину разделяют на узлы, собираемые отдельно: бак и барабан в сборе, электродвигатель, панель управления, насос и трубопроводы, корпусные панели. Подсистемы определяются по признаку удобства сборки и тестирования на конвейере. На этапе **эксплуатации** акценты вновь меняются - систему разумно дробить с точки зрения пользователя и сервисного инженера. Выделяются компоненты, влияющие на опыт использования и обслуживание: интерфейс управления (панель с кнопками/дисплеем), механизм загрузки белья (люк, барабан), система отжима и слива (двигатель, насос), системы безопасности (блокировка люка, датчики протечек) и т.д. Подсистемы здесь определяются по их роли во взаимодействии с пользователем и частоте обслуживания. Наконец, на стадии **утилизации** стиральную машину можно разложить по материалам и узлам для переработки: металлические части (бак, барабан, мотор) как лом, пластиковый корпус, электроника (плата управления) как электронные отходы. Таким образом, на каждой стадии ЖЦ - свое «предпочтительное» разбиение: конструктор видит функциональные модули, технолог - сборочные узлы, пользователь - пользовательские и обслуживаемые части, эколог - материальные фракции для утилизации. Расширенный системный оператор учитывает все эти представления, не ограничивая анализ одной схемой подсистем.

Пример. Цифровая система (мобильное банковское приложение). Для программного продукта жизненный цикл также вносит разные перспективы декомпозиции. На этапе **проектирования и разработки** мобильного банка преобладает **функционально-структурный анализ**: система делится на модули по функциональности и архитектуре - интерфейс (UI/UX) для клиентов, модуль авторизации, модуль операций (переводы, платежи), модуль аналитики, интеграционные сервисы (API к банковским системам) и т.д. Подсистемы определяются как логические компоненты со своими задачами, что соответствует практике многослойной архитектуры ПО. При переходе к **внедрению и эксплуатации** (релиз и поддержка приложения) появляется иной ракурс: **процессная декомпозиция** по пользовательским сценариям и сервисам. Например, выделяются подсистемы: регистрация нового пользователя, проведение платежа, оформление кредита, служба поддержки и обновления приложения. Здесь система разложена по **сценариям использования** - сквозным процессам, проходящим через несколько модулей. Это облегчает анализ опыта пользователя и

надежности работы каждого сценария в реальных условиях. Параллельно служба эксплуатации может рассматривать систему через **технические подсистемы** (инфраструктурный взгляд): клиентское приложение, серверная часть, база данных, внешние API - то есть структурно, но с уклоном на поддерживаемость (например, какие компоненты можно обновлять независимо). На этапе **сопровождения и обновлений** приложения вновь применяют функциональный подход (при доработке функций) в сочетании с процессным (при оптимизации бизнес-процессов в приложении). Таким образом, в разных фазах команда продукта работает с различными моделями подсистем: архитектор - с модульной архитектурой, менеджер продукта - с картой пользовательских потоков, инженер сопровождения - с инфраструктурными компонентами. **Системный оператор с осью ЖЦ** позволяет совместить эти представления, выявляя, например, что изменение в одном функциональном модуле скажется на нескольких пользовательских сценариях, или что новый пользовательский сценарий требует добавить компонент в инфраструктуру. Это ведет к более согласованному развитию приложения.

Пример. Сервис (доставка пиццы). Услуга, как и техническая система, может рассматриваться системно на протяжении своего жизненного цикла. В случае сервиса роль стадий играют шаги оказания услуги и поддерживающие процессы. На этапе **проектирования сервиса** (бизнес-модели доставки) полезна **функциональная декомпозиция**: выделяются подсистемы сервиса по функциям - прием заказа (колл-центр или онлайн-платформа), кухня (приготовление пиццы), логистика (доставка курьерами), поддержка клиента, управление запасами ингредиентов и т.д. Эта модель отражает структуру сервиса с точки зрения организации бизнеса. Когда сервис запущен и функционирует, на первый план выходит **процессная декомпозиция** полного цикла выполнения заказа. Систему доставки можно разбить на последовательные этапы: оформление заказа → приготовление → упаковка → передача курьеру → транспортировка → вручение клиенту → обратная связь. Каждый этап - как подсистема-процесс со своими ответственными, требованиями по времени и качеству. Такое разбиение позволяет анализировать узкие места сервиса (например, задержки на кухне или в пути) и оптимизировать каждый шаг. Параллельно можно выделить подсистемы по ресурсам и ролям: отдел курьеров, отдел поваров, IT-система заказа, склад продуктов - это скорее **структурная декомпозиция сервиса** с упором на материально-техническую базу. На этапе **развития сервиса** (масштабирование, франшиза и т.п.) снова может понадобиться функциональный взгляд (какие новые функции добавить - например, отслеживание доставки или программа лояльности) и процессный взгляд

*(как меняются бизнес-процессы при росте объема заказов). В итоге, доставка пиццы как система не имеет единственной раз и навсегда заданной структуры - как **техническая, социотехническая и процессная** система она устроена по-разному. Благодаря расширенному системному оператору, менеджеры сервиса могут рассматривать все эти аспекты: и операционные процессы, и организационную структуру, и ценность для клиента - в увязке между собой. Это помогает выявлять проблемы и точки роста: скажем, понять, что **ускорение доставки** требует не только больше курьеров (структурно), но и оптимизации маршрутов (процессно), и улучшения приложения для заказов (функционально).*

Приведенные примеры демонстрируют главный тезис: **декомпозиция системы на подсистемы не является строго фиксированной**, она зависит от того, на каком отрезке жизненного цикла и под каким углом мы рассматриваем систему. Классический подход ТРИЗ опирался на одну декомпозицию, часто упрощая картину. Напротив, интеграция оси ЖЦ в системный оператор продукта позволяет «поворачивать» систему разными сторонами, оставаясь в единой модели. Это расширяет возможности анализа сложных продуктов и сервисов, делая системное мышление гибче и ближе к реальности их эволюции. В дальнейших главах будет показано, как практическое применение этой идеи помогает в поиске нетривиальных решений и прогнозировании развития продукта.

2.6. Модель эволюции продукта

2.6.1. Ось эволюции в классическом системном операторе ТРИЗ

В системном операторе (многоэкранка) классической ТРИЗ предоставлен системный подход к анализу развития техники во времени и по иерархическим уровням системы [38]. Согласно системному оператору, время рассматривается как отдельная ось анализа: центральный столбец матрицы соответствует системе в настоящем, левый столбец - её прошлому состоянию, а правый - будущему состоянию [3]. Такой временной разрез позволяет проследить **эволюцию системы** - переход от предшествующих форм к современному виду и далее к потенциальным будущим поколениям анализируемой системы.

Как отмечалось в главе 1, использование оси времени облегчает выявление причинно-следственных связей развития системы и прогнозирование направления ее улучшения см. раздел 1.2, Системный оператор). Например, **прошлое системы** анализируется для понимания исходных решений и причин возникновения текущих проблем, а **будущее системы** - для оценки потенциальных последствий и требований, которые возникнут, если проблема

останется нерешенной [39]. В классическом применении системного оператора акцент делается на поиск ресурсов и идей в различных «окнах» - в надсистеме и подсистеме на прошлых и будущих этапах - что расширяет область решения задач [40].

Таким образом, ось времени служит для **расширения контекста проблемы**: талантливый изобретатель анализирует систему не изолированно, а во времени - учитывая предысторию объекта и возможную эволюцию в будущем [39]. Это помогает выявлять **закономерности развития**: как отмечал Г.С. Альтшуллер, технические системы развиваются по объективным законам в направлении повышения идеальности [38].

Примечание: Закон повышения идеальности гласит, что «развитие всех систем идет в направлении увеличения степени идеальности», то есть соотношение полезного эффекта системы к сумме затрат и вреда неуклонно растет [41].

2.6.2. Роль оси жизненного цикла продукта

Новая ось ЖЦ продукта позволяет выявить **микро-стадии** продукта, в том числе скрытые или пассивные, которые не отражены напрямую на оси «прошлое-настоящее-будущее» [42]. Так, между ключевыми событиями эволюции продукта существуют промежуточные этапы: периоды хранения, ожидания, переналадки, транспортировки, простои между активными циклами работы и т.д. Без специального рассмотрения эти стадии остаются «за кадром» анализа, хотя именно на них часто скрыты резервы и проблемы. Введение оси ЖЦ позволяет разложить жизнь продукта на последовательность **активных и пассивных фаз**, получив более детальную картину его функционирования на каждом этапе времени.

Ось жизненного цикла дополняет временной разрез системного оператора, фактически добавляя измерение «**внутреннего времени**» системы. Если ось эволюции отвечает на вопрос, *как меняется сама система от поколения к поколению*, то ось ЖЦ отвечает на вопрос, *что происходит с продуктом на каждой фазе его существования*. Такая комбинация дает возможность проводить анализ в формате «эволюция × жизненный цикл»: для каждого состояния системы (прошлого, настоящего, будущего) рассматривать полный цикл её работы, включая скрытые стадии. **Активные стадии** - это этапы, когда продукт выполняет свою основную функцию (например, непосредственное использование, работа устройства), а **пассивные стадии** - периоды, когда активность продукта приостановлена, но система продолжает существовать и может предъявлять требования или ресурсы (например, хранение, транспортировка, зарядка, техобслуживание). Включение пассивных стадий значительно обогащает анализ.

Во-первых, эти стадии часто связаны с потерями (временными, энергетическими, финансовыми) или рисками, которые можно уменьшить.

Во-вторых, на пассивных этапах можно найти дополнительные ресурсы для повышения эффективности продукта. Согласно закону согласования частей системы в ТРИЗ, для жизнеспособности системы необходимо согласовать работу всех её частей и процессов во времени, избегая неиспользуемых «простоев» [43]. Ось ЖЦ как раз и нацелена на выявление таких простоев и возможностей их преобразования в полезные действия.

Таким образом, добавление измерения жизненного цикла к классическому системному оператору обосновано: оно обеспечивает **более полный системный анализ**, охватывающий не только изменение продукта во времени, но и циклическую активность внутри каждой временной точки [42].

2.6.3. Изменение микро-стадий при эволюции продукта

При переходе от прошлого состояния продукта к настоящему и будущему наблюдаются закономерные изменения в структуре его жизненного цикла. Как правило, **количество микро-стадий** увеличивается, а их **характер (качество)** усложняется [60]. Прошлые поколения технических систем обычно имели более простой жизненный цикл с минимальным числом этапов. Современные продукты становятся более сложными: добавляются новые функции и режимы работы, а вместе с ними - новые стадии в цикле (например, этапы настройки, обновления ПО, зарядки аккумуляторов, взаимодействия с цифровыми сервисами).

Качество стадий также эволюционирует: многие процессы, ранее выполнявшиеся пользователем вручную (активные стадии), автоматизируются или переносятся в скрытые фазы. Например, в современных устройствах значительная часть активности происходит пассивно (фоновые процессы, автодиагностика, ожидание событий), тогда как в устаревших аналогах таких стадий не было. При движении в будущее тенденция дробления жизненного цикла продолжается: прогнозируемые продукты включают еще больше микро-стадий - как активных, так и пассивных - за счёт интеграции интеллектуальных функций, сетевого взаимодействия и адаптивных режимов [44].

Анализ динамики микро-стадий на оси времени показывает, что от прошлого к будущему происходит:

(1) **рост числа стадий** - цикл становится более детализированным, охватывая все моменты взаимодействия продукта с пользователем и средой;

(2) **специализация стадий** - каждая фаза выполняет более узкую функцию (например, отдельные этапы: инициализация, нормальная работа, режим энергосбережения, самодиагностика, обновление, завершение работы);

(3) **повышение сопряженности стадий** - переходы между фазами становятся более управляемыми, плавными, сокращаются мёртвые промежутки времени. Таким образом, эволюция продукта сопровождается превращением линейного жизненного цикла в все более разветвленный и непрерывный процесс. Этот факт подтверждает необходимость использовать ось ЖЦ при прогнозировании: только учитывая изменение микроструктуры цикла, можно адекватно оценить будущие требования к системе и возможные улучшения на каждой стадии ее работы (см. раздел 1.4, Развитие технических систем).

2.6.4. Примеры эволюции с учетом оси эволюции и оси ЖЦ

Рассмотрим три примера - **зубная щётка, смартфон и сервис доставки пиццы** - чтобы проиллюстрировать сочетание классического временного анализа и анализа жизненного цикла продукта.

***Пример. Зубная щётка. Эволюция во времени:** В прошлом роль зубной щетки выполняли простейшие средства гигиены - от жевательных веточек до примитивных щеток с натуральной щетиной. К середине XX века сформировалась привычная система настоящего - ручная щётка с пластиковой ручкой и синтетической щетиной. В настоящее время появились электрические зубные щетки с вибрацией и вращением, а также «умные» щетки, способные подключаться к смартфону для мониторинга качества чистки. Будущие поколения зубных щеток могут включать автоматизацию процесса (например, ультразвуковые каппы, роботизированные системы очистки) и интеграцию с системой ухода за здоровьем (сенсоры состояния зубов, подключение к интернет-сервисам).*

*Эволюция жизненного цикла: Упрощённая щётка прошлого имела короткий и прямолинейный ЖЦ: производство → использование (чистка зубов) → износ и выброс. **Активная стадия** фактически ограничивалась самим процессом чистки, а пассивные стадии почти не рассматривались (хранение щётки мало влияло на ее функцию). В современной электрической щетке жизненный цикл гораздо сложнее: добавлены стадии **зарядки батареи** (пассивная стадия, когда щётка не чистит, но потребляет энергию), **ожидания/хранения** между применениями, **обслуживания** (смена насадок, чистка устройства). Качество стадий изменилось: например, во время хранения современные щетки могут выполнять само сушку и дезинфекцию (скрытая активность). В будущем ожидается дальнейшее усложнение ЖЦ зубной щётки - появление стадий **цифрового взаимодействия** (передача данных о гигиене в приложение - пассивно-фоновый процесс), **автономной***

очистки/стерилизации щетки после использования (пассивная стадия, повышающая гигиену), а также этап рециклинга электронных компонентов при утилизации. Количество микро-стадий возрастет, при этом каждая стадия будет более специализированной (например, отдельные режимы работы для разных зон полости рта) и более связанной с общим циклом ухода за здоровьем пользователя.

***Пример. Смартфон. Эволюция во времени:** Прошлые поколения мобильных телефонов (1990-е годы) представляли собой простые устройства для звонков и SMS, с ограниченной функциональностью. **Настоящий этап** - современный смартфон - это многофункциональная платформа, сочетающая связь, камеру, компьютер, датчики и доступ к интернет-сервисам. За последние десятилетия смартфон эволюционировал от монофункционального прибора к сложной системе, став центром цифровой жизни. **Будущее** смартфона может включать его трансформацию в носимые или имплантируемые устройства, полноценно интегрированные с окружением (концепция «умного окружения»), либо замещение смартфона распределенной системой устройств (очки дополненной реальности, умные наушники и пр.). Эволюция жизненного цикла: Ранний мобильный телефон имел относительно простой цикл: производство → период эксплуатации (регулярные звонки, периодическая подзарядка) → устаревание и замена новым. Активная стадия использования (разговор) чередовалась с пассивными периодами ожидания вызова, но в старых аппаратах эти пассивные фазы не играли значительной роли (устройства ничего не выполняли в ожидании, потребляя минимум энергии). Современный смартфон функционирует практически непрерывно, распадаясь на множество микро-стадий. Помимо основных **активных стадий** (взаимодействие с пользователем: звонки, приложения, съемка фото и т.д.), существует множество **пассивных стадий**: режим ожидания с подключением к сети, фоновая синхронизация данных, загрузка обновлений, геолокационный трекинг, периодическая зарядка аккумулятора, периоды бездействия экрана и пр. Качество пассивных стадий резко возросло - устройство даже в спящем режиме выполняет работу (приём уведомлений, обновление системы). Жизненный цикл смартфона теперь включает также **стадии обновления программного обеспечения** (регулярные апдейты продлевают актуальность устройства - новая стадия между производством и устареванием), **ремонта/замены компонентов** (например, замена аккумулятора - отдельный этап в середине жизненного цикла продукта) и **утилизации/рециклинга электронных компонентов**. В будущем можно ожидать, что границы между активным использованием и пассивным состоянием будут еще больше размываться: устройства будут постоянно подключены и активны на низком уровне (например,*

непрерывный мониторинг окружающей среды или состояния здоровья пользователя). Число микро-стадий возрастает за счёт появления этапов взаимодействия с другими элементами экосистемы (например, **автоматическая передача задач между устройствами** - новая стадия в цикле работы гаджета). Таким образом, с эволюцией телефонов их жизненный цикл из простого и прерывистого (зарядить-поговорить-выключить) превратился в сложный **непрерывный процесс**, требующий анализа всех фаз для совершенствования продукта.

Пример. Сервис доставки пиццы. Эволюция во времени: В прошлом доставка пиццы осуществлялась по простой схеме: клиент звонил по телефону в пиццерию, заказывал пиццу, после приготовления курьер вручную доставлял заказ по адресу. Этот вариант сервиса имел ограниченные возможности отслеживания и контроля (клиент не знал точно времени прибытия, оплата - наличными при получении). **Настоящий этап** - современные онлайн-сервисы доставки - представляет собой высокоинтегрированную систему: заказ делается через мобильное приложение или сайт, клиент может в реальном времени отслеживать статус (готовится, выехал курьер, прибытие), доступны онлайн-оплата и отзывы. На стороне компании используется оптимизация маршрутов, системы управления курьерами, учёт предпочтений клиентов. **Будущее** доставки пиццы можно прогнозировать как ещё более автоматизированное: использование дронов или автономных роботов для доставки, «умные» кухни с роботизированным приготовлением, предиктивный заказ (система сама предлагает доставку, предугадывая желания клиента) и полная интеграция с домашними устройствами (умный холодильник заказывает пополнение). Эволюция жизненного цикла: В традиционной доставке прошлого жизненный цикл услуги был кратким и линейным: **прием заказа → приготовление пиццы → доставка клиенту → завершение**. Активные стадии здесь - непосредственное приготовление и доставка, пассивные - время ожидания (клиент ждёт, пицца остывает в пути). В современной службе доставки цикл разбивается на большее число этапов с повышенной прозрачностью и управляемостью. После размещения заказа онлайн запускается стадия **подтверждения и обработки заказа** (автоматизированная, часто мгновенная). Далее **приготовление** - активная фаза на кухне. Затем **ожидание курьера** - короткая пассивная стадия, но управляемая: система держит пиццу в термоконтейнере, чтобы сохранить тепло (микро-стадия сохранения качества продукта). **Доставка** - активная стадия для курьера, но теперь сопровождается **отслеживанием GPS** (новая параллельная активность системы, информирующая клиента). **Получение и оплата** - активная фаза с участием клиента, которая упростилась благодаря бесконтактной оплате

(сокращённое время). После доставки добавились стадии, которых раньше не было: обратная связь (клиент оставляет отзыв - ценная информация для системы, хотя для клиента это пассивная часть сервиса) и аналитика данных по заказу (скрытая активность сервиса: обновление рейтинга клиента, пополнение статистики предпочтений). В будущем жизненный цикл услуги доставки пиццы ещё более усложнится: появятся стадии автономного управления (дроны сами планируют маршрут, обходя препятствия - активная стадия без человека), взаимодействия с умным домом (например, холодильник принимает доставку, поддерживая температуру - пассивная стадия для человека, но активная для системы). Кроме того, может добавиться стадия возврата/реутилизации упаковки (для экосистемных целей) - сейчас это редко учитывается. Количественно микро-стадий станет больше (включая процессы подготовки данных, проверок качества на каждом шаге), качественно каждая стадия станет более технологически насыщенной (например, этап готовки может включать ИИ-контроль рецептуры, а этап транспортировки - автопилот). Все это призвано минимизировать пассивные потери времени (ожидание клиента сводится к минимуму) и повысить надежность на каждом отрезке цикла.

2.6.5. Выводы о полезности совмещения двух осей

Комбинация оси эволюции и оси жизненного цикла продукта дает мощный инструмент для системного анализа и прогнозирования развития технических решений. Классический подход ТРИЗ, опирающийся только на ось эволюции, позволяет увидеть **общую траекторию развития системы** и сформулировать образ будущей системы на более высоком уровне идеальности. Добавление оси ЖЦ углубляет этот анализ, рассматривая **внутреннюю динамику** продукта в каждую эпоху его развития. Совмещенный подход, учитывающий обе оси, обладает следующими преимуществами:

- **Полнота анализа.** Никакой важный этап не упускается: мы учитываем и крупномасштабные изменения между поколениями продукта, и мелкомасштабные процессы внутри каждого поколения. Это особенно важно при поиске скрытых проблем или ресурсов - например, выясняется, на каком этапе цикла возникают основные потери, либо какие пассивные стадии можно превратить в активные полезные функции.

- **Повышение точности прогнозов.** Ось времени дает направление эволюции (как продукт должен измениться согласно законам развития систем), а ось ЖЦ показывает, где именно в работе продукта есть потенциал для этих изменений. Совмещая их, исследователь может более обоснованно предсказать будущие улучшения: например, не просто «продукт станет более

автоматизированным», а *на каких стадиях* будут внедрены автоматические функции и как это повлияет на весь цикл использования.

- **Синергия микро- и макроэволюции.** Традиционная макро-эволюция (новое поколение продукта) тесно связана с микро-эволюцией (усовершенствованиями в рамках жизненного цикла текущего продукта). Использование двух осей одновременно позволяет выявлять эти связи. В некоторых случаях усовершенствование на уровне микро-стадии (например, добавление стадии самодиагностики устройства во время простоя) ведет к качественному скачку, фактически формируя новое поколение продукта. Обратная ситуация: требование нового поколения может быть реализовано через модификацию определенной стадии ЖЦ существующей системы. **Двухосевой анализ** делает такие взаимосвязи наглядными и учитывает их при генерации идей.

Практические примеры (зубная щётка, смартфон, доставка пиццы) демонстрируют, что двухосевой анализ позволяет выявить больше инсайтов о продукте, чем раздельное рассмотрение эволюции или жизненного цикла. Таким образом, объединение оси эволюции и оси ЖЦ продукта оказывается полезным для более глубокого понимания текущих систем и разработки инновационных решений на перспективу, что подтверждается как теоретическими основаниями [38], так и результатами современных исследований [42].

2.7. Заключение к главе 2

В главе 2 предложена расширенная модель системного оператора ТРИЗ, адаптированная для продуктового мышления. Эта модель сохраняет базовые измерения классического оператора (иерархические уровни системы: подсистема-система-надсистема, а также временные срезы: прошлое-настоящее-будущее), дополняя их новой осью ЖЦ продукта. **Ось жизненного цикла продукта** позволяет проследить изменение системы на последовательных стадиях - от разработки и производства до эксплуатации, поддержки и утилизации. Появилась возможность детализировать ключевые этапы эксплуатации через концепцию **микро-стадий использования**: вместо одного усредненного сценария продукт рассматривается в совокупности разных ситуаций применения, чередующихся во времени. Кроме того, в модель интегрирован учет **ролей потребителей (стейкхолдеров)** на каждой микро-стадии, что отражает участие различных пользователей (или смену ролей одного пользователя) при взаимодействии с продуктом. Наконец, расширенный оператор предусматривает **вариативную декомпозицию системы**: состав и структура подсистем продукта могут гибко пересматриваться в зависимости от текущей стадии жизненного цикла и задачи

анализа. Например, на этапе концепции продукт представляется через функциональные блоки и пользовательские ценности, тогда как на этапе производства - через технологические узлы и компоненты, важные для сборки, а на этапе эксплуатации - через модули, влияющие на пользовательский опыт и обслуживание. Такой многоосевой подход значительно обогащает картину анализа продукта.

Введенные расширения усиливают системное мышление при работе с продуктом. Разработчик получает методический инструмент для всестороннего рассмотрения решений: проверка гипотез проводится на **всех стадиях ЖЦ**, что снижает риск упущений требований или проблем, проявляющихся лишь на определенных этапах. Учет множества **сценариев взаимодействия (использования) и ролей стейкхолдеров** дает целостное представление о ценности продукта в разных контекстах, тем самым помогая выявлять скрытые противоречия и точки улучшения продукта. Вариативная декомпозиция, в свою очередь, позволяет взглянуть на систему под разными углами (конструкция, функции, процессы) в наиболее уместном разрезе для каждой стадии. Совокупно эти изменения делают системный оператор продукта более гибким и приближенным к реальности развития изделий и сервисов. Расширенная модель не только охватывает эволюцию техники во времени (по Альтшуллеру), но и добавляет измерение «жизни» самого продукта, стимулируя продумать всю «судьбу» продукта от замысла до вывода из эксплуатации.

Отметим, что классический системный оператор ТРИЗ предполагает также анализ **анти-системы** - условно говоря, системы-антагониста для исходной системы. По аналогии можно ожидать, что в расширенном операторе целесообразно рассматривать и «**анти-продукт**» - объект или процесс, нейтрализующий ценность продукта или выступающий его противоположностью. Однако данная концепция в рамках проделанной работы пока не разработана и явно не включена в модель. В перспективе дальнейших исследований представляет интерес введение оси «анти-продукта», чтобы полнота системного анализа охватывала и противодействующие факторы. Таким образом, глава 2 заложила основу расширенного системного оператора продукта, обозначив новые измерения и направления для развития методики в будущем.

Глава 3. Методика использования

3.1. Методика использования СО в классической ТРИЗ

Построение модели системы в классическом системном операторе включает следующие шаги:

- 1) Определяется сама целевая система анализа и ее границы - например, продукт или техническое устройство, над которым ведется работа.
- 2) Фиксируются основные подсистемы (ключевые компоненты или части системы) и надсистема (окружение, суперсистема, в рамках которой система функционирует). Эти элементы заполняют центральный ряд схемы: текущая система, ее подсистемы и надсистема в настоящем времени.
- 3) Анализируется *предыстория* системы: заполняются левые колонки матрицы. Для каждой из трех уровней (подсистема, система, надсистема) описывается прошлое - что предшествовало текущему состоянию? Например, какими были компоненты системы ранее, как эволюционировала сама система, и каковы были условия в надсистеме (внешней среде) в прошлом.
- 4) Строятся *проекции на будущее*: для каждого уровня формулируются прогнозы или предполагаемые изменения в будущем (эволюция подсистем, развитие системы, изменение надсистемы). Эти шаги позволяют последовательно заполнить все девять ячеек классического оператора.
- 5) Информация, собранная о системе в модели системного оператора, используется в других методиках ТРИЗ и Бизнес-ТРИЗ.

Заполненная многоэкранная схема служит когнитивной картой для выявления проблем и поиска идей решения. Аналитик просматривает “окна” матрицы, чтобы обнаружить потенциальные противоречия и ресурсы. Например, анализируя подсистему в прошлом и настоящем, можно понять, какие технические решения привели к текущим ограничениям; рассматривая надсистему в будущем, выявить новые требования или угрозы для системы. Классический подход тем самым помогает выйти за рамки узкого видения проблемы и учесть *системный контекст во времени*.

Известно, что использование системного оператора позволяет избежать психологической инерции и одностороннего взгляда на задачу: изобретатель рассматривает проблему не изолированно, а во взаимосвязи с окружающей средой и эволюцией системы. В частности, классическая схема облегчает поиск доступных ресурсов (в подсистемах и надсистеме) и

причинно-следственных связей возникновения проблемы на разных уровнях системы.

Для иллюстрации представим применение 9-экранки к простому продукту - например, зубной щетке. Центральный экран (система, настоящее) описывает текущую зубную щетку, выполняющую свою основную функцию - чистку зубов. Подсистемы (настоящее) - основные части щетки: ручка, щетина, возможно, гибкая головка. Надсистема (настоящее) - контекст применения: организм человека, гигиеническая процедура, ванная комната и пр. Левые экраны (прошлое): подсистема - материалы и конструкция щетки прошлых поколений (скажем, щетина из натуральной щетины в старых щетках вместо современных синтетических волокон); система - прежние версии продукта (традиционные щетки прошлого, например, деревянные палочки для жевания или примитивные прототипы щетки); надсистема - исторический контекст (культура гигиены прошлого, отсутствие электрических щеток, иные санитарные нормы). Правые экраны (будущее): подсистема - возможные улучшения компонентов (например, щетина с микроскопическими сенсорами или сменные насадки с инновационным материалом); система - будущее самого продукта (например, умная зубная щетка, способная самостоятельно диагностировать состояние зубов); надсистема - прогноз эволюции контекста (например, изменения в потребительских привычках ухода за здоровьем, появление новых требований к гигиене полости рта). Подобный анализ по классической схеме (см. раздел 1.5) помогает разработчику понять, как продукт развивался и куда может двигаться, а также выявить узкие места: например, сравнение условий надсистемы прошлого и будущего покажет, соответствуют ли текущие характеристики щетки новым трендам и ожиданиям пользователей.

Отметим, что классический системный оператор, хотя и мощен, имеет ряд ограничений применительно к современному продуктовому подходу. В классической 9-экранной модели обычно рассматривается одна фиксированная декомпозиция системы и, как правило, один сценарий применения (текущая эксплуатация). Такой подход не в полной мере учитывает, что современные продукты могут использоваться в различных условиях и иметь сложный жизненный цикл. Кроме того, роли пользователя и требования к продукту могут меняться на разных этапах его жизни. Классический системный оператор также не сдержит механизмов выявления других стейкхолдеров и их требований. Эти недостатки требуют расширения методики классического системного оператора, о чем речь пойдет далее (см. раздел 3.2).

3.2. Отличия методики от классической

Расширенный системный оператор для продукта предложен как развитие классической методики (см. главу 2), включающее дополнительные измерения, отражающие специфику продуктового анализа. Ниже перечислены ключевые отличия методики построения *расширенного* системного оператора по сравнению с классическим (см. также раздел 2.3.4) и обоснована их необходимость:

1) Ось жизненного цикла продукта. В классической модели рассмотрение системы фактически ограничивается одной фазой - обычно фазой эксплуатации (текущего использования). Однако продукт проходит через последовательность стадий: концепция, разработка, производство, внедрение, эксплуатация, обслуживание, утилизация и т.д. Поэтому в расширенной модели вводится дополнительная ось, отражающая стадии *жизненного цикла (ЖЦ) продукта*. По сути, вместо плоской матрицы 3×3 рассматривается многослойная структура, где каждый «слой» соответствует определенной стадии ЖЦ (рис. 2.5 в главе 2 демонстрирует эту идею схематично).

Добавление оси ЖЦ позволяет анализировать систему не только в измерении эволюции техники (прошлое-настоящее-будущее для самой системы), но и в измерении последовательных этапов *жизни продукта* - от замысла до вывода из эксплуатации. Необходимость такого дополнения обусловлена тем, что многие проблемы и требования проявляются лишь на отдельных стадиях: например, на стадии производства встают вопросы технологичности и себестоимости, на стадии обслуживания - удобства ремонта, на стадии утилизации - экологической безопасности.

Классический оператор не гарантирует учета всех этих аспектов (разработчик может непреднамеренно сосредоточиться только на текущей эксплуатации), тогда как ось ЖЦ дисциплинирует мышление проверять решения на каждом из этапов продукта. Включение жизненного цикла в анализ соответствует принципам системной инженерии и концепции проектирования с учетом жизненного цикла (Design for X: manufacturability, maintainability, recyclability и др.), то есть стимулирует выявление требований и ограничений разных фаз при проработке продукта.

2) Сценарии использования и микро-стадии. Реальные продукты обычно эксплуатируются не в единственном, а в нескольких различающихся сценариях применения. Даже простой предмет, как зубная щетка, может использоваться в разных условиях (например, *ежедневная чистка дома vs. чистка в путешествии* - разные контексты применения), а многофункциональный гаджет вроде смартфона - тем более (звонки, фотосъемка, интернет-серфинг, навигация и т.д. - каждый сценарий предъявляет свои требования). Классический оператор оперирует одним фиксированным надсистемным контекстом, тогда как для описания продукта в разных условиях требуется

учесть *множественность надсистем*. В расширенной методике это достигается за счет введения понятия «*микро-стадия*» использования.

Под микро-стадией понимается ограниченный по времени эпизод взаимодействия пользователя с продуктом, соответствующий отдельному сценарию применения. По сути, общая стадия *эксплуатации* продукта делится на совокупность микро-стадий, отражающих разные ситуации, в которых продукт приносит пользу. Эти эпизоды могут следовать последовательно или чередоваться во времени, разделяясь периодами бездействия продукта.

Например, для зубной щетки активная микро-стадия - непосредственно чистка зубов, а пассивная микро-стадия - хранение щетки на полке до следующего использования. Для сервиса доставки пиццы активные микро-стадии - оформление заказа, приготовление, доставка, получение, а пассивные - ожидание (пицца едет, клиент ждет, пицца остывает). **Включение множества сценариев и микро-стадий в модель** нужно для того, чтобы не упустить важные частные эпизоды использования. В каждой сценарии могут возникать свои проблемы и требования. Классическая 9-экранка не показывает внутренних вариаций на стадии эксплуатации, тогда как расширенный оператор явным образом их фиксирует. Это особенно актуально для продуктов-сервисов, процессов и человеко-машинных систем, где *сценарный анализ* позволяет детально изучить путь пользователя (customer journey) и найти узкие места в каждом шаге взаимодействия.

3) Роли пользователей и других стейкхолдеров. Добавление микро-стадий неразрывно связано с учетом различных стейкхолдеров - участников, взаимодействующих с продуктом. В классическом системном операторе надсистема обычно подразумевает пользователя или внешнюю среду в общем виде, без дифференциации, кто именно и как именно взаимодействует с системой на разных этапах. В расширенной методике для каждой микро-стадии явно указывается, **какой участник (актер)** задействован и в какой роли. Один и тот же человек может выступать в разных ролях: например, в одном сценарии смартфона пользователь - это фотограф (делает снимок камерой), в другом - навигатор (следует маршруту), в третьем - слушатель музыки; или в сервисе доставки пиццы клиент последовательно играет роль заказчика (размещает заказ онлайн), получателя (принимает доставку), а затем оценщика (оставляет отзыв).

Кроме того, помимо конечных пользователей, на разных стадиях жизненного цикла появляются и другие стейкхолдеры: инженеры, производственники, обслуживающий персонал, дистрибьюторы, утилизаторы и т.д.. Расширенный системный оператор учитывает все эти категории. Необходимость такого учета очевидна: разные участники предъявляют разные требования к продукту, и часто именно несогласованность между этими требованиями рождает

противоречия. Например, то, что удобно маркетологам (частые обновления продукта), может быть неудобно сервисным инженерам или конечным пользователям (слишком частое изменение интерфейса и функциональности).

Классическая схема неявно предполагает единого «усредненного» пользователя, тогда как современный подход требует многогранного рассмотрения продукта с позиций разных заинтересованных сторон. **Учет ролей** в методике расширенного оператора обеспечивает более полный сбор требований и выявление конфликтов между ними (см. также раздел 3.5.3 ниже).

4) Вариативная декомпозиция системы. Наконец, важное отличие - гибкость в разбиении системы на подсистемы. В 9-экранной схеме Альтшуллера разработчик один раз выбирает, какие подсистемы (элементы) входят в состав системы, и держит эту структуру постоянной во всем анализе. На практике же *состав значимых подсистем* может меняться от этапа к этапу.

Расширенная методика предполагает, что на разных стадиях ЖЦ продукт может рассматриваться через **различные декомпозиции**. То есть набор иерархических элементов не высечен в камне: он адаптируется под тот ракурс, который наиболее уместен на данной стадии.

К примеру, на стадии концепции разумно декомпозировать продукт на **функциональные блоки** или пользовательские функции, чтобы увидеть соответствие замысла потребностям (абстрактная функциональная архитектура). На стадии проектирования - выделить **технические модули** (структурная архитектура изделия), соответствующие главным техническим решениям. На стадии производства - подсистемы определить как **технологические узлы и компоненты**, критичные для сборки и изготовления. В эксплуатации - сфокусироваться на тех подсистемах, что влияют на пользовательский опыт и обслуживание (например, батарея как отдельный узел, интерфейсные модули, расходимые материалы). При обслуживании - рассмотреть подсистемы с точки зрения заменяемости (запасные части, модули, подлежащие ремонту). И так далее, вплоть до стадии утилизации, где на первый план выходят материалы и компоненты, важные для переработки.

Таким образом, расширенный оператор предусматривает *динамическую декомпозицию*: на каждой фазе жизни продукта используется наиболее подходящий способ разделения системы. Это позволяет учесть структурные изменения продукта во времени - что особенно актуально для модульных или эволюционирующих продуктов. **Необходимость вариативной декомпозиции** диктуется тем, что единый статичный список подсистем не отражает полноту картины. Гибкое «пересобирание» модели продукта на разных этапах дает более адекватное представление о системе в контексте этапа и помогает

выявить специфические проблемы, невидимые при одном ракурсе. Практическая ценность такого подхода подтверждается тем, что он облегчает коммуникацию в команде: каждая функциональная группа видит на модели “свою” часть продукта именно там, где это релевантно. Например, маркетологи работают с отображением продукта на этапе вывода на рынок, инженеры - с архитектурой на этапе разработки, производственники - с компонентами на этапе изготовления, и все эти представления интегрированы в единую многомерную схему (куб).

Перечисленные расширения по сути превращают классическую двухмерную матрицу в трехмерную модель (иногда ее образно называют “кубом мышления”). Две оси - иерархические уровни и время - заимствованы у классической ТРИЗ, третья ось - жизненный цикл - добавлена для прохождения по стадиям развития продукта. Помимо них, в модели фигурируют “вспомогательные” измерения: разбиение использования на микро-стадии (детализация оси времени внутри стадии эксплуатации) и распределение стейкхолдеров по ролям (связывает ось ЖЦ с осью надсистемы).

В целом, расширенный системный оператор продукта близок по духу к расширенной «многоэкранной схеме» в OTSM-ТРИЗ (Общая теория сильного мышления), где также предлагается анализировать проблему одновременно по нескольким осям. Идея включить дополнительные измерения восходит к работам Г.С. Альтшуллера: еще при разработке книги «Творчество как точная наука» рассматривалось добавление новой оси (размер-время-стоимость, “PBC-оператора”), однако графически изобразить более сложную модель оказалось затруднительно.

В позднейших разработках (Н. Хоменко и др.) классический системный оператор был обобщен в виде многомерной схемы мышления. Таким образом, предлагаемая расширенная модель опирается как на практические потребности продуктового анализа, так и на идеи, заложенные в развитых версиях ТРИЗ. Она сохраняет преимущества классического подхода, дополняя его новыми измерениями - это подтверждается успешными примерами применения в главе 2 и работами других исследователей [47]. В результате системный оператор продукта становится более гибким и приближенным к реальности эволюции современных изделий и услуг, позволяя учитывать полный путь продукта и разнообразие его использования.

3.3. Этап сбора исходных данных

Разработка расширенного системного оператора продукта начинается с тщательного сбора исходных данных. Цель этого этапа - собрать всю необходимую информацию о продукте, его контексте и использовании, чтобы

на ее основе построить многомерную модель. Ниже перечислено, *что именно* нужно собрать и определить на этом этапе, а также описано, *как* идентифицируются сценарии, микро-стадии, стейкхолдеры и конфигурации системы перед построением модели:

1) Определение продукта и границ системы. Сначала четко формулируется, какой продукт или систему предстоит проанализировать. Нужно описать назначение продукта, его основную функцию и состав. Важно установить границы: что входит в систему, а что относится к внешней среде. Например, если рассматривается электрическая зубная щетка, система включает сам прибор (мотор, щетинки, батарея и пр.), а не включает зубы пользователя или электрическую сеть (они являются частью надсистемы).

Четкое понимание границ необходимо, чтобы в дальнейшем корректно различать подсистемы и надсистему продукта (см. раздел 2.1.1 о минимальной системе, а также раздел 1.1 о понятии системы в ТРИЗ). На этом же шаге фиксируется текущая стадия жизненного цикла, на которой находится продукт (например, опытный прототип, серийное производство, эксплуатируется на рынке или устарел и снимается с производства).

2. Идентификация ключевых сценариев использования. Далее определяется, *как* продукт используется конечными потребителями. Необходимо выявить **основные сценарии взаимодействия** пользователя с продуктом (или услуги с клиентом). Под сценарием понимается отдельная ситуация или задача, ради которой продукт применяется. Здесь полезно опираться на исследования пользовательского опыта, интервью с потребителями, методы маркетинга (включая концепцию *Jobs to Be Done* - «работы, для которой нанимается продукт»). Фактически, нужно ответить на вопрос: *для каких целей, в каких условиях* потребители используют данный продукт? Например, для смартфона можно выделить такие сценарии, как “совершение звонка”, “прослушивание музыки”, “фотографирование”, “навигация в автомобиле”, “общение в мессенджерах” и т.д. Для сервиса доставки пиццы сценарии - “заказ пиццы для себя домой”, “заказ пиццы для вечеринки (групповой заказ)”, “предварительный заказ к определенному времени” и пр.

Даже для простого продукта типа зубной щетки могут быть варианты сценариев: *ежедневное использование дома, взять в поездку* (портативность) и т.д. (см. раздел 2.2.1). Важно ограничиться несколькими (несколькими десятками максимум) **типовыми сценариями**, которые покрывают основные случаи использования. Перебор всех возможных ситуаций не требуется - достаточно охватить диапазон типичных и критичных с точки зрения ценности

для пользователя случаев.

3. Описание стадии использования через микро-стадии. Для каждого выбранного сценария следует описать пошагово, как он реализуется во времени - то есть выделить микро-стадии взаимодействия пользователя с продуктом в рамках этого сценария.

Микро-стадия - это минимальный значимый этап или фаза, в течение которой выполняется часть работы продукта. Например, возьмем сценарий “заказ пиццы через приложение с доставкой на дом”. Его можно разложить на такие микро-стадии: (a) пользователь оформляет заказ в приложении; (b) пицца готовится на кухне ресторана; (c) курьер везет пиццу; (d) пользователь получает пиццу и оплачивает; (e) пользователь употребляет пищу; (f) утилизация упаковки и оставшейся еды; (g) пользователь оставляет отзыв о сервисе. Среди них (a), (b), (c), (d), (e) - **активные стадии** (идет полезный процесс: взаимодействие или работа системы), а (f) и (g) можно отнести к *последовательным пассивным или завершающим стадиям* (прямо пользы не добавляют, но являются частью общего опыта).

Для другого сценария, например “групповой заказ заранее на определенное время”, набор и последовательность микро-стадий будет несколько иной (добавится стадия согласования заказа среди группы, ожидание назначенного времени и т.д.). Таким образом, **результатом** этого шага является карта или список микро-стадий для каждого сценария.

Часто удобно представить их в виде последовательной цепочки (или нескольких параллельных процессов) с указанием, какие стадии идут одна за другой, а какие могут выполняться независимо или повторяться. Здесь пригодятся техники моделирования процессов (например, BPMN-диаграммы, сценарии использования UML, storyboards), но на conceptual уровне можно ограничиться текстовым перечислением шагов.

Важно при этом отметить, какие стадии являются активными (система выполняет функцию) и пассивными (система в ожидании, пауза). Пассивные стадии тоже фиксируются, так как в дальнейшем они могут оказаться источником проблем или резервов (см. п. 3.5.4).

Для физических продуктов микро-стадии в основном относятся к стадии эксплуатации (например, для зубной щетки: смачивание щетки - чистка зубов - ополаскивание - просушка/хранение до следующего раза). А для сервисов или продуктов, представляющих процессы, микро-стадии могут соответствовать шагам процесса (как в примере с пиццей, где процесс *доставки* по сути и определяет микро-стадии). Если продукт задействован в нескольких крупных сценариях, на этом этапе появится несколько наборов микро-стадий (по

одному на сценарий). Далее при построении модели это будет отражено наличием нескольких последовательностей в рамках стадии использования.

4. Выявление стейкхолдеров и ролей на каждом этапе. Параллельно с описанием микро-стадий фиксируется, *кто* участвует в каждой стадии. То есть составляется матрица “стадия - стейкхолдер”: для каждого шага сценария указывается, какие категории пользователей или других участников задействованы и в какой роли.

В простых случаях один и тот же пользователь проходит через все стадии сценария (например, человек сам осуществляет и заказ, и получение пиццы). Но даже тогда его роль может меняться: сначала он **клиент-заказчик**, затем **получатель**, далее **потребитель** продукта, а под конец - возможно, **оценщик** (если оставляет отзыв). В более сложных случаях на разных стадиях участвуют *разные акторы*: в примере с пиццей на стадии приготовления - **повар** (сотрудник ресторана), на стадии доставки - **курьер**, на стадии обслуживания приложения - **ИТ-система** (автоматизированный стейкхолдер), на стадии обратной связи - **менеджер по работе с клиентами** (который читает отзывы) и т.д..

Полный список стейкхолдеров включает не только конечных пользователей, но и все службы и системы, прямо влияющие на жизненный цикл продукта. Для физического продукта (например, смартфона или щетки) помимо пользователя-оператора нужно учесть: **производителя** (инженеры, сборщики - участвуют на стадиях разработки и производства), **дистрибьютора/продавца** (стадия внедрения на рынок), **обслуживание** (сервисные центры на стадии эксплуатации), **утилизатора** или переработчика (конец ЖЦ) и др. Например, для смартфона карта стейкхолдеров по стадиям ЖЦ может выглядеть так: концепция - **продуктовый менеджер, инвестор**; разработка - **инженеры, дизайнеры, тестировщики**; производство - **технологи, сборщики, поставщики компонентов**; внедрение - **маркетологи, дистрибьюторы**; использование - **конечный пользователь, оператор связи, разработчики приложений (косвенно)**; обслуживание - **сервис-инженеры**; утилизация - **компания по переработке отходов** (см. пример в разделе 2.3.4). Все эти роли и участники следует выписать и привязать к соответствующим стадиям и сценариям. Удобно, когда на предыдущем шаге вы записываете микро-стадии, сразу напротив каждой стадии пометить ответственного/участника.

Такой анализ дает ценную информацию: становится понятно, **кто взаимодействует с продуктом на каждом отрезке его “жизни”**. Позже при построении оператора эти данные позволят отмечать в каждом “окне” модели соответствующую точку зрения (например, надсистема на стадии эксплуатации для сценария 1 - это *пользователь в роли А*, а для сценария 2 - *пользователь*

в роли В, на стадии производства надсистема - производственное подразделение, и т.д.).

5. Сбор сведений о конфигурации и структуре системы. Последний важный блок данных - информация о том, как продукт устроен и функционирует на разных этапах. Поскольку мы планируем вариативную декомпозицию, нужно подготовить несколько **представлений системы**:

- Текущее или основное представление: классическая структура продукта - из каких компонентов или модулей он состоит, как они взаимосвязаны.

Например, для щетки: ручка, головка, щетина, возможно, вибромотор (для электрической щетки) и батарейка. Для смартфона: корпус, экран, батарея, камера, процессорный модуль, ПО, и т.д. Для сервиса: ключевые подсистемы сервиса - платформы (мобильное приложение, сервер), физические элементы (кухня, курьеры), бизнес-процессы.

- Представления для разных стадий ЖЦ: если известны особенности на этапах, их фиксируют. Например, на этапе производства смартфона - его *технологическая* структура (узлы сборки: экранный модуль, материнская плата, корпусные детали, которые собираются на конвейере). На этапе обслуживания - структура с точки зрения заменяемых узлов (дисплейный модуль, аккумулятор, которым оперируют сервисные инженеры). На этапе концепции - структура в терминах функциональных блоков (например, модуль коммуникации, модуль обработки данных, модуль интерфейса пользователя - более абстрактно). Для сервиса доставки пиццы: на этапе запуска (концепции) система может рассматриваться через функции (*логистика, кухня, маркетинг*), на этапе масштабирования - через организационные подсистемы (*отдел доставки, отдел производства, ИТ-отдел, подразделение франчайзи*), на этапе эксплуатации - через процессные шаги, как уже делалось (заказ, готовка, доставка).

- Данные о текущих и ожидаемых параметрах системы на разных этапах: например, если имеются технические характеристики прототипа и целевые показатели для серийного образца - их тоже стоит собрать. Или известные ограничения: бюджет производства, ресурсы на обслуживание и прочее. Проще говоря, этот подпункт означает собрать *техническую и организационную информацию* о продукте, релевантную для анализа на разных стадиях. Источниками могут быть техническая документация, бизнес-процессы компании, требования нормативов, результаты предыдущих этапов разработки (концептуальные схемы, прототипы) и т.д. Частично эти данные могли быть разобраны в главе 2 (например, модели продукта, жизненного цикла, подсистем) - на этапе сбора их нужно актуализировать и

конкретизировать под ваш продукт.

В итоге, на выходе этапа сбора данных мы должны иметь: **описание продукта и его функции; перечень основных сценариев использования; сценарии декомпозированы на микро-стадии; список стейкхолдеров/ролей для каждой стадии; описания структуры системы и ее изменений на разных этапах.** Все эти сведения станут “сырьем” для построения расширенного системного оператора. По сути, происходит предварительное информационное моделирование: вы строите ментальную картину того, как продукт живет, кем и когда используется, из чего состоит. Если этот шаг выполнен тщательно, последующая формализация в виде куба (матрицы матриц) не представляет больших трудностей - модель заполняется практически напрямую из собранных списков и описаний.

Следует отметить, что сбор исходных данных для расширенного оператора - итеративный процесс. Новые инсайты могут потребовать вернуться и скорректировать сценарии или добавить стейкхолдеров. Полезно привлекать к этому этапу экспертов из разных областей: инженеров, маркетологов, специалистов по обслуживанию - чтобы информация покрывала все аспекты.

Такой кросс-функциональный подход снижает риск пропуска важных деталей. В реальных проектах данные могут частично извлекаться из существующих документов (технических заданий, карт пользовательского опыта, бизнес-процессов), частично добываться заново через исследования. Но чем полнее и структурированнее будут исходные сведения, тем более качественным получится итоговый анализ с помощью системного оператора.

Пример (иллюстрация сбора данных): Рассмотрим в краткой форме продукт “умная зубная щетка”. 1) Границы системы: сама щетка (включая электронику), пользователь и его рот - надсистема, зубная паста - скорее внешняя взаимодействующая система. 2) Сценарии: ежедневная чистка зубов; обучение ребенка чистке (особый сценарий с ролью родителя-наставника); экспресс-чистка вне дома (поездка, отель). 3) Микро-стадии (для ежедневной чистки): подготовка (нанесение пасты) - активная; чистка зубов - активная; полоскание рта - активная (щетка в это время пассивна); очистка щетки, установка на зарядку - активная короткая; ожидание до следующей чистки - длительная пассивная стадия (щетка хранится на стойке). 4) Стейкхолдеры: на стадиях использования - пользователь (роль: чистящий зубы); на стадии обучения ребенка - ребенок (непосредственно чистит, ученик) и родитель (наставник, контролирует); на стадии зарядки - электросеть (внешняя система); на этапе производства щетки - завод, инженеры; на этапе продажи - ритейлеры; утилизация - служба переработки. 5) Структура:

функционально щетка = чистящий модуль + модуль управления + коммуникационный модуль (передает данные на смартфон) + питание; конструктивно = головка с щетинками + ручка с мотором + батарея + Bluetooth-чип; на этапе производства разложение на компоненты для сборки (корпус, двигатель, плата, аккумулятор, щетинки); на этапе обслуживания - заменяемые части (сменная насадка, сменный аккумулятор). Собрал все это, разработчик готов перейти к построению модели - о чем следующий раздел.

3.4. Этап построения моделей

Имея на руках необходимую информацию, эксперт приступает к построению расширенного системного оператора. Как отмечалось, визуальную модель можно представить в виде *трехмерного куба*, где одна ось - время (прошлое, настоящее, будущее), вторая - системные уровни (подсистема, система, надсистема), а третья - стадии жизненного цикла продукта. Практически же “куб” реализуется как набор взаимосвязанных матриц или таблиц. Построение модели включает следующие шаги:

- **Выбор формы представления.** В зависимости от удобства, это может быть: большая таблица, разбитая на блоки по стадиям ЖЦ (например, отдельная таблица 3×3 для каждой крупной стадии); либо объемная схема на трех координатных осях (например, на бумаге или электронной доске рисуются три взаимно перпендикулярных оси, и информация вписывается в образовавшиеся ячейки); либо другая схема, позволяющая однозначно отразить три измерения.

В главе 2 (рис. 2.5) предложено именно *слоистое* представление: классическая матрица системного оператора “умножается” на несколько слоев по оси жизненного цикла. Мы будем исходить из этого способа: то есть фактически строим серию 9-экранных матриц - по одной на каждую ключевую стадию ЖЦ. Эти матрицы образуют “стопку”, где нижний слой - ранние стадии (например, концепция), верхний - поздние (утилизация), а между ними - разработка, производство, внедрение, эксплуатация и др.

При этом между соответствующими ячейками слоев есть логические связи (например, экран “надсистема-настоящее” на стадии разработки связан с экраном “надсистема-настоящее” на стадии эксплуатации, и т.д., поскольку речь идет об одном и том же уровне-времени, но в разных стадиях продукта). Можно сказать, мы строим *многомерную таблицу*, которую двумерно изобразить полностью сложно, но мыслительно представляем ее как куб.

- **Заполнение оси жизненного цикла.** Сначала определяется набор стадий ЖЦ, которые будем использовать как “слои” модели. Этот набор уже должен быть понятен из этапа сбора данных: мы перечислили жизненный цикл

продукта (например: Концепция → Проектирование → Производство → Внедрение → Эксплуатация → Обслуживание → Утилизация; или укрупненно: Разработка → Производство → Использование → Вывод из эксплуатации). Здесь важно выбрать оптимальную дробность - не слишком много и не слишком мало этапов. Обычно 5-7 фаз достаточно, чтобы охватить основные отличающиеся состояния продукта (см. раздел 2.3.4, где приведен пример стандартных стадий и типичных участников).

Например, для технического продукта: Инициирование (Концепт) - Разработка (Проектирование и тестирование) - Производство - Маркетинг и сбыт (внедрение) - Эксплуатация (у потребителя) - Поддержка/обслуживание - Утилизация. Для сервисного продукта (например, доставка пиццы): Дизайн сервиса - Запуск на рынке - Операционное функционирование - Расширение/модернизация - Завершение услуги.

Выбранные стадии упорядочиваются хронологически. В расширенном операторе они выступают как “пласты” анализа, между которыми можно “перемещаться”, меняя контекст. На практике, начиная построение, часто удобно взять одну стадию (например, эксплуатацию) как отправную - именно она обычно хорошо известна - и затем расширять модель вверх и вниз, добавляя предшествующие и последующие этапы.

- **Построение классических срезов (матриц) по стадиям.** Для каждой стадии ЖЦ создается своя мини-модель по осям “время-иерархия”. Проще всего начать с **центрального слоя** по времени - настоящего времени для данной стадии. Например, возьмем стадию “эксплуатация” и построим классическую 3×3 матрицу: в центре - система на этой стадии в настоящем (как продукт функционирует при использовании сейчас), слева - система на этой же стадии в прошлом (как продукт выполнял функцию в прошлых версиях на этапе эксплуатации), справа - система на этой стадии в будущем (как, предположительно, продукт будет выполнять функцию на стадии эксплуатации в будущем).

В центральном столбце этой матрицы: подсистема-настоящее (что из себя представляет устройство продукта на данной стадии сейчас - по сути, структура подсистем, выделенная для этой стадии), система-настоящее (сам продукт в работе, описание текущего функционирования), надсистема-настоящее (окружение и пользователь на этой стадии сейчас).

Аналогично заполняются строки для подсистемы и надсистемы в прошлом и будущем. Затем переходят к следующей стадии, скажем “производство”. Там матрица трактуется так: рассматриваем продукт на этапе производства. Центр - система на стадии производства в настоящем (т.е. текущий процесс производства продукта), левый столбец - как производился продукт в

прошлом, правый - прогноз производства в будущем. Иерархические уровни: подсистема - это, например, компонентная база, сырье, линии (то, из чего складывается процесс на производстве); надсистема - производственное окружение (завод, снабжение, рабочая сила, технологии отрасли). Таким образом, для каждой стадии получается своя 3×3 сетка.

Фактически мы размножаем классический системный оператор *по стадиям*. Важно подчеркнуть: **все стадии связаны одной логикой продукта**, это не отдельные независимые матрицы. Между матрицами по вертикали (по оси ЖЦ) есть связи преемственности. Например, экран “система-настоящее” на стадии Разработка будет предшествовать по жизни экрану “система-настоящее” на стадии Производство (проще говоря: то, что спроектировано, далее поступает в производство), а экран “система-настоящее” стадии Производство предшествует “система-настоящее” стадии Эксплуатация (произведенное поступает к потребителю). Эти связи неявно присутствуют и обеспечивают целостность модели во времени. При построении их надо отслеживать: информация, вписанная в одно «окно», может перекликаться с соседним по оси ЖЦ. Например, если на стадии эксплуатации (настоящее) отмечено, что продукт требует частой подзарядки аккумулятора, то на стадии разработки (прошлое относительно эксплуатации) можно добавить соответствующую причину (недостаточно емкий аккумулятор был заложен) или на стадии обслуживания (будущее относительно эксплуатации) - следствие (необходимость частой замены батареи на сервисе). Таким образом, заполняя модель, эксперт двигается не только внутри одной матрицы, но и между матрицами разных стадий, обеспечивая логическое соответствие.

- **Отражение сценариев и микро-стадий.** Как упоминалось, расширенный оператор учитывает множественность сценариев использования продукта. Это реализуется следующим образом: на **стадии эксплуатации** (и потенциально на других стадиях, где тоже могут быть разные сценарии взаимодействия, например на этапе внедрения могут быть разные сценарии вывода на рынок) каждая ячейка может иметь несколько вариантов - по числу основных сценариев.

Проще говоря, внутри экрана “надсистема-настоящее (эксплуатация)” мы рассматриваем *несколько надсистем*, соответствующих разным сценариям (разным контекстам применения). Визуально это можно изобразить как дробление ячейки на под-ячейки или просто перечисление альтернатив. Например, надсистема-настоящее (эксплуатация) для смартфона: сценарий 1 - пользователь как телефонный абонент (контекст связи), сценарий 2 - пользователь как фотограф (контекст съемки), сценарий 3 - пользователь как слушатель музыки (контекст развлечения) и т.п. (см. раздел 2.2.1 о множестве сценариев). Аналогично для подсистемы-настоящее (эксплуатация): в разных сценариях продукт может задействовать разные комбинации своих модулей.

Микро-стадии же представляют детальную динамику внутри стадии. Их удобно отображать вдоль оси времени на конкретной стадии. Иногда их изображают как дополнительный уровень детализации временных экранов: например, вместо одного экрана “система-настоящее (эксплуатация)” рисуют несколько последовательных под-кадров, соответствующих ключевым микро-стадиям сценария. Однако слишком детально расписывать всю последовательность внутри модели может быть избыточно.

Практичнее следующим образом: зафиксировать наличие *активных* и *пассивных* микро-стадий и отмечать особенности системы на них. Например, на стадии эксплуатации (настоящее) для сценария “смартфон - звонок” можно указать: активная микро-стадия - разговор (телефон активен, экран выключен, работает аудио тракт), пассивная - ожидание между звонками (телефон на сети, фоновые процессы идут). Для каждой такой микро-стадии можно вписать отдельные заметки или параметры, особенно если в пассивной стадии происходят скрытые действия (см. ниже про ресурсы). В некоторых вариантах представления микро-стадии всех сценариев изображают отдельной диаграммой (например, *временная шкала использования*, как в разделе 2.2.2, рис. 2.3), а в системном операторе делают на нее ссылки. Выбор зависит от удобства: главное, чтобы информация о существовании разных эпизодов использования была интегрирована в модель и могла учитываться при анализе.

- **Учёт ролей стейкхолдеров.** Для каждого экрана, описывающего надсистему (контекст), добавляется указание, **какой стейкхолдер** рассматривается. Проще всего это делать непосредственно в названии или примечании к ячейке. Например: “Надсистема-настоящее (стадия эксплуатации, сценарий 1): Пользователь (роль: оператор смартфона)” или “Надсистема-настоящее (стадия производства): Завод, цех сборки”.

Таким образом, каждое окно, где фигурирует взаимодействие с окружением, фактически помечается конкретной ролью/актером. Это важно для дальнейшего анализа: глядя на модель, специалист сразу видит, **чьи интересы и влияния** учтены в данном элементе. Кроме того, роли могут быть отражены и на уровне системы или подсистем, если на них тоже влияют внешние участники. Например, экран “система-настоящее (внедрение)” - можно добавить: система = продукт на полке магазина, стейкхолдер = продавец (как оператор по демонстрации товара).

Визуально учет ролей может быть оформлен через цветовую кодировку (разные цвета или иконки для разных стейкхолдеров) или текстовые пометки. В примере сервиса доставки пиццы (раздел 2.2.3) была приведена схема связи микро-стадий и ролей пользователей (рис. 2.4); при построении оператора продукта эти связи встраиваются прямо в соответствующие экраны.

- **Вариативная структура подсистем.** На этапе заполнения модели необходимо использовать данные о конфигурациях системы (из п. 3.3.5) для отражения *разных декомпозиций* продукта по стадиям. Практически это значит: в экране “подсистема-настоящее” для каждой стадии ЖЦ описывается состав подсистем с учетом особенностей этой стадии.

Например, на **стадии разработки:** подсистемы-настоящее - перечисляются основные функциональные модули концепта (что видит конструктор; возможно, это абстрактные блоки типа “модуль А отвечает за функцию X, модуль В за функцию Y”). На **стадии производства:** подсистемы-настоящее - указаны узлы и компоненты, из которых продукт собирается (например, для смартфона: модуль дисплея, модуль камеры, плата, батарея, корпус - то, как они идут по сборке). На **стадии эксплуатации:** подсистемы-настоящее - перечислены элементы, определяющие пользовательские свойства (экран, аккумулятор, ПО, услуги связи и т.д., то есть то, на что смотрит пользователь и что влияет на опыт). На **стадии обслуживания:** подсистемы - элементы, подлежащие обслуживанию (сменная батарея, обновляемое ПО, сменные щетки у электрической щетки и пр.). На **стадии утилизации:** подсистемы - компоненты по материалам (пластиковый корпус, литиевый аккумулятор, электронная плата - это важно для переработки). Все эти представления - разные срезы одного продукта, поэтому между ними есть соответствия (например, модуль дисплея на стадии производства соответствует экрану на стадии эксплуатации, и т.д.). Отмечая такие соответствия, мы фиксируем эволюцию подсистем: как элемент трансформируется или проявляет себя на разных стадиях.

Сформировав таким образом модель, эксперт получает объемную картину, где каждая ячейка можно рассматриваться как отдельное “окно” ситуации: (стадия ЖЦ, уровень системы, временной срез). Всего окон получается больше, чем 9 - в зависимости от числа стадий, их может быть $3 \times 3 \times N$, где N - количество стадий (например, при 6 стадиях ЖЦ будет $3 \times 3 \times 6 = 54$ позиции в кубе). На практике *не все ячейки будут содержательно заполнены* - где-то информация может быть несущественной или отсутствовать. Это нормально: оператор не требует заполнения абсолютно всех комбинаций. Главное, чтобы каждая значимая комбинация была рассмотрена, и ни одна важная не была забыта.

Полученный расширенный системный оператор обычно фиксируется в некотором артефакте: таблице, диаграмме или программе. Возможно использование специализированных шаблонов или софта, но можно и просто с помощью текстового документа и рисунков. Важно, чтобы модель была **наглядной и удобной для чтения командой**. Часто ее представляют в виде набора иллюстраций в отчете (см. примеры в главе 2, разделы 2.4.4 и 2.5.4,

где рассматриваются эволюция смартфона и сервиса доставки пиццы с учетом оси ЖЦ и мульти-сценарности). В тех примерах, по сути, описаны части расширенного системного оператора: сравниваются прошлое, настоящее, будущее для разных стадий, раскрываются микро-стадии и роли.

Следует добавить, что построение настолько комплексной модели - итеративный процесс. Часто после первоначального заполнения выявляются пробелы или противоречия, требующие дополнительно собрать данные или переработать сценарии. Модель можно постепенно уточнять, обсуждая в междисциплинарной группе: совместное рассмотрение “куба” стимулирует вопросы вроде “а что происходит с компонентом X на этапе Y?” или “кто отвечает за функцию Z после запуска продукта?”. Таким образом, формирование модели не только документирует знания о продукте, но и выявляет зоны неопределенности, которые нужно прояснить. Этот процесс сам по себе ценен для команды разработки, так как повышает общее системное понимание продукта.

Итак, результатом этапа 3.4 является **многомерная модель системного оператора продукта**, охватывающая: (а) *эволюцию во времени* (прошлое-настоящее-будущее), (b) *иерархические уровни системы* (подсистемы, сама система, надсистема), (с) *стадии жизненного цикла* (от зарождения до окончания жизни продукта), а также дополнительно учитывающая (d) *разные сценарии применения* и *микро-стадии* внутри эксплуатации и (e) *разных стейкхолдеров* на всех фазах. Такая модель существенно богаче классической 9-экранной схемы. Она предоставляет “единое пространство” для анализа множества аспектов продукта. В следующем разделе 3.5 мы рассмотрим, как практически использовать этот расширенный оператор для решения различных задач - от поиска проблем до генерации новых идей.

3.5. Особенности применения расширенного системного оператора

Методика расширенного системного оператора продукта открывает перед исследователем и разработчиком широкие возможности анализа. Благодаря заложенной многомерности, модель можно применять для разных целей: выявление проблем и противоречий в существующем продукте, анализ интересов стейкхолдеров, поиск скрытых ресурсов для улучшений, поддержка разработки нового продукта и даже прогнозирование его эволюции. В данной главе сосредоточимся на трех прикладных аспектах (3.5.1-3.5.3), связанных с действующим продуктом, его проблемами, участниками и ресурсами. Ниже описаны особенности использования расширенного СО для решения соответствующих задач, с примерами на различных продуктах.

3.5.1. Поиск проблем и противоречий в существующем продукте

Одна из первичных задач, ради которых создается системный оператор, - *обнаружение проблем* в анализируемой системе. Расширенный оператор значительно повышает эффективность этого процесса, позволяя выявлять **противоречия на пересечении различных измерений** - там, где классический анализ их мог упустить.

Методика поиска проблем через расширенный СО: Эксперт “просматривает” полученный куб модели, сравнивая содержимое разных окон и переходя по осям, задавая вопросы: «Какие несоответствия видны между требованием на одном уровне и реализацией на другом?», «Не противоречат ли друг другу требования разных стейкхолдеров на одной стадии?», «Не возникает ли конфликт между оптимумом на стадии А и стадией В?» и т.д. Особое внимание уделяется следующим ситуациям:

1) Противоречия между стадиями жизненного цикла. Сопоставляя соседние слои модели, можно найти случаи, когда решение, оптимальное для одной фазы, создает проблемы на другой. Например, материал корпуса смартфона, облегчающий производство (пластик - дешево и просто в изготовлении), может ухудшать пользовательские свойства на этапе эксплуатации (низкая прочность, ощущается менее премиально) - конфликт между требованиями производства и маркетинга/эксплуатации. Или, как отмечалось ранее, конструктивное решение, отлично работающее в прототипе, может оказаться непригодным в массовом производстве (сложно тиражировать, высокие допуски). Расширенный оператор буквально заставляет проверять каждую стадию: пролистав модель вдоль оси ЖЦ, команда может выписать перечень несоответствий между этапами.

2) Противоречия в рамках эксплуатации (между сценариями и микро-стадиями). Множественность сценариев использования означает, что продукт должен удовлетворять порой противоположные потребности в разных ситуациях.

Классический пример: смартфон при активном использовании (разговор, работа с приложением) - пользователь хочет большой экран для удобства, а в пассивной стадии (ношение в кармане) - хочет небольшой компактный размер. Это физическое противоречие (противоречие свойства), проявляющееся между двумя микро-стадиями эксплуатации одного продукта: “большой vs. маленький” размер, требуемый в разных условиях. Расширенный оператор сразу выявляет его, так как мы описываем отдельно стадию “использование (активно)” и “хранение/перенос (пассивно)” и видим разные требования к системе. В данном случае противоречие уже решается на рынке появлением складных смартфонов -

техническое решение, фактически позволяющее устройству менять формат под сценарием (компромисс: большой экран раскладывается для активного использования, но складывается для ношения).

Другой пример: сервис доставки пиццы - в активной стадии “доставка” курьеру важно ехать максимально быстро, а в параллельной активности “сохранение качества” пицце важно минимизировать тряску и колебания температуры. Возникает конфликт скорости и качества. Анализ микро-стадий показал эту проблему, и современные сервисы решают ее введением новых ресурсов: термоконтейнеры (уменьшают охлаждение, сглаживают противоречие “быстро vs. качественно”) и трекаеры (сообщают клиенту время прибытия, снижая негатив от ожидания).

Таким образом, **проход по микро-стадиям сценария с фиксацией параметров** на каждой позволяет увидеть места, где требования одной стадии прямо противоположны требованиям другой. Классический системный оператор, рассматривающий усредненно “стадию эксплуатации” без деталей, мог бы этого не показать, а расширенный - выявляет автоматически.

3) Конфликты между разными стейкхолдерами. Благодаря тому, что модель помечена ролями, можно систематически анализировать, нет ли противоречий между целями и потребностями различных участников.

Например, при взгляде на стадию эксплуатации: конечный пользователь хочет максимальной простоты продукта, а обслуживающий персонал (например, IT-администратор или механик) - максимальной оснащенности средствами диагностики и настройки. Если на экране “надсистема-настоящее (эксплуатация)” обозначены обе эти роли, мы уже можем формулировать противоречие: “простота vs. управляемость”. Или, на стадии маркетинга: маркетолог желает добавить больше функций для привлечения клиентов, а инженер производства - минимизировать разнообразие конфигураций для удешевления. Такого рода межрольевые противоречия часто всплывают в проектах, но не всегда явны.

Модель СО позволяет их выписать, пройдясь по каждой стадии: **сравнить требования стейкхолдеров** на этой стадии. Если модель строилась коллективно, то на этапе создания она уже вскрывает такие моменты (когда представители разных функций обсуждают единый “куб”). Если же модель составлена аналитиком, он может постфактум проверить: нет ли в заполненных ячейках взаимоисключающих или конфликтных утверждений.

Например, в надсистеме-настоящее для стадии “разработка” указано: инвестор требует сократить время разработки до минимума, а рядом (может, в системе-настоящее или надсистеме-настоящее следующей стадии) - указано, что для качественного производства нужна длительная

отработка технологии. Это сигнал противоречия “быстрота вывода продукта vs. технологическая тщательность”.

4) Проблемы невидимые при одностадийном рассмотрении. Расширенный оператор помогает также идентифицировать упущенные проблемы.

Например, зачастую основные усилия сосредоточены на эксплуатации продукта, а проблемы, связанные с его утилизацией, остаются вне поля зрения. Добавив стадию “утилизация” и проанализировав ее, можно вдруг обнаружить проблему экологического характера: “в продукте используются несъемные батареи, что затрудняет переработку” - этот нюанс мог быть проигнорирован без рассмотрения стадии утилизации. Или, рассмотрев стадию “внедрение на рынок”, можно выявить, что жизненный цикл клиента (customer journey) слишком длительный и сложный, что есть “бутылочные горлышки” при переходе от покупки к использованию.

Такие проблемы часто выходят за рамки чисто технической системы, их можно назвать *системно-ориентированными*: они возникают на стыке техники, пользователя и процесса.

Расширенный оператор, охватывая весь контекст, позволяет методично пройти по этим стыкам и задать вопрос: *нет ли здесь проблемы?* Как отмечает ТРИЗ, “проблема не должна рассматриваться изолированно” - нужно видеть ее связи. Наш подход расширяет диапазон связей, тем самым вероятность “вскрыть” скрытую проблему гораздо выше.

5) Документирование найденных проблем: Практически выводом работы по анализу модели становится список проблем, противоречий или узких мест с указанием, где именно в модели они обнаружены. Например:

- Противоречие А: обнаружено между стадией X и Y, суть - требования такого-то стейкхолдера vs. техническая реализация.
- Проблема В: на микро-стадии M обнаружен простой (пассивное время), ведущий к потерям ресурса R.
- Проблема С: стейкхолдер U не получает на стадии Z нужного ему эффекта (например, роль клиента на стадии обратной связи - низкая вовлеченность, что ведет к нехватке данных для улучшения продукта). Каждый пункт списка можно “привязать” к элементу системного оператора - тем самым обосновывая, откуда взято это наблюдение.

*Пример (поиск противоречия смартфона): Возьмем простой конфликт смартфона, уже упомянутый - между размером экрана и габаритами устройства. В расширенном СО он проявляется так: стадия эксплуатации, надсистема-настоящее, сценарий активного использования - пользователь хочет **большой экран** для удобства; там же, надсистема-настоящее, сценарий пассивного ношения - пользователь хочет **маленький телефон**, чтобы помещался в карман. Модель фиксирует оба факта, и при сравнении получается противоречие. Классический оператор, скорее всего, описал бы просто “пользователь хочет компактный, удобный телефон” и “пользователь хочет функциональный, с большим экраном”, но не разнес их по разным временам использования - потенциально заметил бы конфликт, но не так явно. Расширенный же явно показывает, в какие моменты проявляются эти взаимоисключающие требования.*

*Пример (поиск проблемы сервиса): В модели сервиса доставки пиццы, как показывали примеры в главе 2, пассивная микро-стадия “ожидание клиента” выявляет проблему остывания продукта. В операторе это видно при анализе пассивных стадий: стадия “доставка/эксплуатация услуги”, система-настоящее, пассивная фаза: пицца находится в пути без активных действий, но происходит **потеря тепла** - ресурс (горячая температура) тратится впустую. Это сразу дает наводку: нужно задействовать ресурс (тепло) или компенсировать его потерю. Решение - использовать изолирующую коробку или подогрев - фактически устраняет проблему. Благодаря расширенному СО проблема была формализована как: “пассивная стадия транспортировки ведет к ухудшению потребительских свойств (остывшая пицца)” - что является важной формулировкой для инженеров сервиса.*

Таким образом, расширенный системный оператор служит своего рода *детектором противоречий*. Обход “куба” по разным траекториям (между уровнями, стадиями, сценариями) обеспечивает систематическое выявление конфликтов. Как подчеркивает Г.С. Альтшуллер, точная формулировка противоречия - залог эффективного решения в ТРИЗ. С помощью расширенной модели мы получаем более полный список противоречий, многие из которых раньше могли оставаться неочевидными. Это повышает качество исходного анализа: найденные противоречия затем можно устранить известными способами (приемами ТРИЗ, поиском компромиссов или инновационных решений). Можно сказать, что расширенный оператор делает шаг “выявление проблем” практически автоматизированным: структура модели направляет внимание на все проблемные точки.

Данная особенность подтверждается опытом: например, в проекте с кофемашинами (раздел 2.4 главы 2) анализ по оси ЖЦ помог сменить фокус проблемы на оптимальную стадию, что привело к решению без больших

издержек. В общем случае, сначала находим противоречия - потом их решаем, и в этом процессе расширенный СО незаменим как генератор гипотез о “слабых звеньях” продукта.

3.5.3. Анализ стейкхолдеров и ролей

Расширенный системный оператор, как мы показали, интегрирует в модель разнообразных стейкхолдеров продукта. Это дает возможность провести глубокий **анализ стейкхолдеров**: понять, как каждая категория участников взаимодействует с продуктом, какую ценность получает, какие испытывает затруднения, и как их роль может меняться на протяжении жизненного цикла.

Подход к анализу стейкхолдеров через СО: Здесь удобно “идти” вдоль оси ЖЦ, фокусируясь на надсистемных экранах, где и фигурируют стейкхолдеры на каждой стадии. Фактически, расширенный СО позволяет выстроить **карту стейкхолдеров и ролей по жизненному циклу**. Например, составив модель, можно выписать по стадиям:

- Концепция: стейкхолдеры - заказчик, инвестор, аналитик рынка.
- Разработка: стейкхолдеры - инженер, дизайнер, тестировщик, потенциальный пользователь (через исследования).
- Производство: стейкхолдеры - технолог, рабочий сборочной линии, снабженец, QC-инженер.
- Внедрение: стейкхолдеры - маркетолог, продавец, дистрибьютор, клиент-покупатель.
- Эксплуатация: стейкхолдеры - конечный пользователь (различные роли), техническая поддержка, оператор инфраструктуры (например, мобильный оператор для смартфона).
- Обслуживание: стейкхолдеры - сервисный инженер, пользователь (в роли клиента сервиса).
- Утилизация: стейкхолдеры - перерабатывающая компания, возможно сам пользователь (который сдает или выбрасывает изделие).

Такой перечень дает целостное представление: **кто вовлечен в “жизнь” продукта от начала до конца**. Уже это поучительно, так как часто проектные команды склонны концентрироваться лишь на “покупателе” и “пользователе”,

забывая про других акторов (например, про сервисные службы или про конечное утилизационное звено).

Роли и их смена: Особенно интересный аспект - проследить *трансформацию ролей одного и того же субъекта* во времени. Расширенный СО фиксирует, что *на разных стадиях один участник может выступать в разных амплуа*.

Например, конечный пользователь:

- На стадии выбора и покупки - он **клиент**, принимающий решение (важна информация, бренд, цена).
- На стадии начального освоения - он **новичок**, которому нужна поддержка (важны инструкции, простота).
- В процессе эксплуатации - **опытный пользователь**, требующий удобства и надежности в длительном пользовании.
- При сбоях - **расстроенный клиент**, взаимодействующий с техподдержкой.
- В конце - **утилизатор**, решающий, что делать с отработавшим ресурс устройством (сдать, выбросить, продать б/у).

Все это один человек, но *разные роли на разных этапах*. Модель явным образом позволяет задать вопрос: удовлетворяется ли потребность пользователя в каждой из этих ролей? Не меняются ли его приоритеты? Например, что ценил клиент при покупке (дизайн, инновационность), и что ценит через год использования (надежность, совместимость) - есть ли тут разрыв? Анализ модели может выявить, что **ценностное предложение** продукта покрывает начало, но не покрывает хорошо продолжение (или наоборот). Это ценно для управления продуктом: мы видим, где может снижаться лояльность или где нужны дополнительные функции. В маркетинге подобную вещь изучают через понятие *Customer Journey* и *опыта клиента во времени* - наш подход дает систематический инструмент это отобразить.

Другой пример - **смена ролей между разными людьми**: кто-то передает эстафету ответственности за продукт. Скажем, на этапе внедрения ведущую роль имеет отдел маркетинга, а на этапе эксплуатации - отдел поддержки. Модель показывает точки передачи: например, *стадия продажи* - пересечение, где маркетолог передает клиента службе поддержки. Выявляя эти моменты, можно улучшить взаимодействие подразделений (что часто

страдает в реальности). Таким образом, оператор способствует *согласованию действий разных стейкхолдеров* во времени.

Применение для выявления требований стейкхолдеров: Каждый стейкхолдер предъявляет свои **требования и критерии успеха**. Разложив по модели, мы можем пройтись и собрать в одном месте: вот требования инвестора (окупаемость, ROI, срок вывода); вот требования пользователя (функциональность, удобство, цена); вот требования обслуживающего персонала (удобство ремонта, диагностики); требования регуляторов (соответствие стандартам на стадии производства и внедрения) и т.д. Такой многопрофильный сбор соответствует принципам системной инженерии - удовлетворить всех заинтересованных лиц - но часто требования собираются текстово и неструктурированно. Здесь же, благодаря операторам, они распределены по контекстам. Это позволяет видеть **конфликтующие и согласующиеся требования** (что мы обсуждали в предыдущем разделе) и искать **точки синергии**. Например, может оказаться, что введение определенной функции одновременно улучшит опыт пользователя и снизит нагрузку на службу поддержки - тогда это “легкая победа” (quick win). Или, напротив, некоторые стейкхолдеры пока “не охвачены” решениями - например, нет явно функций под нужды утилизатора (разборный ли корпус? перерабатываемые ли материалы?). Расширенный СО подсвечивает такие пробелы.

Примеры ролей на микро-стадиях: Не только по стадиям ЖЦ, но и внутри эксплуатации можно рассмотреть ролевую картину. Возвращаясь к *сервису доставки пиццы*: модель микро-стадий (раздел 2.2.3) показывает, что в **разных шагах участвуют разные люди**. На этапе заказа - сам клиент (в роли заказчика), на этапе приготовления - повар (исполнитель), на этапе доставки - курьер, на этапе получения - снова клиент (но уже в роли получателя), на этапе обратной связи - клиент (в роли рецензента) + менеджер, обрабатывающий отзыв. Таким образом, продукт-сервис “взаимодействует” поочередно с целой цепочкой людей. Это важно осознать: ценность создается совместно всеми этими участниками. Анализ показывает, где происходят **смены ответственности** (от клиента к компании и обратно), где могут быть **разрывы** (например, клиент не знает, что происходит в процессе готовки - вводится трекинг; компания не знает впечатления клиента после доставки - вводится сбор отзывов). Подобный анализ ролевых переходов позволил, например, в онлайн-сервисах бронирования путешествий выявить феномен, что клиент сам берет на себя роль турагента (делая всю работу по подбору и бронированию) - и это стало основой для многих бизнес-моделей (см. «эффект самообслуживания»). В нашем подходе такая трансформация роли пользователя из пассивного получателя услуги в активного участника видна на модели (на стадии заказа туристического продукта надсистема - клиент в роли турагента, тогда как раньше надсистемой была туристическая фирма). Эта

смена ролей со временем - интересный вывод анализа стейкхолдеров, на который указывают источники [31] и [35].

Практическое использование результатов анализа: После анализа стейкхолдеров на основе СО, команда продукта получает:

- Полный перечень стейкхолдеров с их ролями на каждой фазе.
- Понимание эволюции участия: кто подключается, кто исчезает, кто меняет роль.
- Выявленные конфликты и несовместимости интересов (обсуждалось ранее).
- Идеи для улучшений: как лучше удовлетворить каждого актора, где добавить функционал или коммуникацию. Например, после такого анализа может родиться решение “разработать обучающий модуль для новых пользователей, чтобы переход из роли покупателя в роль опытного пользователя был гладким” или “создать инструментарий для сервис-инженеров, уменьшающий их трудозатраты, не влияя на пользователя напрямую, но повышая качество обслуживания”.

Пример (анализ ролей смартфона): Проследим пользователя смартфона. На стадии покупки - покупатель: его интересует цена, репутация бренда, характеристики. На стадии начала использования - новый пользователь: критичны удобство интерфейса, инструкции. На стадии регулярной эксплуатации - опытный пользователь: важные обновления ПО, совместимость, доступ к экосистеме сервисов. При поломке - клиент сервиса: важна гарантия, ремонтпригодность устройства. Расширенный СО позволит нам убедиться, что в архитектуре продукта и в стратегии учтены все эти роли. Например, производитель может решить: “мы включим простое руководство в коробку” (для новой роли), “мы разработаем программу трейд-ин для утилизации” (для роли утилизатора), “сделаем модульную конструкцию для удобства ремонта” (для сервис-инженеров), и т.д. Здесь видно, как системный оператор трансформируется в инструмент стратегического планирования продукта, обеспечивая ориентацию на всех заинтересованных лиц, а не только на абстрактного среднего пользователя.

Пример (смена ролей в сервисе): Услуга каршеринга (краткосрочная аренда авто) - на стадии регистрации пользователь - клиент (вводит данные, компания - надсистема, проверяет); при использовании машины - пользователь становится водителем (фактически оператор системы

“автомобиль”), одновременно компания выступает как инфраструктура (следит за поездкой дистанционно); при завершении поездки - пользователь снова клиент (оплачивает, дает оценку), а автомобиль идет к следующему арендатору. Анализ модели показал бы, например, что клиент берет часть роли прокатного агента на себя (он сам бронирует, открывает машину через приложение), что снижает издержки компании, но требует удобного интерфейса. Таким образом, роль клиента динамична, и продукт должен быть спроектирован под эту динамику (например, приложение - как часть продукта, помогающая клиенту в роли агента).

Подводя итог, расширенный системный оператор дает структурированный подход к учету стейкхолдеров. Он превращает абстрактный “анализ заинтересованных сторон” в конкретное заполнение ячеек модели и их рассмотрение. Это обеспечивает полноту (ни один участник не забыт) и глубину (учтены изменения ролей и контекстов). Результат - продуктовая команда получает многогранное представление о том, для кого и как они создают ценность. В современном продуктовом менеджменте, ориентированном на пользователя (*customer-centricity*), подобный инструмент особенно полезен. Он расширяет карту путешествия пользователя (*customer journey*) до карты путешествия всех стейкхолдеров (*all-stakeholder journey*), что важно для комплексных систем и сервисов.

3.5.4. Анализ доступных ресурсов

В ТРИЗ важное место занимает понятие **ресурсов** системы - всего, что уже имеется или происходит в системе и её окружении, и что можно использовать для решения проблемы без значительных затрат. Расширенный системный оператор существенно облегчает *поиск скрытых ресурсов*, особенно тех, что распределены во времени или между подсистемами.

Ресурсы на пассивных стадиях: Одним из наиболее ценных аспектов расширения модели является явное отражение пассивных (неактивных) микро-стадий и фаз ЖЦ. Именно там зачастую скрыты неиспользуемые ресурсы. Согласно одному из законов развития техники - *закону согласования ритмики* - эффективно работающая система стремится к тому, чтобы все её части работали согласованно, без длительных простоев. Долгие периоды бездействия - это указание на резерв: можно либо задействовать систему в это время, либо убрать необходимость простоя. Расширенный СО, показывая эти паузы (например, время ожидания между использованием, время хранения, транспортировки, простои между операциями), позволяет систематически задать вопрос: *что полезного можно сделать в эту паузу?*

Примеры:

- Для зубной щетки мы видим длительную пассивную стадию хранения между утренней и вечерней чисткой. Это ресурс времени, когда щетка просто лежит. Решение: использовать это время для самоочистки щетки (например, встроить ультрафиолетовый дезинфектор в стаканчик-хранитель) - щетка “работает”, пока не используется, улучшая гигиену. Такие продукты существуют, и их идея фактически следует из рассмотрения пассивной стадии как ресурса.
- Для смартфона модель показывает, что устройство даже в пассивном режиме (режим ожидания) подключено к сети и обладает вычислительной мощностью. Современные смартфоны активно используют эти “фоны”: выполняют обновления, синхронизацию, сбор данных сенсоров даже когда пользователь не взаимодействует. Тем самым ресурсы пассивных периодов (время, подключение, сенсоры) превращаются в ценность - пользователь всегда получает актуальную информацию, а производитель - данные об использовании. Ранние мобильные телефоны этого не делали (пассивные стадии были “потеряны”), а сейчас благодаря более развитому пониманию системы эти ресурсы задействованы. В контексте оператора мы явно видим: пассивная стадия “ночь, телефон на прикроватной тумбочке” - ресурс: время + Wi-Fi + зарядка; можно ночью выполнить резервное копирование, загрузить обновления, не беспокоя пользователя. И индустрия так и делает.
- В сервисе доставки пиццы пассивное ожидание клиента и остывание пиццы - тоже ресурс (время + температура), который частично сейчас расходуется впустую. Решения: либо сократить время (ускорить доставку), либо **управлять средой** в это время (термосумки, информация клиенту). Интересный ресурс выявлен в расширенном операторе современного сервиса: отзывы клиентов после доставки. Раньше, в традиционной доставке, этот этап отсутствовал или не учитывался (клиент получил пиццу - цикл закончен). Теперь обратная связь стала ценным ресурсом (данные о удовлетворенности), который собирается через приложение и используется для улучшения сервиса. Расширенный СО позволил включить “пассивную” для клиента стадию (написать отзыв - не обязательная активность, многие могут пропустить) в жизненный цикл продукта, тем самым компания получила новый ресурс - информацию.

Ресурсы разных подсистем и надсистем: Классический оператор призывает смотреть на подсистемы и надсистему для поиска ресурсов (например: избыточная энергия двигателя - ресурс, температура окружения - ресурс, и т.п.). Расширенный оператор усиливает этот поиск тем, что предлагает *разные разрезы подсистем*. На разных стадиях могут проявляться разные ресурсы. К примеру,

- на стадии производства можно обнаружить ресурс материалов или отходов, который потом использовать на стадии эксплуатации. В модели это может выглядеть так: на стадии утилизации отмечено, что упаковка пиццы выбрасывается, а на стадии производства - что упаковка стоит денег; возникла идея: стимулировать возврат коробок или использовать их как рекламный носитель (превратить отход в ресурс).
- Между подсистемами одного уровня: расширенный СО легко отображает, какие подсистемы в текущей эксплуатации задействованы, а какие нет. Например, в смартфоне при определенном сценарии активна только часть модулей (сказать, при телефонном звонке CPU загружен на 10%, экран выключен). Это значит, что 90% CPU - свободный ресурс, экран не горящий - ресурс энергосбережения и т.д. Такие детали можно рассмотреть, хотя оператор оперирует более концептуально. Однако сам принцип - **замечать незадействованные функции или компоненты** - может навести на идеи. В известном примере Altshuller о противогазе, который мешал солдату пить воду - ресурсом решения оказалось давление выдоха (чтобы протолкнуть воду) - классический случай использования *внутреннего ресурса* системы. Расширенный оператор тут помог бы увидеть микро-стадию “солдат хочет пить (маска надета)” - пассивная подфункция, и ресурсы внутри системы (выдыхаемый воздух).
- **Ресурсы времени и пространства:** За счет рассмотрения эволюции во времени, можно искать ресурсы *в прошлом* и *в будущем*. Например, в прошлом продукта могла существовать функция, от которой отказались, но которую можно реинкарнировать под другую задачу (исторический ресурс). Или изучая будущие окна, можно “создать” ресурс: представить идеальный конечный результат, и оттуда понять, какого ресурса не хватает сейчас и как его получить. Такой прием - “идеи из будущего” - в ТРИЗ известен, и оператор облегчает его реализацию: просто смотришь на экран “система-будущее” и рассматриваешь, какие изменения там предполагаются, чтобы уже сейчас понять, что можно использовать. Например, если в будущем автомобиля предполагается, что он будет автономным, то уже сейчас некоторые датчики автономности можно ставить для сбора данных - ресурс *будущей функции*, используемый в

настоящем.

Интеграция с поиском решений: Когда мы составили список ресурсов, их можно соотнести с проблемами (из 3.5.2). ТРИЗ-решения часто сводятся к тому, чтобы вовлечь дополнительный ресурс в разрешение противоречия. Расширенный СО, раскрывая больше ресурсов, увеличивает шанс найти безболезненное улучшение. Как отмечено в литературе, системный оператор помогает “расширить область решения задачи”, выходя за рамки исходной формулировки. Добавление оси ЖЦ еще сильнее расширяет область: смотрим не только вверх-вниз (подсистемы, надсистема) и назад-вперед (время эволюции), но и вширь по жизненному циклу и сценариям. Это фактически многократно увеличивает “поле” поиска ресурсов и идей.

Пример (ресурс на пассивной стадии смартфона): Модель показала, что смартфон в режиме ожидания подключен к электросети (во время зарядки ночью) и к интернету. Этот ресурс энергии и связи разработчики использовали для внедрения функции **ночной синхронизации**: когда телефон стоит на зарядке, он автоматически резервирует данные в облако. В результате пользователь не замечает процесса, а имеет актуальную резервную копию. В классическом анализе можно было бы тоже догадаться, но расширенный СО четко обозначил: стадия зарядки - телефон бездействует, но есть электричество и Wi-Fi, значит, ресурс налицо. Аналогично, пассивное прослушивание окружающей среды (микрофон включен даже в пассивном режиме для голосовых ассистентов) - спорный, но тоже ресурс (время простоя CPU+микрофон).

Пример (ресурс в подсистемах доставки): На стадии ожидания курьера в пицца-сервисе выявлен ресурс - **тепло самой пиццы**. Решение: сделать термоконтейнер, что по сути сохранение этого тепла. Другой ресурс - **время клиента, пока он ждет**. Решение: занять его приложением с отслеживанием доставки, или предложить ему купон на следующий заказ - то есть время ожидания превращается в маркетинговый ресурс (вовлечь клиента, чтобы не скучал). Расширенный оператор, зафиксировав пассивную стадию “ожидание”, стимулировал вопрос “что делаем с клиентом в это время?”.

Пример (закон ритмики): Если какой-то компонент системы сильно простаивает относительно других, TRIZ рекомендует либо задействовать его, либо избавиться. В промышленном примере: станок работает 1 минуту, затем 9 минут ждет (цикл 10 минут). Расширенный оператор для производственного процесса сразу выделил бы микро-стадии: работа - простой, и отметил простой как ресурс. Решения: либо добавить второй станок (чтоб пока один ждет, второй работал - синхронизация ритмов), либо модифицировать станок, чтобы

работал непрерывно. В любом случае, мы бы не пропустили такой дисбаланс, так как модель “подсвечивает” длительную пассивную стадию. В литературе [38] подобные примеры приводятся, и наш инструмент просто систематизирует их поиск.

Суммируя, анализ ресурсов с помощью расширенного СО проходит в несколько шагов:

1. отметить все пассивные элементы (во времени, в структурах, в окружении),
2. для каждого спросить “можно ли использовать/сократить это бездействие?”,
3. просмотреть все системы и подсистемы на предмет незадействованных свойств (избыточная прочность, колебания, побочные эффекты - часто ресурсы),
4. взглянуть на надсистемы - какие ресурсы среды не используются (например, температура окружающей среды, солнечный свет, взаимодействие пользователей между собой и т.д.),
5. рассмотреть историю и будущее - что было или будет доступно.

Благодаря всеобъемлющему охвату, такой анализ, как правило, приводит к длинному списку потенциальных ресурсов. Конечно, не каждый из них сразу приведет к идее, но наличие списка - это первый шаг. Далее применяются фильтры и творческие техники (например, специфицируются противоречия и подбираются приемы для их разрешения с использованием найденных ресурсов).

В итоге расширенный системный оператор способствует тому, чтобы ни один ценный ресурс не остался “спрятанным” в углу. Часто инновации рождаются из вдруг замеченного ранее игнорируемого фактора. Наш метод повышает шансы таких инсайтов. В комплексе с предыдущими разделами (поиск проблем, анализ стейкхолдеров) он обеспечивает всестороннее изучение продукта. Как показано на практике (см. главы 1-2 и работы по OTSM-ТРИЗ), многоосевой подход генерирует более богатое пространство решений, что особенно важно на этапах разработки и совершенствования современных продуктов [38].

3.6. Область применения и ограничения расширенного системного оператора продукта

Расширенный системный оператор продукта представляет собой методический инструмент системного анализа и проектирования, предназначенный для структурирования знаний о продукте как системе в нескольких взаимодополняющих проекциях (уровни системы, временные срезы, контекст/среда, жизненный цикл, ценность и стейкхолдеры, функции и качества, ограничения).

В настоящем подразделе фиксируются границы применимости РСО, типы систем и классы задач, для которых использование инструмента является целесообразным, а также ограничения и случаи, в которых применение РСО не обеспечивает ожидаемого эффекта.

3.6.1. Предпосылки и условия эффективности применения РСО

РСО является наиболее результативным при выполнении следующих условий:

1. **Наличие объекта анализа**, допускающего описание в терминах системы: определяемые границы, состав элементов, связи между элементами, взаимодействие со средой и надсистемой.
2. **Наличие временного аспекта** (изменчивость продукта во времени) либо потребность в обоснованном планировании развития: версии, релизы, стадии жизненного цикла, эволюционные траектории.
3. **Множественность стейкхолдеров** и необходимость согласования их требований, в том числе, в условиях конфликта интересов.
4. **Неопределенность постановки задачи** (или неоднозначность требований), требующая систематизации гипотез и альтернатив.
5. **Требование отчуждаемости результата**: необходимость фиксировать модель, обоснование решений и принятые допущения таким образом, чтобы их можно было передать другим участникам деятельности и воспроизвести ход рассуждений.

При отсутствии указанных условий инструмент может становиться избыточным по трудоёмкости относительно получаемого эффекта.

3.6.2. Типы систем для работы с PCO

PCO применим к следующим классам систем, распространённым в инженерной и продуктовой практике:

1. **Технические и киберфизические системы** (устройства, оборудование, IoT, медицинские приборы, промышленные комплексы), где структурирование уровней системы и взаимодействий со средой позволяет выявлять узкие места и варианты модернизации.
2. **Программные системы и цифровые продукты** (SaaS, мобильные приложения, платформы, SDK/API, сервисы на базе данных и ML/AI), для которых существенны архитектурные зависимости, интерфейсы, сценарии использования, нефункциональные требования и жизненный цикл.
3. **Информационные и социотехнические системы**, в которых технологические компоненты неразрывно связаны с человеческой деятельностью и процессами (сервисы поддержки, продажи, логистика, образование, банковские и государственные услуги).
4. **Организационно-процессные системы** (с оговорками), когда процесс можно описать как систему деятельности с фиксированными ролями участников, входами/выходами, правилами и метриками качества результата.
5. **Продукт как объект потребления и как объект управления**, при явном разведении клиентской перспективы (ценность/опыт) и управленческой перспективы производителя (планирование, развитие, эксплуатация, поддержка).

Таким образом, PCO ориентирован на системы, для которых возможно задать границы системы, выделить элементы и взаимодействия, а также зафиксировать наблюдаемые критерии качества и ограничения.

3.6.3. Типы систем, для которых применение PCO ограничено

PCO применим существенно хуже либо нецелесообразен в следующих случаях:

1. **Объекты, не допускающие операционализации критериев** (в частности, чисто художественные/эстетические артефакты), если не задана рамка потребления/эксплуатации/рынка, позволяющая перейти к измеримым

свойствам, функциям и ограничениям.

2. **Системы с принципиально нефиксируемыми границами и структурой**, когда невозможно устойчиво выделить состав, связи и управляемые воздействия (например, отдельные социальные явления, описываемые лишь на уровне метафор без доступа к данным).
3. **Ситуации критического дефицита информации**, при которых ключевые элементы и связи неизвестны и не могут быть установлены ни наблюдением, ни экспертно, ни экспериментально: в этом случае РСО формирует лишь спекулятивную карту предположений.
4. **Тривиальные и локальные объекты**, где стоимость моделирования несоизмерима выгоде (простые изменения без рисков и значимых стейкхолдеров).
5. **Задачи, требующие строгих формальных доказательств**, где РСО ошибочно используется как замена обязательным процедурам (например, сертификация, доказуемая корректность, требования функциональной безопасности). В таких задачах РСО может выступать вспомогательным средством структурирования требований и аргументации, но не источником доказательства.

3.6.4. Классы задач, решаемых с помощью РСО

РСО целесообразно применять для задач, где критичны системность рассмотрения, полнота охвата и прослеживаемость решений:

1. **Системный анализ продукта**: уточнение границ продукта и надсистемы; декомпозиция на подсистемы; выявление интерфейсов и ключевых потоков (информационных, материальных, энергетических, финансовых и др.); описание контекста эксплуатации и ограничений среды.
2. **Выявление и согласование требований**: структурирование требований по стейкхолдерам и сценариям; разрешение конфликтов требований; формирование критериев качества и метрик (включая нефункциональные требования).
3. **Генерация и оценка альтернатив**: формирование вариантов решений на разных уровнях системы и в разных временных срезах; сопоставление вариантов по ограничениям и эффектам; поддержка решений по развитию продукта (дорожная карта).

4. **Локализация проблем и построение причинных гипотез:** определение уровня возникновения проблемы (подсистема, интерфейс, надсистема, процесс); формирование проверяемых гипотез причин и планов верификации (в сочетании с методами анализа причин).
5. **Поиск ресурсов для улучшения системы:** выявление и структурирование ресурсов, которые могут быть использованы для улучшения продукта как системы. Использование РСО позволяет: (1) локализовать ресурсы по уровням «подсистема-система-надсистема» и по стадиям жизненного цикла; (2) выявить недоиспользуемые, «скрытые» и побочные ресурсы (включая «простои» системы, побочные эффекты и отходы процессов); (3) связать ресурсы с конкретными функциями, ограничениями и проблемными зонами системы; (4) сформировать набор направлений улучшений, опирающихся на имеющиеся ресурсы и минимизирующих необходимость привлечения новых.
6. **Управление знаниями и коммуникация:** создание единого представления о продукте для межфункциональных команд; фиксация обоснований (rationale) и допущений; обеспечение отчуждаемости и воспроизводимости результатов анализа.

3.6.5. Практические ограничения использования

Качество результата применения РСО существенно зависит от компетенции автора модели и дисциплины фиксации допущений. На практике выделяются следующие ограничения:

- **Риск избыточного моделирования:** чрезмерное увеличение числа осей и детализации снижает управляемость модели и удорожает обсуждение. Рекомендуется придерживаться принципа минимально достаточной детализации, расширяя модель только при наличии подтвержденной полезности.
- **Терминологическая неоднозначность:** без согласованного словаря понятий (например, трактовок «функции», «качества», «надсистемы», «жизненного цикла») снижается воспроизводимость и возрастает шум в коммуникации.
- **Ограниченная проверяемость без данных и кейсов:** практическая ценность модели возрастает при подтверждении выводов эмпирическими данными, статистикой, и кейсами; в противном случае результаты следует интерпретировать как гипотезы.

В качестве критерия целесообразности применения РСО предлагается использовать соотношение «стоимость моделирования – цена ошибки»: инструмент оправдан при высокой цене ошибок, множественности стейкхолдеров, сложности и изменчивости продукта, а также при необходимости отчуждаемого, проверяемого и прослеживаемого знания. В задачах малой сложности и низкого риска применение РСО может быть заменено более легкими средствами анализа.

3.7. Выводы по Главе 3

В этой главе была представлена методика практического применения расширенного системного оператора для продуктового анализа. Краткая версия методики применения представлена в приложении (см. Приложение 4). Рассмотрены этапы ее реализации: от сбора исходных данных (сценарии, микро-стадии, стейкхолдеры, конфигурации) до построения трехосевой модели “куба” (время × иерархия × жизненный цикл) и использования этой модели для решения прикладных задач.

Приведенные примеры (зубная щетка, смартфон, сервис доставки пиццы) демонстрируют универсальность подхода для самых разных типов продуктов - от простых физических устройств до сложных технико-сервисных систем.

Расширенный системный оператор позволяет выявлять скрытые проблемы и противоречия, анализировать интересы и роли всех вовлеченных сторон, а также обнаруживать резервы для улучшения продукта на различных этапах его жизни. По сравнению с классическим системным оператором (см. раздел 1.5) расширенная методика показывает значительно более широкие возможности применения, отвечая требованиям современного продуктового мышления (ориентация на весь жизненный цикл и пользовательскую ценность во всех аспектах).

Завершая главу 3, подчеркнем, что разработанный подход опирается как на принципы классической ТРИЗ, так и на современные практики системной инженерии и продуктового менеджмента, объединяя их в едином инструменте - расширенном системном операторе продукта [38, 24, 16].

Заключение

Основные результаты исследования

В ходе исследования **разработан расширенный системный оператор продукта**, и получены результаты, отражающие цели работы.

Во-первых, проведен аналитический обзор базовых понятий и методов (Глава 1). Выявлено, что системный оператор классической ТРИЗ (модель «9 экранов» Г. Альтшуллера) хотя и является мощным средством анализа эволюции технических систем, но не учитывает ряда специфических аспектов продукта. В частности, традиционная схема недостаточно отражает **пользовательские сценарии применения, полный жизненный цикл продукта** и разнообразные роли стейкхолдеров. Это определило необходимость разработки расширенной модели, адаптированной к продуктовой тематике.

Во-вторых, в рамках Главы 2 разработана **модель расширенного системного оператора**, учитывающая особенности продуктового подхода. Модель **дополнена новыми измерениями**, которые углубляют и расширяют анализ по сравнению с классической 9-экранной схемой.

- 1) Введена ось жизненного цикла продукта - от концепции и разработки до эксплуатации и утилизации, позволяющая рассматривать продукт на всех этапах его существования, а не только в текущем состоянии.
- 2) Учтено разнообразие условий использования: предложено понятие сценариев использования и соответствующих **микро-стадий**, на которые дробится каждая стадия эксплуатации продукта. Это даёт возможность фиксировать требования и проблемы в различных контекстах применения.
- 3) В модель явно введён фактор **различных категорий пользователей и стейкхолдеров**: для каждого сценария и этапа определяются участники (конечные пользователи, инженеры, сервисные специалисты, производитель, партнёры и др.) и их роль во взаимодействии с продуктом. Такой многосторонний учёт позволяет выявлять противоречивые требования разных заинтересованных сторон на разных этапах жизненного цикла продукта.
- 4) Реализована **вариативная декомпозиция системы**: структура продукта (набор подсистем) рассматривается как динамичная и меняющаяся от стадии к стадии, отражая актуальную конфигурацию продукта на каждой фазе - функциональную на этапе концепции, техническую на этапе разработки, компонентную на этапе производства, пользовательско-сервисную на этапе эксплуатации и т.д.. Все перечисленные расширения интегрированы в единый

подход, фактически превращающий плоскую двумерную матрицу 9 экранов в многомерный «куб» анализа продукта.

Главный результат - сформирована целостная модель системного оператора, объединяющая оси системной иерархии, технической эволюции и жизненного цикла продукта, а также учитывающая сценарное разнообразие и многообразие стейкхолдеров. Такая модель обеспечивает более полное и адекватное представление о продукте, охватывая его техническое развитие и контекст использования на всех этапах существования.

В-третьих, в **Главе 3 разработана и описана методика построения и применения** расширенного системного оператора на практике. Сформирован **четкий алгоритм шагов** для построения многомерной модели продукта, включающий: сбор исходных данных о продукте; определение границ системы и ее основной функции; выделение ключевых стадий жизненного цикла; идентификацию сценариев применения и связанных с ними микро-стадий; выделение категорий пользователей и других стейкхолдеров для каждой стадии; определение значимых подсистем на каждом этапе. Заполнение расширенного оператора по предложенной методике позволяет получить целостную картину продукта во времени, в различных условиях и с разных точек зрения.

В **Главе 3 продемонстрированы примеры практического применения** разработанного инструмента, подтверждающие его эффективность. Показано, что расширенный системный оператор пригоден для комплексного анализа существующего продукта: визуализируя его историю, текущее состояние и потенциальное развитие, можно выявить узкие места и наметить направления улучшений. Методика также позволяет системно искать проблемы и **разрешать противоречия**: сопоставление «экранов» выявляет скрытые противоречия между требованиями разных стадий жизненного цикла или разных стейкхолдеров, а также между текущим состоянием продукта и будущими потребностями рынка.

Кроме того, модель помогает учитывать интересы различных участников: сравнение представлений продукта для разных ролей (пользователя, производителя, сервис-инженера и др.) выявляет конфликты ожиданий, требующие устранения. Наконец, расширенный оператор служит средством обнаружения резервов для развития продукта - как внутренних (неиспользованные возможности подсистем, скрытые функции), так и внешних (ресурсы надсистемы, новые технологии и т.п.) на разных этапах жизненного цикла.

Главный итог заключительной главы состоит в демонстрации высокой практической ценности разработанного подхода: методика расширяет

аналитические возможности классического инструментария ТРИЗ, позволяя всесторонне проанализировать продукт, выявить множество разнообразных проблем и точек роста, учесть множественность требований участников и найти дополнительные ресурсы для развития продукта.

Значение результатов для развития Бизнес ТРИЗ

Полученные результаты обладают существенной теоретической и практической значимостью для развития Бизнес-ТРИЗ.

С теоретической точки зрения, работа обогатила методологию ТРИЗ новым инструментом и понятийным аппаратом, расширив границы ее применения. В частности, в рамках исследования введены **новые понятия**: «*минимальный продукт*», «*микро-стадия жизненного цикла*» и «*сценарий взаимодействия стейкхолдера с продуктом*». Эти концепции позволяют детально описывать продукт с позиций пользовательской ценности и контекста применения, что ранее выходило за рамки классических ТРИЗ-инструментов. Тем самым **расширен понятийный аппарат** Бизнес ТРИЗ, благодаря чему методология теперь способна оперировать категорией продукта как носителя потребительской ценности на всех стадиях его жизненного цикла, а не только абстрактной технической системой. Предложенная модель и методика интегрируют рыночные и пользовательские аспекты в систему понятий ТРИЗ, адаптируя её к задачам комплексного продуктового анализа. **Формализация подхода** в виде чёткой структурной модели и последовательности шагов повышает воспроизводимость и наукоемкость метода. Разработанный расширенный системный оператор, таким образом, вносит вклад в развитие теории решений изобретательских задач, сближая ее с реалиями современного инжиниринга, продуктового менеджмента и инноваций.

С практической точки зрения, результаты исследования открывают путь к более тесной интеграции ТРИЗ с современными продуктово-инженерными подходами. Расширенный системный оператор по своей сути объединяет элементы системной инженерии, управления жизненным циклом продукта и продуктового менеджмента с классическим инструментарием ТРИЗ. Учет полного жизненного цикла и множества стейкхолдеров согласуется с принципами системной инженерии и бережливого продуктового развития (Lean), где уже на этапе проектирования закладываются требования технологичности эксплуатации, обслуживания, утилизации и др. Включение сценариев использования и анализа пути пользователя (customer journey) роднит разработанную методику с практиками дизайн-мышления и UX-исследований, дополняя их фирменным для ТРИЗ поиском и устранением противоречий. Таким образом, предложенный подход служит **еще одним мостом между Бизнес ТРИЗ и другими дисциплинами**, обеспечивая

междисциплинарным командам единый инструмент для совместного проектирования инноваций. Специалисты по ТРИЗ, системные инженеры, продуктовые менеджеры, дизайнеры могут работать с общей «3-мерной картой» продукта, каждый в своем разрезе, но в общей координатной системе. Это повышает эффективность коммуникации при разработке сложных продуктов и ускоряет внедрение инновационных решений.

Методика документирована в виде **тиражируемого инструмента** (алгоритм и рекомендации по применению), прошедшего апробацию на примерах реальных продуктов, что облегчает её внедрение в практику. В целом, значимость результатов для Бизнес-ТРИЗ заключается в расширении области применения методологии и повышении ее ценности для высокотехнологичного бизнеса. ТРИЗ-инструментарий стал более интегрирован с процессами управления жизненным циклом продукта и формирования потребительской ценности, что способствует его распространению в сфере продуктового инновационного менеджмента.

Особо следует подчеркнуть вклад в методическое развитие **стадии Problem-Solution Fit** в инновационном цикле. Предложенный расширенный системный оператор может выступать **ядром новой методики Бизнес ТРИЗ, ориентированной на достижение соответствия продукта проблеме пользователя (Problem-Solution Fit)**. На ранних этапах создания продукта инструмент позволяет системно описать концепцию решения в увязке с конкретной проблемой и потребностями целевых пользователей, выявить возможные противоречия между «проблемой и решением» и найти способы их разрешения. Тем самым методика ТРИЗ встраивается в процесс формулирования и проверки ценностного предложения на стадии Problem-Solution Fit.

Это повышает вероятность того, что разрабатываемый продукт действительно решает значимую задачу клиента и обладает жизнеспособностью до выхода на рынок. В дальнейшем, по мере перехода к Product-Market Fit, расширенный системный оператор продолжает приносить пользу, позволяя выявлять ограничения для масштабирования и точки роста продукта в контексте более широкой бизнес-системы. Таким образом, разработанный инструмент укрепляет роль Бизнес ТРИЗ в управлении инновациями на всех фазах - от зарождения концепции и проверки гипотез до развития продукта и его масштабирования.

Направления для будущих исследований

В ходе работы обозначен ряд перспективных направлений, развивающих предложенную методику.

Первое направление связано с введением понятия «*анти-продукта*» по аналогии с классической концепцией «*антисистемы*» в ТРИЗ. Анти-продукт можно определить как продукт или решение, преследующее цель, противоположную исходному продукту. Развитие этой идеи расширит аналитический инструментарий: учитывая гипотетический анти-продукт, исследователь сможет выявлять скрытые противодействующие силы, альтернативные пути решения задачи и новые ракурсы для устранения противоречий. Включение дополнительной оси «продукт - анти-продукт» в структуру расширенного системного оператора формализует работу с противоположностями в продуктовом контексте и усилит диалектический подход Бизнес-ТРИЗ. Дальнейшие исследования в этом русле позволят генерировать принципиально новые идеи и прогнозировать развитие систем с учетом противоборствующих тенденций рынка.

Второе направление - переосмысление понятия *идеальности продукта* в бизнес-контексте и сопоставление его с классическим критерием идеальности технической системы в ТРИЗ. Идеальность в ТРИЗ предполагает стремление к системе, которая выполняет свою функцию при минимуме затрат и негативных эффектов (в пределе - «системы нет, а функция выполняется»). Однако для продукта на рынке идеальность включает дополнительные критерии: максимальную ценность и удобство для пользователя, экономическую эффективность для производителя, экологичность и другие факторы. Возникает задача объединить эти перспективы при выработке идеального конечного образа продукта.

Будущие исследования могут сфокусироваться на методах **синтеза технической и потребительской идеальности**: как соотносить устранение технических противоречий с максимальным удовлетворением пользовательских потребностей и бизнес-целей. Возможно, потребуется скорректировать определение идеальности или ввести новые показатели, учитывающие ценностную составляющую и полный жизненный цикл продукта. Такое расширенное понимание идеальности обогатит теорию ТРИЗ и даст методологические ориентиры для создания продуктов, максимально приближенных к идеалу не только по техническим параметрам, но и по пользовательским свойствам. Это усилит связь между ТРИЗ и современным продуктовым мышлением, что особо важно для Бизнес-ТРИЗ.

Третье направление - создание новых моделей бизнес-системы, которые будут интегрироваться с ранними стадиями ЖЦ продукта и помогать решать задачи Problem - Solution Fit.

Четвертое направление - развитие существующих инструментов и моделей ТРИЗ и Бизнес ТРИЗ за счет интеграции в них расширенного системного оператора. Например, интеграция расширенного системного оператора для

продукта на этапе MVP-анализа помогает команде проекта системно выявлять всех ключевых стейкхолдеров и их требования. Это повышает полноту и достоверность требований уже на ранних этапах - на старте проекта фиксируются критерии успеха системы и цели всех участников. Такой подход обеспечивает чёткую связь между сценариями использования и ролями акторов: например, каждый сценарий использования напрямую привязан к конкретному пользователю или системе, что значительно улучшает согласованность в работе команды. В итоге такой подход снижает риск упущенных требований и конфликтов: пропущенные сценарии часто проявляются лишь на более поздних этапах разработки.

Подводя итог, разработанный **расширенный системный оператор для продукта** заложил основу для дальнейшей эволюции Бизнес ТРИЗ на стыке с продуктовыми и инженерными дисциплинами. Реализация обозначенных направлений (концепция анти-продукта, новая трактовка идеальности продукта и др.) будет способствовать формированию целостной методологии, объединяющей изобретательское решение проблем с системным подходом к продуктам.

Литература

1. **Альтшуллер Г. С., Шапиро Р. Б.**
Психология изобретательского творчества // *Вопросы психологии*. — 1956. — № 6. — С. 37-49.
2. **Альтшуллер Г. С.**
Как научиться изобретать. — Баку: Азербайджанское государственное издательство, 1961.
3. **Альтшуллер Г. С.**
Алгоритм изобретения. — М.: Московский рабочий, 1969.
4. **Альтшуллер Г. С.**
Алгоритм решения изобретательских задач (АРИЗ-75). — Баку, 1975.
5. **Гуд Г. Х., Макол Р. Э.**
Системотехника. Введение в проектирование больших систем. — М.: Советское радио, 1962.
(пер. с англ. Good, H. H., Machol, R. E. *Systems Engineering*, 1957)
6. **ISO. ISO 9000:2015**
Quality management systems — Fundamentals and vocabulary. — Geneva: ISO, 2015;
ISO/IEC/IEEE 15288:2015
Systems and software engineering — System life cycle processes. — Geneva: ISO, 2015.
7. **INCOSE**
Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. — 4th ed. — Hoboken, NJ: Wiley, 2015.
8. **ISO/IEC/IEEE 15288:2015**
Systems and software engineering — System life cycle processes. — Geneva: ISO, 2015.
9. **Kotler P., Keller K. L.**
Marketing Management. — 15th ed. — Pearson Education, 2016.
10. **Levitt T.**
Marketing Myopia // *Harvard Business Review*. — 1960. — Vol. 38, No. 4. — P. 45-56.
11. **Stanton W. J.**
Fundamentals of Marketing. — 10th ed. — New York: McGraw-Hill, 1994.
12. **Drucker P. F.**
Management: Tasks, Responsibilities, Practices. — New York: Harper & Row, 1973.
13. **Cagan M.**
Inspired: How to Create Tech Products Customers Love. — Hoboken, NJ: Wiley, 2017.

14. **Cagan M.**
Статьи Silicon Valley Product Group (SVPG). – 2015. – URL:
<https://svpg.com> (дата обращения: ...)
15. **Christensen C. M., Cook S., Hall T.**
Marketing Malpractice: The Cause and the Cure // *Harvard Business Review*.
– 2005.
16. **Christensen C. M.**
Competing Against Luck: The Story of Innovation and Customer Choice. –
New York: Harper Business, 2016.
17. **Drucker P. F.**
The Practice of Management. – New York: Harper & Brothers, 1954.
18. **TRIZ Body of Knowledge (TRIZBoK)**
International TRIZ Association (MATRIZ). – Электронный ресурс.
19. **ТРИЗ-гlossарий**
Международная ассоциация ТРИЗ (MATRIZ). – Электронный ресурс.
20. **ISO/IEC/IEEE 15288**
Systems and software engineering – System life cycle processes. – Geneva:
ISO.
21. **Levitt T.**
Exploit the Product Life Cycle // *Harvard Business Review*. – 1965.
22. **Rogers E. M.**
Diffusion of Innovations. – 5th ed. – New York: Free Press, 2003.
23. **ISO/IEC/IEEE 12207:2017**
Software life cycle processes. – Geneva: ISO, 2017.
24. **Blanchard B. S., Fabrycky W. J.**
Systems Engineering and Analysis. – 5th ed. – Pearson, 2008.
25. **Cooper R. G.**
Winning at New Products: Creating Value Through Innovation. – 5th ed. –
Basic Books, 2019.
26. **ISO/IEC/IEEE 12207:2017**
Software life cycle processes. – Geneva: ISO, 2017.
(дублируется намеренно – используется в разных контекстах)
27. **Vargo S. L., Lusch R. F.**
Evolving to a New Dominant Logic for Marketing // *Journal of Marketing*. –
2004. – Vol. 68. – P. 1-17.
28. **Альтшуллер Г. С.**
Развитие системного мышления – конечная цель обучения АРИЗ //
Материалы ТРИЗ-семинаров. – 1975.
29. **Хоменко Н. Н.**
Теория сильного мышления (ОТСМ-ТРИЗ). – 2013. – Учебные и
методические материалы.
30. **Альтшуллер Г. С.**
Алгоритм решения изобретательских задач (АРИЗ-75). – Баку, 1975.

31. **Normann R., Ramirez R.**
From Value Chain to Value Constellation // *Harvard Business Review*. – 1993.
32. **Norton M. I., Mochon D., Ariely D.**
The IKEA Effect // *Journal of Consumer Psychology*. – 2012. – Vol. 22. – P. 453-460.
33. **Cusumano M. A.**
The Business of Software. – New York: Free Press, 2004.
34. **O'Reilly T.**
What Is Web 2.0 // *O'Reilly Media*, 2007.
35. **U.S. Bureau of Labor Statistics**
Occupational Outlook Handbook. – Washington, DC, 2022.
36. **Simon H. A.**
The Sciences of the Artificial. – 3rd ed. – MIT Press, 1969.
37. **Maier M. W., Rechtin E.**
The Art of Systems Architecting. – 3rd ed. – CRC Press, 2009.
38. **Альтшуллер Г. С.**
Творчество как точная наука. – М.: Советское радио, 1984.
39. **Mann D.**
System Operator Tutorial – 9 Windows. – 2001.
40. **Mann D.**
Hands-On Systematic Innovation. – Clevedon: IFR Press, 2002.
41. **Альтшуллер Г. С.**
Найти идею. – Новосибирск: Наука, 1986.
42. **Fiorineschi L. et al.**
Adapted Use of the TRIZ System Operator // *Procedia CIRP*. – 2021.
43. **Альтшуллер Г. С.**
Законы развития технических систем. – 1979.
44. **Porter M. E., Heppelmann J. E.**
How Smart, Connected Products Are Transforming Competition // *Harvard Business Review*. – 2014.
45. **Рубин М. С.**
Основы ТРИЗ. Применение ТРИЗ в программных и информационных системах: учебное пособие. – СПб: СПбГУ, 2011.
46. **Kuryan A., Rubin M., Shchedrin N., Eckardt O., Rubina N.**
TRIZ Ontology: Current State and Perspectives // *Proceedings of TDS-2020*. – Minsk, 2020.
47. **Khomenko N.**
Advanced Multi-Screen Scheme. – 2015.
48. **Souchkov V.**
Typical Patterns of Business Model Innovation // *Proceedings of TRIZ Future 2015*. – 2015.

49. Сушков В.
Модель бизнес-системы. Презентация с кейсами. – 2022.
50. Бланк С.
Стартап: Настольная книга основателя, 2013
51. Рис Э.
Бизнес с нуля: Метод Lean Startup для быстрого тестирования идей и выбора бизнес-модели, 2016
52. Остервальдер А.
Разработка ценностных предложений, 2017
53. Остервальдер А.
Построение бизнес-моделей, 2012
54. Гассман О. И др.
Бизнес-модели: 55 лучших шаблонов, 2017
55. Andreessen M.
The only thing that matters, 2007
56. Litvin S.,
Main Parameters of Value, 2011
57. Kozhemyako A.,
TRIZ: A Practical Guide for Business, 2021
58. Sims & Kogan,
Bringing Innovation to the Innovation Process, 2005
59. GEN3 Partners,
What Are Main Parameters of Value?, 2012
60. Ульрих, Эппингер,
Product Design and Development, 2016
61. Саунин Н.Е.,
Функциональный анализ в организационно-управленческих задачах.
Обратный подход, 2019
62. Саунин Н.Е., Фоменко А.Ю., Савенков А.Ю.
Анализ ключевых параметров успешности (АКПУ): Методическое пособие, 2023

Приложение 1. Сравнительный анализ понятия “система” в СИ и ТРИЗ

| Критерий сравнения | Системная инженерия (СИ) | ТРИЗ | Отличия |
|----------------------------|--|--|--|
| <p>Определение системы</p> | <p>Рассматривает систему как совокупность взаимодействующих элементов, организованных для достижения определённой цели. В определение системы включаются как технические компоненты, так и люди, процессы и другие элементы (социотехнический подход).</p> | <p>Опирается на понятие технической системы - искусственно созданной совокупности частей, предназначенной для выполнения определенной полезной функции, и обладающей свойством целого, превышающим сумму свойств ее частей. Фокус на материальных компонентах и их взаимосвязях, дающих новый эффект (эмерджентность).</p> | <p>ТРИЗ трактует систему преимущественно как техническую функциональную совокупность элементов с эмерджентными свойствами, тогда как СИ задает более широкое определение системы, включающее различные типы элементов (технические, человеческие и пр.) и формальное целевое назначение.</p> |

| | | | |
|------------------------------|--|---|---|
| <p>Цель и задачи подхода</p> | <p>Главная задача СИ - создание успешной системы, удовлетворяющей требования и потребности всех заинтересованных сторон. СИ - междисциплинарный подход, увязывающий потребности заказчика с архитектурой системы и ее реализацией, чтобы обеспечить работоспособность и эффективность системы в жизненном цикле.</p> | <p>Цель ТРИЗ - систематическое решение изобретательских задач: выявление и устранение ключевых противоречий с целью нахождения идеального решения при минимуме проб и ошибок.</p> | <p>ТРИЗ носит инновационно-ориентированный характер - стремится к поиску идеальных решений через разрешение противоречий, тогда как СИ носит целеориентированный характер - фокусируется на удовлетворении заданных требований и балансировке множества критериев для обеспечения успешности системы.</p> |
|------------------------------|--|---|---|

| | | | |
|--------------------------------|--|--|---|
| <p>Работа с противоречиями</p> | <p>При конфликте требований или критериев СИ проводит анализ компромиссов: применяется оптимизация и trade-off исследования, чтобы сбалансировать альтернативы и выбрать решение, наилучшим образом удовлетворяющее всем ограничениям (стоимость, надежность, эффективность и т.д.). Традиционно в инженерном проектировании противоречия приводят к поиску компромиссного решения, удовлетворяющего минимум необходимого.</p> | <p>Противоречия рассматриваются как источник изобретательских решений. ТРИЗ предлагает выявлять технические и физические противоречия и применять специальные инструменты для их разрешения без компромисса. Решение по ТРИЗ - это решение, устраняющее противоречие, а не оптимизирующее его.</p> | <p>ТРИЗ ориентирована на устранение противоречий (нахождение решения без взаимных уступок), СИ – на их балансировку, т.е. достижение приемлемого компромисса между конфликтующими требованиями посредством анализа и оптимизации.</p> |
|--------------------------------|--|--|---|

| | | | |
|-----------------------------|--|--|---|
| <p>Методы и инструменты</p> | <p>Опирается на формальные процессы и модели. В СИ применяются структурированные методы: инженерия требований, функционально-структурное моделирование системы, архитектурный дизайн, анализ рисков, верификация и валидация и др. - всё это документируется и отслеживается, увязывая потребности с техническими решениями.</p> | <p>В ТРИЗ разработаны алгоритмы и приемы для решения изобретательских задач: АРИЗ, 40 приемов, 76 стандартов, законы развития технических систем и др.</p> | <p>Методология СИ базируется на строгих аналитических процедурах и формальном моделировании (рациональный подход), тогда как ТРИЗ использует набор эвристик и алгоритмов изобретательства для систематизации творчества (интуитивно-логический подход).</p> |
|-----------------------------|--|--|---|

| | | | |
|-------------------------------|---|---|--|
| <p>Эволюция и идеальность</p> | <p>Системная инженерия управляет изменениями системы через жизненный цикл системы: проектирование, разработку, тестирование, внедрение и эксплуатацию. Улучшения вносятся итеративно по мере появления новых требований и технологий, но универсальных законов эволюции система не имеет - развитие определяется конкретным контекстом и потребностями проекта. Главный критерий - соответствие системы заданным целям и ограничениям на каждой стадии.</p> | <p>ТРИЗ включает представление о закономерностях развития технических систем. В теоретическом ядре ТРИЗ сформулированы законы и линии эволюции систем, например рост идеальности, переход на надсистемный уровень, разрешение противоречий и др.. Вводится понятие Идеальной системы - состояние, когда система выполняет свою функцию без затрат или вредных факторов; практический поиск приближается к идеальному решению через устранение противоречий.</p> | <p>ТРИЗ основывается на законах развития технических систем и идеале системы (стремлении к максимально эффективной, «идеальной» системе), тогда как СИ оперирует универсальными законами эволюции - она фокусируется на поэтапном улучшении системы в рамках конкретного проекта и управления ее жизненным циклом в заданных условиях. ТРИЗ направляет развитие системы по общим тенденциям, СИ - по специфическим требованиям и этапам.</p> |
|-------------------------------|---|---|--|

Приложение 2. Сравнительный анализ понятий “система” и “процесс”

| Критерий | Система | Процесс | Комментарии |
|-------------------------|--|--|--|
| Определение | «Совокупность взаимодействующих элементов, организованных для достижения одной или более заявленных целей» [ISO/IEC/IEEE, <i>System Life Cycle Processes</i> , 2015]. | «Набор взаимосвязанных или взаимодействующих действий, преобразующих входы в выходы» [ISO, <i>ISO 9000</i> , 2015]. | Определения взяты из ISO/IEC/IEEE 15288 и INCOSE Handbook. |
| Структура (организация) | Пространственная структура: элементы сосуществуют одновременно, образуя конфигурацию в пространстве (статическая совокупность в данный момент) [INCOSE, <i>Systems Engineering Handbook</i> , 2015]. | Временная структура: операции следуют последовательно во времени, образуя динамический поток или этапы процесса [ISO/IEC/IEEE, <i>System Life Cycle Processes</i> , 2015]. | Ключевое различие – пространственная vs временная организация. |
| Составные элементы | Состоит из компонентов разной природы (физические устройства, люди, программы, данные и т.д.), которые взаимодействуют между собой [ISO/IEC/IEEE, <i>System Life Cycle Processes</i> , 2015]. | Состоит из операций/действий, выполняемых в определённой последовательности; каждая операция преобразует входы в выходы [ISO, <i>ISO 9000</i> , 2015]. | Процессы могут быть элементами системы. |

| | | | |
|---------------------------------------|--|---|---|
| <p>Цель и результат</p> | <p>Достигает цели через совместное функционирование элементов и эмерджентные свойства системы [INCOSE, <i>Systems Engineering Handbook</i>, 2015].</p> | <p>Направлен на достижение заданного результата по завершении всех шагов процесса [ISO/IEC/IEEE, <i>System Life Cycle Processes</i>, 2015].</p> | <p>Система реализует функции, процесс – производит результат.</p> |
| <p>Взаимоотношение понятий</p> | <p>Может включать процессы как внутренние элементы или аспекты функционирования [ISO/IEC/IEEE, <i>System Life Cycle Processes</i>, 2015].</p> | <p>Может рассматриваться как система операций с временной структурой [INCOSE, <i>Systems Engineering Handbook</i>, 2015].</p> | <p>Процесс – частный случай системы с временной организацией.</p> |

Приложение 3. Сравнительный анализ понятий “система” и “продукт”

| Критерий | Система | Продукт | Общее/Различие |
|---------------------------|--|--|--|
| <p>Определение</p> | <p>В системной инженерии система определяется как совокупность взаимодействующих элементов, организованных для достижения определённой цели; в понятие системы включаются не только технические компоненты, но и люди, процессы и другие элементы (социотехнический подход). В ТРИЗ под системой прежде всего понимается техническая система - искусственно созданная совокупность частей, предназначенная для выполнения определённой функции или удовлетворения потребности, обладающая свойством целого, превышающим сумму свойств частей (эмерджентность).</p> | <p>В классическом маркетинге под продуктом понимается чрезвычайно широкое понятие: «всё, что может быть предложено рынку для удовлетворения какой-либо потребности или желания», включающее не только физические товары, но и услуги, идеи, места и даже личности. Иными словами, продукт - это не просто материальный объект, а совокупность свойств и выгод, способных удовлетворить запросы потребителей; покупатель приобретает не вещь как таковую, а пользу или решение своей проблемы, которое эта вещь ему приносит.</p> | <p>Общее: Оба понятия обозначают целостный комплекс элементов, создаваемый с целью удовлетворения определенных потребностей или выполнения функций. Различие: «Система» трактуется преимущественно как технический объект с внутренней структурой и эмерджентными свойствами, тогда как «продукт» - как рыночное предложение ценности, ориентированное на решение проблемы потребителя.</p> |

| | | | |
|-----------------------------|--|---|--|
| <p>Цель и задачи</p> | <p>Главная задача при работе с системой (инженерный подход) - создание работоспособного технического решения, удовлетворяющего заданным требованиям и потребностям всех заинтересованных сторон на протяжении жизненного цикла системы. В ТРИЗ цель формулируется как систематическое решение изобретательских задач через выявление и устранение ключевых противоречий, что позволяет найти идеальное решение с минимумом проб и ошибок. Таким образом, системный подход фокусируется на достижении заданных технических целей и балансе критериев эффективности системы.</p> | <p>С точки зрения управления продуктом, продукт рассматривается как единица бизнеса, которую необходимо планировать, развивать и поддерживать. Цель продакт-менеджмента - обеспечить конкурентоспособность и прибыльность продукта, управляя им на всех стадиях жизненного цикла (от идеи и разработки до спада). Питер Друкер отмечал, что цель бизнеса - «создать потребителя», поэтому управление продуктом должно основываться на понимании потребностей клиента и их постоянном удовлетворении лучше конкурентов. Со стороны потребителя задача продукта - наиболее полно решить проблему пользователя или удовлетворить его потребность, предоставив ценность, за которую он готов заплатить.</p> | <p>Общее: И система, и продукт создаются целенаправленно - они разрабатываются для выполнения определенных задач и удовлетворения чьих-либо потребностей. Различие: Системный подход ориентируется на технические критерии успеха (функциональность, надежность и соответствие спецификациям) и решение инженерных задач, тогда как продуктовый подход ставит во главу угла ценность для пользователя и успех на рынке (удовлетворенность клиента, конкурентоспособность).</p> |
|-----------------------------|--|---|--|

| | | | |
|-----------------------------------|---|---|--|
| <p>Эволюция и развитие</p> | <p>В ТРИЗ развитие технических систем рассматривается как закономерный процесс: выделены законы и линии эволюции (повышение идеальности, переход на надсистемный уровень, разрешение противоречий и др.), которые направляют систему к идеальному состоянию, при котором функция выполняется без затрат или вредных побочных эффектов. Системная инженерия, напротив, не оперирует универсальными законами развития: эволюция системы управляется поэтапно в контексте конкретного проекта и требований. Изменения в системе вносятся итеративно по мере появления новых потребностей и технологий, а основной критерий - соответствие системы целям и ограничениям на каждой стадии ее жизненного цикла.</p> | <p>Продукт также не статичен: его разрабатывают, выводят на рынок, он проходит стадии роста, зрелости и спада. Классический маркетинг вводит понятие жизненного цикла продукта - от идеи через запуск и зрелость до упадка - и подчеркивает необходимость управлять продуктом на каждом из этих этапов. В современном продуктовом менеджменте развитие продукта носит непрерывный характер: продукт постоянно улучшается и адаптируется под меняющиеся потребности пользователей и условия рынка, чтобы оставаться востребованным. Таким образом, продукт эволюционирует под влиянием внешних факторов (обратной связи потребителей, конкуренции, технологий) с целью сохранять ценность для клиента на протяжении времени.</p> | <p>Общее: И система, и продукт обладают жизненным циклом и со временем претерпевают изменения и улучшения, то есть развиваются, а не существуют в неизменном виде.</p> <p>Различие: Развитие системы в инженерном понимании часто руководствуется внутренними закономерностями технического прогресса и достижением идеального состояния, тогда как развитие продукта определяется внешними рыночными факторами - этапами рыночного жизненного цикла и изменяющимися потребностями потребителей.</p> |
|-----------------------------------|---|---|--|

| | | | |
|---|---|---|---|
| <p>Подход к решению проблем/противоречий</p> | <p>В инженерной практике при возникновении противоречивых требований система дорабатывается через поиск компромиссов: методами оптимизации и trade-off анализа достигается баланс, позволяющий удовлетворить минимально необходимые условия по всем критериям (стоимость, надежность, эффективность и т.д.). Напротив, в ТРИЗ противоречия рассматриваются как источник инноваций: изобретательский подход предполагает явное выявление технических или физических противоречий и применение специальных инструментов для их устранения без взаимных уступок.</p> | <p>В продуктовом подходе под «проблемой» подразумевается проблема пользователя, которую продукт должен решить. Современные концепции (например, теория Jobs to Be Done) предлагают сфокусироваться на задаче, ради которой потребитель «нанимает» продукт: успешным считается продукт, который наиболее точно и удобно выполняет нужную работу для клиента. Если продукт перестает справляться с задачей (не решает проблему должным образом), потребитель «увольняет» его - отказывается от него в пользу другого решения. При этом продакт-менеджмент должен одновременно учитывать и устранять три вида ограничений: обеспечить полезность продукта для пользователя, его техническую реализуемость и коммерческую целесообразность для бизнеса. Таким образом, решение в сфере продукта - это создание такого предложения, которое полностью закрывает проблему пользователя в рамках технических и бизнес-ограничений.</p> | <p>Общее: Оба подхода нацелены на эффективное преодоление ключевых проблем, препятствующих успешному функционированию системы или продукта.</p> <p>Различие: В работе с системами основной упор делается на устранение внутренних технических противоречий или поиск инженерных компромиссов для удовлетворения требований стейкхолдеров, тогда как в работе с продуктом акцент смещен на внешнюю задачу пользователя - сначала нужно понять проблему пользователя, затем сформировать решение, которое должно решить проблему пользователя максимально полно, преобразовать решение в продукт, адаптированный к рынку, и затем масштабировать продукт для максимально широкого круга потребителей.</p> |
|---|---|---|---|

| | | | |
|------------------------------------|---|--|--|
| <p>Методы и инструменты</p> | <p>Разработка и анализ систем опираются на формализованные методы инженерии. В системной инженерии применяются строгие методологические подходы: инженерия требований, функционально-структурное моделирование, архитектурное проектирование, анализ рисков, верификация и валидация и т.д., причем все шаги документируются и прослеживаются, чтобы увязать потребности с техническими решениями. В ТРИЗ, со своей стороны, сформирован набор собственных инструментов для решения изобретательских задач.</p> | <p>Инструментарий управления продуктом и маркетинга отличается ориентацией на пользователя и рынок. Применяются методы изучения потребностей и поведения клиентов (маркетинговые исследования, интервью, анализ Jobs to Be Done), итеративная разработка и тестирование продукта (гибкие методологии разработки, прототипирование, A/B тесты), а также стратегические инструменты планирования: определение ценностного предложения, управление продуктовым портфелем, дорожная карта развития продукта. В маркетинге выделяют концепцию уровней продукта - ядро (базовая выгода), реальный продукт (конкретные характеристики) и расширенный продукт (сервисное окружение, бренд и пр.) - для понимания ценности предложения для потребителя. Продакт-менеджмент интегрирует усилия разных функций (разработка, дизайн, маркетинг, продажи) вокруг продукта, обеспечивая комплексный процесс от идеи до вывода на рынок и дальнейшего улучшения продукта.</p> | <p>Общее: И в системной инженерии, и в продуктовом менеджменте используются системные подходы и инструменты - отказ от хаотичного поиска решений в пользу методических процедур. Различие: Методы работы с системой носят преимущественно технико-аналитический характер (формальные модели, расчёты, алгоритмы изобретательства), тогда как методы работы с продуктом ориентированы на рынок и пользователя (исследование потребностей, дизайн пользовательского опыта, итеративные улучшения, межфункциональное управление).</p> |
|------------------------------------|---|--|--|

| | | | |
|--|---|---|--|
| <p>Ориентация на пользователя</p> | <p>В контексте системного подхода интересы конечного пользователя учитываются как часть требований: например, системная инженерия включает человека в число элементов системы (социотехнический подход) и стремится удовлетворить потребности заказчика/пользователя наряду с другими критериями. Однако прямой пользовательский опыт не всегда находится в фокусе - при проектировании системы упор делается на функциональность, надежность и соответствие спецификации, а удовлетворенность пользователя выступает одним из аспектов эффективности системы. В истории ТРИЗ ориентация была прежде всего на техническое совершенствование системы и разрешение её внутренних противоречий; потребительские аспекты результата предполагаются (идеальная система приносит максимальную</p> | <p>Продукт по своей сути является носителем ценности для пользователя, поэтому изначально определяется через призму потребительских выгод и опыта. Маркетинг подчеркивает, что покупатель ценит продукт за решение своей проблемы и удовлетворение потребности. Если ожидания пользователя не оправдываются, продукт не будет иметь спроса, каким бы совершенным ни были его технические характеристики. В управлении продуктом ориентация на клиента - ключевой принцип: понимание потребностей потребителей и повышение их удовлетворенности лежит в основе разработки и эволюции продукта. Успех продукта на рынке напрямую измеряется степенью удовлетворения и лояльности пользователей.</p> | <p>Общее: Оба понятия нацелены на создание ценности для конечного пользователя/заказчика - система выполняет функцию, имеющую полезный эффект, а продукт решает проблему клиента, принося ему пользу. Различие: Продуктовый подход изначально и явно ставит пользователя в центр внимания - ценность и успешность продукта определяются удовлетворением потребительских ожиданий. В системном же подходе потребительские требования являются лишь одной из множества переменных; ориентированность на пользователя присутствует, но без приоритизации его требований по сравнению с требованиями других стейкхолдеров.</p> |
|--|---|---|--|

| | | | |
|--|------------------------------------|--|--|
| | пользу), но явно не анализируются. | | |
|--|------------------------------------|--|--|

Приложение 4. Методика РСО для продукта

| Шаг | Что делаем | Результат |
|--|--|---|
| 1. Определение продукта и границ системы | <p>Чётко формулируем цель анализа (поиск проблем, поиск ресурсов, анализ стейкхолдеров, прогноз развития, разработка нового продукта и т.д.). Фиксируем, <i>какой продукт</i> анализируется.</p> <p>Определяем границы системы: что относится к продукту, а что – к надсистеме.</p> <p>Определяем уровень детализации, необходимый именно для этой цели.</p> | Согласованный объект анализа |
| 2. Построение оси жизненного цикла продукта в настоящем | <p>Определяем стадии жизненного цикла продукта, релевантные цели анализа (не обязательно полный ЖЦ – только значимые стадии).</p> <p>Для выбранных стадий выделяем микро-стадии - эпизоды существования продукта во времени.</p> <p>Выделяем активные и пассивные микро-стадии.</p> | Ось жизненного цикла продукта в настоящем, детализированная до уровня микро-стадий, релевантных цели анализа. |
| 3. Описание сценариев взаимодействия со стейкхолдерами | <p>Для каждой стадии и/или микро-стадии описываем сценарии взаимодействия.</p> <p>Для каждого сценария фиксируем:</p> <ul style="list-style-type: none"> а) стейкхолдера, б) его роль (один и тот же стейкхолдер может иметь разные роли), в) цель взаимодействия, г) характер воздействия на продукт. | Связка: <i>стадия / микро-стадия → сценарий взаимодействия → стейкхолдер → роль</i> |
| 4. Декомпозиция продукта на подсистемы (вариативная) | <p>Для каждой стадии ЖЦ (или группы микро-стадий) выбираем адекватный способ декомпозиции:</p> <ul style="list-style-type: none"> а) функциональный, б) конструктивный, в) процессный (для сервисов); г) организационный и т.д. <p>Допускаем разные декомпозиции для разных стадий.</p> | Набор представлений продукта в настоящем как системы, адаптированных под конкретные стадии и сценарии. |
| 5. Построение базовой многомерной модели системного | <p>Для выбранных стадий ЖЦ строим системный оператор по осям:</p> <ul style="list-style-type: none"> а) иерархия, б) ось ЖЦ | Статическая многомерная модель продукта в настоящем |

| | | |
|--|---|---|
| оператора | Надсистемы задаются через сценарии взаимодействия , а не абстрактно. | («здесь и сейчас») (разрез по иерархии × стадиям ЖЦ) |
| 6. Анализ эволюции (ось эволюции) | <p>Для ключевых элементов модели (микро-стадий, подсистем):</p> <p>а) анализируем прошлые состояния, б) фиксируем текущее состояние, в) формируем образы будущего.</p> <p>Эволюция рассматривается:</p> <p>а) как развитие продукта, б) как изменение структуры ЖЦ, в) как изменение ролей и сценариев.</p> | <p>Динамическая модель продукта, объединяющая:</p> <p>а) эволюцию системы, б) эволюцию ЖЦ продукта, в) эволюцию сценариев взаимодействия и ролей.</p> |
| 7. Аналитическое использование расширенного оператора | <p>Выявляем:</p> <ul style="list-style-type: none"> - противоречия между стадиями ЖЦ, - противоречия между сценариями, - противоречия между ролями стейкхолдеров, - неиспользуемые ресурсы (особенно на пассивных микро-стадиях). <p>Формируем направления развития продукта и постановки изобретательских задач.</p> | <p>Изобретательские задачи.</p> <p>Основание для решений в логике ТРИЗ и продуктового мышления.</p> |

Приложение 5. Пример

В данном разделе представлен практический пример применения PCO: подключение нового клиента к ИИ-агенту квалификации лидов (HubSpot + маркетплейс)

1. Постановка задачи и границы целевой системы («Система / Настоящее»)

Целевая система: услуга квалификации лидов, реализуемая ИИ-агентом, встроенным в процесс обработки лидов клиента и интегрированным с CRM HubSpot.

Цель: обеспечить работоспособное подключение и устойчивую эксплуатацию агента у нового клиента с гарантированным качеством квалификации и корректной тарификацией.

Граница системы (операционально):

- Внутри границы: агент (модель/оркестрация/промтты/правила), коннектор к HubSpot, конфигурация отображения свойств, тестовый контур, контур оплаты/лицензирования, журналирование и мониторинг.
- Вне границы: внутренняя политика безопасности клиента, качество данных в CRM, организационная зрелость процесса продаж, обученность SDR/менеджеров, ограничения HubSpot API и план подписки.

PCO фиксирует, что «подключение клиента» — это не только техническая интеграция, но и **информационная** система: технологическая подсистема (интеграция/настройка/оплата) неразрывна с процессом продаж и ответственностями ролей.

2. Надсистема и контекст («Надсистема / Настоящее»)

Надсистема: коммерческая и операционная среда клиента, в которой происходит лид-менеджмент:

- CRM HubSpot (объекты Lead/Contact/Company/Deal, пайплайны, свойства, workflow automation);
- процесс SDR/AE (кто принимает решения, кто отвечает за квалификацию, SLA по обработке лидов);

- каналы привлечения лидов и источники данных (формы сайта, интеграции, партнёры);
- требования комплаенса и информационной безопасности (доступы, хранение данных, аудит);
- биллинг/закупки и финансовые процедуры (счета, лимиты, бюджет).

Практический вывод PCO: качество работы агента зависит от надсистемы (процесса и данных), поэтому успешное подключение требует не только API-ключа, но и согласования **организационных условий**: кто владелец процесса, какие критерии качества, какие права доступа допустимы, как фиксируется результат квалификации в CRM.

3. Подсистемы («Подсистема / Настоящее»)

PCO выделяет ключевые подсистемы, критичные для выполнения 4 этапов подключения:

1. **Интеграционная подсистема:** коннектор (OAuth/Private App), обработка вебхуков/поллинга, лимиты API, ретраи, дедупликация, идемпотентность.
2. **Данные и отображение свойств:** мэппинг свойств HubSpot ↔ входы/выходы агента; нормализация значений; справочники (industries, regions); правила заполнения обязательных полей.
3. **Логика квалификации:** промпт/политики, критерии квалификации (ICP, intent, budget, timeline), обработка неопределённости, fallback-сценарии.
4. **Тестирование и контроль качества:** тестовый набор лидов, метрики точности/полноты, ручная валидация, логирование решений.
5. **Подсистема оплаты:** подписка/usage-based, учёт обработанных лидов, ограничения тарифа, отключение при неуплате, инвойсинг.
6. **Безопасность и доступ:** разграничение прав, токены, хранение секретов, аудит, соответствие политике клиента.

Практический вывод: этап (3) «настройка связей» на деле является интеграцией нескольких подсистем и является главным источником рисков (данные, мэппинг, критерии качества).

4. Временная ось + жизненный цикл как дополнительная ось РСО

В РСО рассматриваем состояние системы в трёх временных срезах («Прошлое-Настоящее-Будущее») и накладываем жизненный цикл подключения клиента как последовательность стадий.

4.1. Прошлое (до подключения)

- У клиента уже есть CRM-процессы, исторические лиды, существующие свойства и пайплайны.
- Качество данных неоднородно: пропуски, разный формат, кастомные поля, дубли.
- Правила квалификации могут быть неформализованы или отличаться по сотрудникам.

Риск, выявляемый РСО: агент будет «казаться плохим», если тестируется на исторических лидах без учёта того, что они создавались в иных процессах и по другим критериям.

4.2. Настоящее (подключение и ввод в эксплуатацию)

Стадии (соответствуют этапам пользователя):

1. подключение к HubSpot;
2. тестирование на существующих лидах;
3. настройка (мэппинг, правила, критерии);
4. настройка оплаты.

РСО-вывод: это не линейный процесс, а итеративный цикл, где этапы 2-3 повторяются до достижения порога качества, а этап 4 требует фиксации, что именно считается «единицей потребления» (lead qualified? action created? deal updated?).

4.3. Будущее (эксплуатация и масштабирование)

- Изменение CRM-схемы у клиента (новые поля, смена пайплайна).

- Рост объёма лидов и необходимость масштабирования.
- Эволюция критериев квалификации (новые сегменты, продукты).
- Появление дополнительных источников данных (внешние enrichment-сервисы).

Требование, выводимое через PCO: нужна поддержка управляемости изменений: версионирование конфигурации, мониторинг деградации качества, контроль изменений схемы HubSpot, алерты и регламент обновления.

5. Разбор этапов подключения через PCO (уровни + жизненный цикл)

Этап 1. Подключение агента (маркетплейс → HubSpot)

- **Система:** агент + коннектор + учетная запись клиента.
- **Подсистемы:** OAuth/ключи, права доступа, вебхуки, лимиты API, журналы интеграции.
- **Надсистема:** политика безопасности, юридическое согласование, SLA по доступам.

Типовые риски (PCO):

- недостаточные права (агент не может читать нужные свойства/объекты);
- ограничение по API (rate limit) → пропуски событий;
- отсутствие наблюдаемости (нельзя диагностировать ошибки интеграции).

Ресурсы для улучшения:

- журнал интеграции HubSpot + события вебхуков;
- тестовая песочница HubSpot;
- стандартные свойства HubSpot как «общий знаменатель» для первичного мэппинга.

Этап 2. Тестирование на существующих лидах

- **Система:** агент выполняет квалификацию, записывает результат обратно в HubSpot (например, score, статус, комментарий).
- **Подсистема контроля качества:** выборка лидов, «golden set», ручная проверка, метрики.

Критерии качества (пример):

- доля лидов, которым присвоен корректный статус квалификации;
- согласованность решения агента с историей (если критерии стабильны);
- доля «неопределенных» решений и причины.

PCO-риски:

- исторические данные не репрезентативны;
- отсутствует эталон («как должно быть»);
- часть полей не заполнена → агент работает на неполном контексте.

Ресурсы:

- существующие сделки и исходы (won/lost) как прокси-разметка;
- примеры квалификационных комментариев SDR;
- правила workflow в HubSpot (как источник бизнес-логики клиента).

Этап 3. Настройка: связи свойств HubSpot ↔ входы/выходы агента

Это ключевой шаг, где PCO позволяет связать уровни системы и жизненный цикл.

Входы агента (примерные категории):

- данные о лиде: источник, UTM, форма, страницы, сообщение;
- данные о контакте/компании: индустрия, размер, география;

- контекст процесса: этап пайплайна, ответственный, SLA.

Выходы агента (примерные категории):

- статус квалификации (Qualified/Disqualified/Need info);
- причина/обоснование (текст);
- следующий шаг (назначить звонок, запросить данные, передать в АЕ);
- скоринг/уверенность;
- предлагаемый сегмент/ICP-класс.

PCO-правила настройки (как методика):

1. на уровне **Системы** фиксируется целевой формат результата (какие свойства в HubSpot должны обновляться и кто их использует);
2. на уровне **Подсистем** задаются мэппинги «поле → входной параметр» и «выход → свойство/действие»;
3. на уровне **Надсистемы** проверяется совместимость с процессом продаж (не нарушаем регламент, не дублируем действия людей/автоматизаций).

PCO-риски:

- семантическая несовместимость полей (одно и то же поле означает разное у разных клиентов);
- конфликты с существующими автоматизациями HubSpot;
- отсутствие минимально необходимого набора данных (MVD – minimum viable data).

Ресурсы:

- словарь свойств HubSpot (property metadata);

- история изменений по объектам (audit trail);
- шаблоны пайплайнов/статусов и best practices.

Этап 4. Настройка оплаты агента

- **Система:** биллинг и ограничение доступа + учёт потребления.
- **Надсистема:** финансовые процедуры клиента (закупки, счета), требования договора.

Ключевой вопрос РСО (границы и измеримость): что является «единицей потребления»?

Варианты: обработанный лид, выполненное действие, обновлённый объект, сгенерированный отчёт, пакет запросов.

Риски:

- несоответствие ожиданий: клиент считает по лидам, а система считает по вызовам;
- отсутствие прозрачности потребления;
- нет механизма «grace period» и контроля лимитов.

Ресурсы:

- лог обработок и idempotency key (для устранения двойного списания);
- отчёт в HubSpot (dashboard) как прозрачный индикатор потребления;
- тарифные планы маркетплейса как стандарт для согласования.

6. Результат применения РСО: требования, риски, артефакты

По итогам моделирования через РСО формируются отчуждаемые результаты:

1. Карта системы (границы, подсистемы, надсистема, интерфейсы).

2. **Карта жизненного цикла подключения** (стадии, входы/выходы стадий, критерии завершения).
3. **Матрица мэппинга свойств HubSpot ↔ параметры агента** (семантика, типы данных, обязательность, источники).
4. **Метрики качества и план тестирования** (golden set, пороги, процедуры ручной проверки).
5. **Паспорт интеграции** (права, лимиты, SLA, мониторинг, обработка ошибок).
6. **Модель тарификации и прозрачности потребления** (единица учёта, отчёты, лимиты, условия отключения).

7. Вывод по примеру

Применение PCO позволяет рассматривать подключение нового клиента к ИИ-агенту квалификации лидов как управляемую социотехническую систему, в которой технические шаги интеграции, настройка семантики данных, обеспечение качества и тарификация рассматриваются согласованно по уровням «подсистема-система-надсистема» и по стадиям жизненного цикла подключения. Это снижает риск «локальной оптимизации» (например, успешное подключение API при провале качества квалификации или некорректной тарификации) и обеспечивает воспроизводимость результата за счёт отчуждаемых артефактов модели.