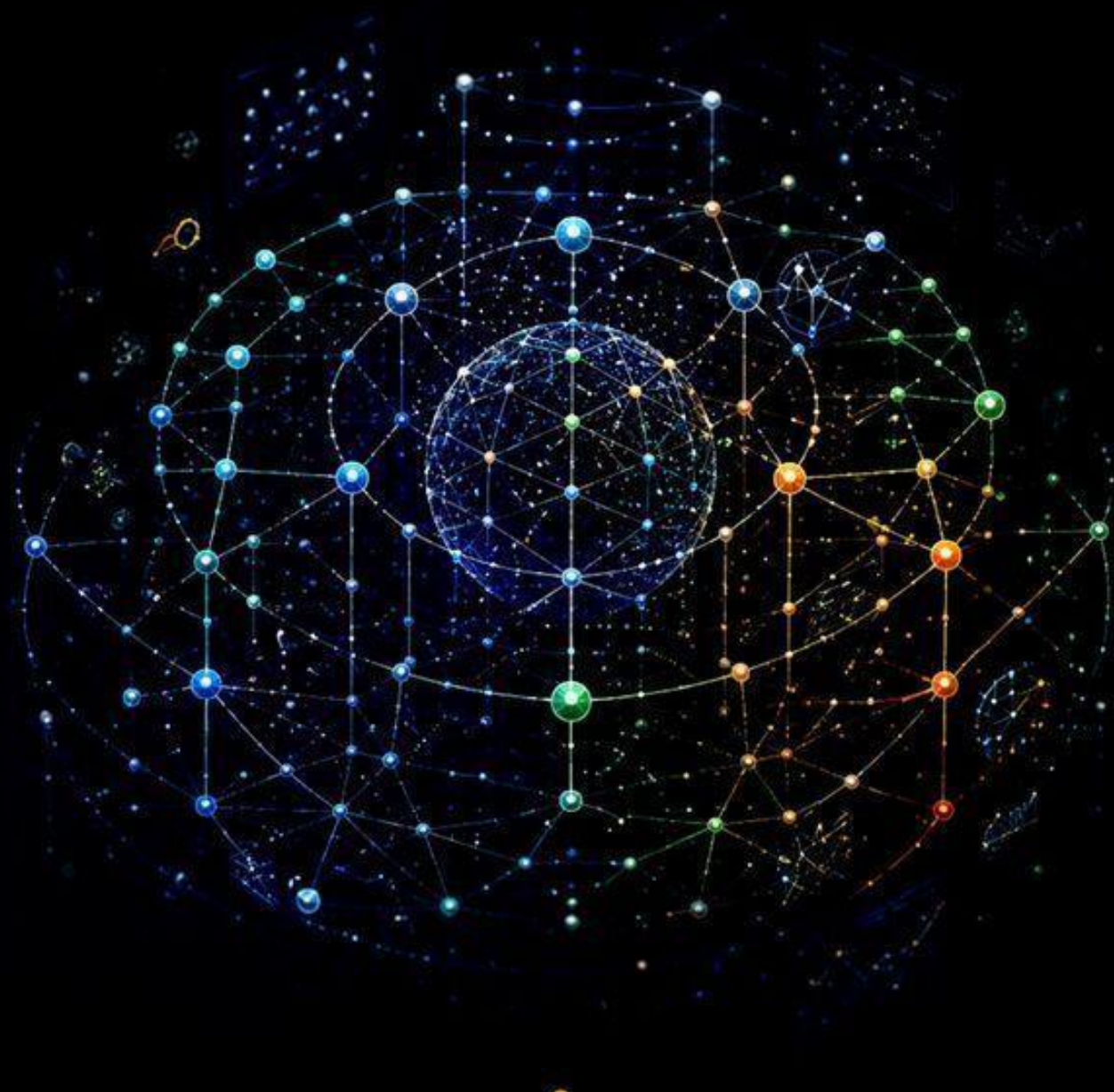


12+

НИКОЛАЙ АТАМАНЕНКО

# СЕТКА ПОНЯТИЙ

МЕТОД НАВИГАЦИИ В СЛОЖНЫХ СИСТЕМАХ



Николай Атаманенко

**Сетка понятий. Метод  
навигации в сложных системах**

«Издательские решения»

## **Атаманенко Н.**

Сетка понятий. Метод навигации в сложных системах /  
Н. Атаманенко — «Издательские решения»,

Вы понимаете ситуацию, но не можете передать её другим — слишком много деталей. Сетка понятий заменяет десять страниц текста одной картинкой. Книга учит проектировать собственные матрицы: от двумерных сеток до многомерных и динамических. Пошаговый метод и двадцать упражнений превращают анализ хаоса в рабочий навык. Для аналитиков, руководителей, стратегов, маркетологов, преподавателей, продукт-менеджеров — всех, кто работает со сложной информацией и принимает решения.

# Содержание

|                                                                  |    |
|------------------------------------------------------------------|----|
| ВВЕДЕНИЕ. Почему сетка понятий — оружие против хаоса             | 6  |
| ЧАСТЬ 1 АНАТОМИЯ СЕТКИ ПОНЯТИЙ                                   | 10 |
| Глава 1. Из чего состоит любая сетка                             | 11 |
| 1.1. Оси — измерения смысла                                      | 12 |
| 1.2. Шкалы и деления — структура оси                             | 12 |
| 1.3. Ячейки — кванты смысла                                      | 13 |
| 1.4. Зоны — области с общим решением                             | 14 |
| Итог главы                                                       | 14 |
| Глава 2. Типология сеток                                         | 15 |
| 2.1. Дихотомическая сетка (2×2, 3×3, N×M)                        | 15 |
| 2.2. Координатная (непрерывная) сетка                            | 15 |
| 2.3. Радиальная (центр-лучевая) сетка                            | 16 |
| 2.4. Иерархическая сетка (дерево)                                | 16 |
| 2.5. Многомерная сетка (более двух измерений) — анонс            | 17 |
| Итог главы                                                       | 17 |
| Глава 3. Зачем нужна сетка понятий: семь функций                 | 18 |
| 3.1. Упорядочивание: из хаоса в структуру                        | 20 |
| 3.2. Классификация: распределение объектов по типам              | 20 |
| 3.3. Диагностика: определение позиции и состояния                | 20 |
| 3.4. Навигация: понимание, где что находится                     | 21 |
| 3.5. Генерация: обнаружение пустых зон как идей                  | 21 |
| 3.6. Коммуникация: передача сложного через простое               | 22 |
| 3.7. Принятие решений: сравнение альтернатив в одних координатах | 23 |
| Итог главы                                                       | 23 |
| Итог Части 1                                                     | 24 |
| ЧАСТЬ 2. ПОСТРОЕНИЕ ДВУМЕРНЫХ СЕТОК                              | 25 |
| Глава 4. Выбор осей: самый важный шаг                            | 26 |
| 4.1. Почему выбор осей решает всё                                | 26 |
| 4.2. Четыре принципа хороших осей                                | 26 |
| 4.3. Техника «Поиск двух измерений»                              | 27 |
| 4.4. Типичные ошибки при выборе осей                             | 29 |
| Итог главы                                                       | 30 |
| Глава 5. Определение шкал и делений                              | 31 |
| 5.1. Типы шкал                                                   | 31 |
| 5.2. Правило «золотой середины»: количество делений              | 32 |
| 5.3. Определение границ делений                                  | 33 |
| Итог главы                                                       | 33 |
| Конец ознакомительного фрагмента.                                | 34 |

# **Сетка понятий**

## **Метод навигации в сложных системах**

**Николай Атаманенко**

© Николай Атаманенко, 2026

ISBN 978-5-0070-5241-2

Создано в интеллектуальной издательской системе Ridero

## ВВЕДЕНИЕ. Почему сетка понятий — оружие против хаоса

Представьте. Вы руководитель. Завтра стратегическая сессия. Перед вами на столе — распечатка на сорок страниц: данные по рынку, отчёты о конкурентах, финансовая аналитика, результаты опросов сотрудников, прогнозы развития технологий. Всё важное. Всё нужно учесть. Вы читаете это несколько часов, делаете пометки, но чем дальше, тем сильнее ощущение: вы тонете. Факты есть, а картины нет. Данные есть, а понимания нет.

Или другая сцена. Вы аналитик. Вам поручили разобраться, почему падают продажи. Вы подняли тридцать семь факторов: от ценовой политики до погоды в регионах. Все факторы — правда. Но как объяснить руководителю, что происходит? Как выделить главное, не упустив второстепенное? Как превратить список в стратегию?

У этого состояния есть точное имя — **когнитивный паралич перед сложностью**. Когда переменных больше, чем способен удержать мозг, мышление выключается и заменяется интуитивным перебором. Мы хватаемся за один-два привычных фактора и игнорируем остальные. Мы принимаем решение, которое «чувствуется правильным», но не можем его обосновать. Мы откладываем анализ, потому что он бесконечен.

Хаос — это не объективное свойство мира. Мир сложен, но не хаотичен. Хаос — это **отсутствие структуры в нашей голове**. Когда у вас нет координатной сетки, любое пространство — джунгли. Когда есть — это территория.

Решение проблемы — не в новых данных. Данных всегда будет больше, чем вы способны переработать. Решение — в способе организации данных. В методе, который превращает список в карту, а карту — в маршрут.

Этому методу посвящена книга.



Рисунок 1 – Путешественник видит деревья. Картограф видит территорию. Сетка понятий превращает джунгли данных в обозримую карту

Есть старая притча о путешественнике и картографе.

Путешественник продирается через джунгли. Он знает каждое дерево, каждый поворот тропы, каждую лозу, которая хлещет по лицу. Его опыт реален, но бесполезен для тех, кто идёт следом. Он не может объяснить, где находится, потому что видит только то, что прямо перед ним. У него нет вида сверху.

Картограф поднимается на холм. Он не чувствует лиан и не знает вкуса воды из этого ручья. Но он видит: река поворачивает здесь, обрыв — там, деревня — на северо-востоке, а джунгли кончаются через три дня пути на юг. Он накладывает на местность воображаемую сетку координат — и джунгли становятся территорией. Территорию можно описать, нарисовать и передать другому.

**Сетка понятий** — это координатная сетка для любых джунглей. Рыночных. Организационных. Технологических. Личных. Сетка превращает хаос в территорию: раскладывает объекты по осям, помещает их в ячейки и создаёт карту, на которой видно одновременно и общую структуру, и положение каждой единицы.

Хорошая новость в том, что вы уже пользуетесь сетками понятий — просто не всегда это осознаёте. Когда вы делите задачи на срочные и важные и заполняете матрицу Эйзенхауэра. Когда оцениваете риски проектов по вероятности и ущербу. Когда на совещании кто-то говорит: «Давайте разложим наших клиентов по двум параметрам: средний чек и частота покупок». Всё это — моменты, когда хаос отступает и появляется структура.

Плохая новость: большинство людей пользуются двумя-тремя готовыми матрицами, которые им когда-то показали. Они заучили шаблон, но не освоили метод. Поэтому, когда готовый шаблон перестаёт подходить, они либо подгоняют задачу под матрицу, либо возвращаются к хаотическому перебору.

Эта книга — о **методе**, а не о коллекции готовых матриц. Она учит не пользоваться матрицей SWOT, а понимать, почему её оси именно таковы — и как построить свои оси под свою задачу. Она превращает интуитивное «разложим по двум параметрам» в технологию, которую можно применять осознанно, воспроизводимо и в любом масштабе.

Определим главное понятие сразу, без излишней строгости (строгость придёт в первой части).

**Сетка понятий** — это способ наложить на хаос два или более измерений так, чтобы каждый объект получил своё место, а вся картина обрела смысл.

«Объектом» может быть что угодно: товар, сотрудник, проект, риск, идея, конкурент, клиент, компетенция. «Измерениями» — любые содержательные критерии: важность, доходность, вероятность, скорость, качество, глубина, охват. Сетка работает везде, где есть множество сущностей и потребность в различении.

Обратите внимание на слово «наложить». Оно ключевое. Сетка не извлекается из данных автоматически. Она не находится «объективным» способом. Вы её создаёте. Вы решаете, какие оси будут координатами. Вы решаете, сколько делений будет на каждой оси. Вы решаете, как назвать ячейки и какие стратегии приписать каждой. Сетка понятий — это инструмент мышления, а не свойство реальности. Это важно понять с самого начала.

Можно сказать так: данные — это хаос. Сетка — это структура. Вы — тот, кто превращает одно в другое.

Эта книга обещает вам конкретный результат.

К концу чтения вы будете способны:

— Перестать тонуть в списках и факторах — и начать видеть структуру любой сложной системы.

— Научиться выделять главные измерения любой задачи, даже если исходно их пятнадцать или двадцать.

— Проектировать собственные матрицы и сетки под конкретную задачу, не подгоняя её под готовый шаблон.

— Работать с многомерными системами — тремя, четырьмя, семью измерениями — без потери наглядности и управляемости.

— Видеть не только статичный снимок ситуации, но и траектории движения объектов, зоны решений и сценарии будущего.

— Превращать коммуникацию сложного в простую и визуальную — вместо десяти страниц текста показывать одну сетку, которая запускает осмысленное обсуждение.

Чего книга **не обещает**, и важно сказать об этом сразу:

— Она не заменит отраслевую экспертизу. Если вы не знаете свой рынок или свою организацию, сетка не сделает вас знатоком — она сделает ваши пробелы в знании более заметными, что тоже ценно, но неприятно.

— Она не является энциклопедией всех существующих матриц. Пятнадцать лучших мы разберём, но цель — не коллекционирование, а овладение методом.

— Она не учит «думать как компьютер». Сетка — инструмент, а не замена мышлению. Она структурирует суждение, но не отменяет его. Последнее слово всегда за человеком.

Книга построена как навигатор, а не как энциклопедия. Вы можете идти последовательно — от простого к сложному, от двумерных сеток к многомерным и динамическим. А можете — выбрать маршрут под свою задачу: если вам нужно прямо сейчас научиться строить сценарии, вы можете перейти к части 4 после того, как освоите базу из частей 1 и 2. Внутренние перекрёстные ссылки помогут не потеряться.

Вот краткая карта маршрута — с указанием того, какую боль снимает каждая часть:

#### **Часть 1. Анатомия сетки понятий.**

Из чего состоит любая сетка: оси, шкалы, ячейки, зоны. Типы сеток — от 2×2 до радиальных и иерархических. Семь функций, которые сетка выполняет в вашем мышлении. *Боль: непонимание, что матрица — не магия, а конструктор, который можно разобрать и собрать заново.*

#### **Часть 2. Построение двумерных сеток.**

Сердце метода. Как выбрать две независимые оси из пятнадцати факторов. Как настроить шкалы и разместить объекты. Разбор десяти классических матриц — не как образцов для слепого копирования, а как примеров чужого проектировочного мышления. Пошаговый алгоритм построения своей сетки за восемь шагов. *Боль: неспособность построить матрицу под уникальную задачу.*

#### **Часть 3. Многомерные сетки**

Выход за пределы плоскости. Как упаковать третье, четвёртое и пятое измерения в цвет, размер, слой и форму. Техники: серии проекций, радары, параллельные координаты, треугольные диаграммы. Матрица решений как инструмент многокритериального выбора. Кубическая сетка 3×3×3 для трёхмерной типологии. *Боль: ограниченность двух осей, когда реальность требует учёта пяти факторов.*

#### **Часть 4. Динамические сетки**

Время как измерение. Как строить траектории движения объектов по сетке. Как задавать пороговые зоны: красное — стоп, жёлтое — внимание, зелёное — вперёд. Как строить сценарные сетки, когда осями становятся неопределённости будущего. *Боль: статичность анализа, когда мир меняется быстрее, чем мы принимаем решения.*

#### **Часть 5. Навигация: как читать сетку и принимать решения.**

Построить сетку мало. Нужно уметь её прочитать. О чём говорят пустые ячейки? Что значит объект в центре, а что — в углу? Как от позиции на сетке перейти к типовой стратегии и конкретному действию? *Боль: есть матрица, но непонятно, что с ней делать дальше.*

#### **Часть 6. Тренажёр: двадцать задач на построение сеток.**

От базового уровня (матрица Эйзенхауэра для своих задач) до мастерского (сценарная сетка для городского планирования). Двадцать задач, каждая с описанием ситуации, заданием,

контрольными вопросами и разбором автора. *Боль: теория без практики забывается за две недели.*

**И в заключение — Сетка как образ мышления.**

Как встроить «сетчатое зрение» в повседневность — и честный разговор о пределах метода: когда сетка не работает и почему это нормально.

В конце книги — **Приложения**. Карта методов: какую сетку выбрать под какую задачу. Чек-лист «Десять вопросов к вашей сетке» для самопроверки. Библиотека из ста возможных осей для разных задач, которую вы можете использовать как справочник при проектировании.

Это не просто книга. Это рабочий набор инструментов. После первого прочтения она должна остаться на столе, а не на полке.

Если вы держите эту книгу в руках, скорее всего, вы уже чувствуете, что готовые рецепты не решают ваших задач. Вы переросли 2×2-матрицы из интернета и типовые шаблоны бизнес-школ. Вам нужно нечто более мощное, гибкое и одновременно более точное. Вам нужен не ещё один слепок чужой мысли, а способ создавать инструменты под собственную мысль.

Этот способ — не сама сетка, а умение её построить. Сетка без метода — просто картинка. Метод без сетки — просто философия. Вместе они дают вам то, что мы называли «оружием против хаоса»: способность входить в любую сложную ситуацию и выходить из неё с картой в руках.

Давайте этому научимся.

*Николай Атаманенко, г. Белгород, 2026 г.*

## ЧАСТЬ 1 АНАТОМИЯ СЕТКИ ПОНЯТИЙ

Введение описало сетку понятий как целостный инструмент — способ превращать неструктурированный массив данных в обозримую карту. Теперь нам нужно понять, как этот инструмент устроен изнутри.

Любая сетка понятий, независимо от её формы и назначения, состоит из ограниченного набора элементов: **оси, шкалы с делениями, ячейки и зоны**. Эти элементы можно осознанно выбирать и настраивать. Именно от их конфигурации зависит, будет ли сетка работать на понимание или останется формальной иллюстрацией.

Практика показывает: ошибки проектирования почти всегда связаны с игнорированием одного из этих элементов. Если выбрать зависимые оси, сетка не сможет развести объекты по разным ячейкам, и анализ подменяется ранжированием. Если шкала окажется слишком грубой, все объекты попадут в одну ячейку, и различий не возникнет. Если не определить зоны и не связать их с решением, сетка покажет позиции, но не подскажет, что с ними делать. Каждый элемент выполняет свою функцию, и потеря любого из них снижает полезность всей конструкции.

Поэтому первая часть книги — анатомическая. Её задача: разобрать сетку на составные части и рассмотреть каждую по отдельности.

**Глава 1** вводит базовые понятия — оси, шкалы, ячейки, зоны — и показывает, как они связаны между собой. **Глава 2** расширяет представление о форме: сетка — это не только прямоугольная матрица 2×2, но и радиальные, координатные, иерархические и многомерные конфигурации. **Глава 3** формулирует семь функций, которые сетка выполняет в мышлении: от упорядочивания и классификации до обнаружения пустот и поддержки решений.

К концу этой части вы сможете смотреть на любую готовую матрицу не как на монолит, а как на набор конструктивных решений — и оценивать, какие из них подходят для вашей задачи, а какие стоит пересмотреть.

Это и есть переход от пользователя готовых схем к проектировщику. Но сначала — элементная база.

## Глава 1. Из чего состоит любая сетка

Сетка понятий — не магия. Это конструкция, собранная из четырёх элементов: **осей**, **шквал (делений)**, **ячеек** и **зон**. Каждый элемент отвечает на свой вопрос.

— **Оси** — по каким критериям мы рассматриваем объекты? Что мы считаем важным?

— **Шкалы и деления** — насколько подробно мы различаем объекты по каждому критерию? Где проходят границы?

— **Ячейки** — в какие группы попадают объекты при пересечении делений?

— **Зоны** — что делать с объектами, попавшими в ту или иную часть сетки?

Разберём эти элементы по порядку.

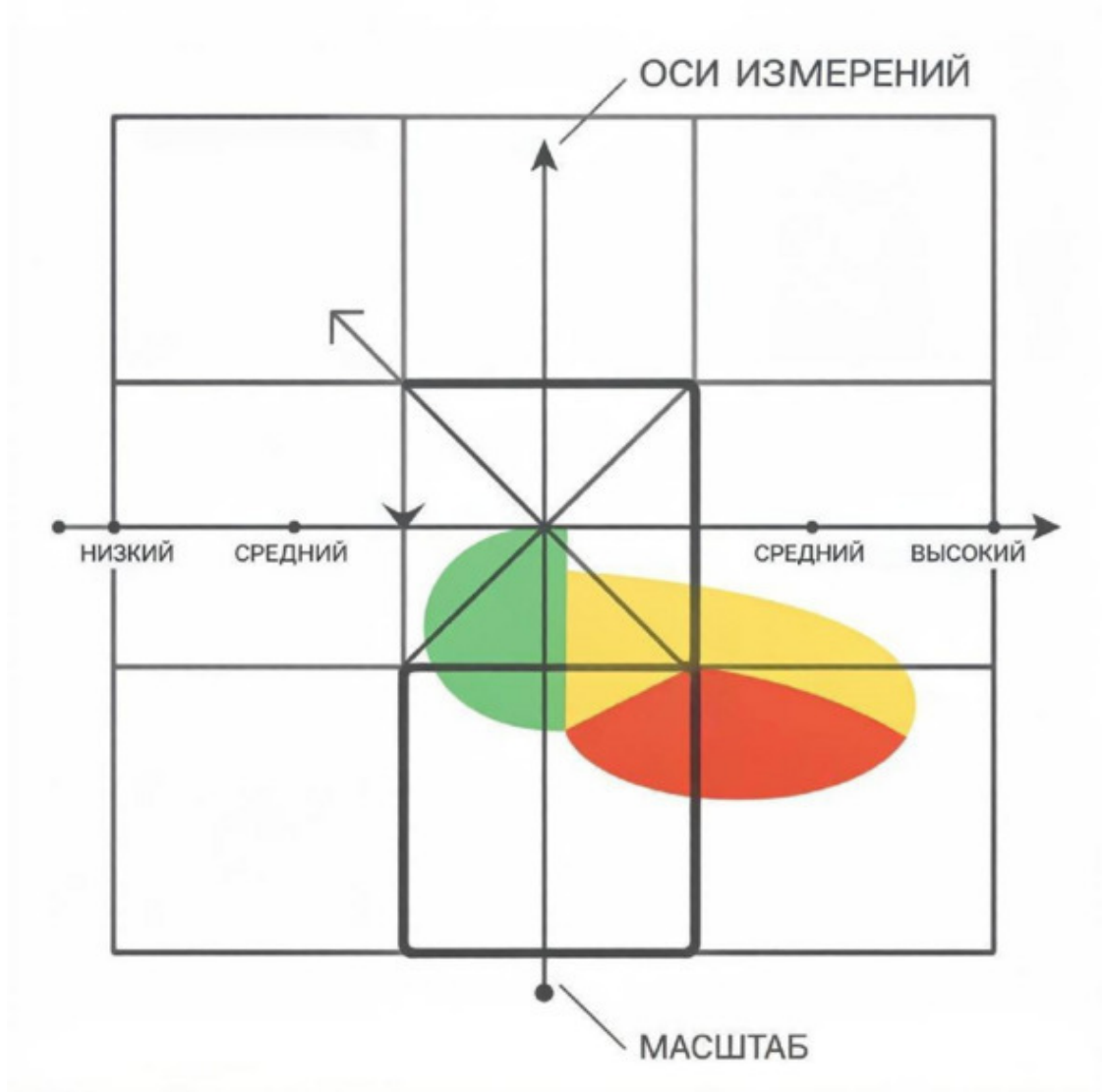


Рисунок 2 Элементы любой сетки понятий: оси задают измерения, шкалы — градации, ячейки — кванты смысла, зоны — области решений.

## 1.1. Оси — измерения смысла

Ось — это критерий, параметр или измерение, которое мы выбираем как значимое для решаемой задачи. Ось задаёт направление смысла: она говорит, на что мы смотрим, когда смотрим на объект.

Оси не извлекаются из данных автоматически. Это не объективные свойства реальности, которые можно вычислить. Это выбор, который делает человек. Для одной и той же совокупности объектов, в зависимости от задачи, содержательными могут оказаться совершенно разные оси. Возьмём восемь сотрудников отдела. Если задача — сформировать кадровый резерв, нас интересуют «Результативность» и «Потенциал». Если задача — оптимизировать фонд оплаты труда, нас интересуют «Вклад в выручку» и «Рыночная стоимость позиции». Данные одни, объекты одни — а оси разные, и сетки получатся разными.

Каждая выбранная ось одновременно и включает, и исключает. Выбирая «Цена — Качество», вы исключаете «Скорость поставки». Выбирая «Риск — Доходность», исключаете «Стратегическую близость». Это нормально. Сетка не может удержать всё. Она ценна именно тем, что принуждает выбрать главное.

Хорошая ось должна обладать тремя свойствами.

**Содержательность.** Ось должна иметь значение для задачи. Если объект находится в крайней точке оси, это должно о чём-то важном говорить. Можно построить сетку сотрудников по осям «Размер обуви» и «Любимый цвет». Формально это сетка. Но для задачи управления она бессодержательна.

**Различимость.** Объекты должны различаться по этой оси. Если для конкретной выборки все значения примерно равны, ось не работает — она не разводит объекты по разным позициям. Например, ось «Прибыльность» для стартапов на pre-seed стадии: почти все убыточны, разброса нет.

**Независимость от другой оси (ортогональность).** В паре две оси должны измерять разные свойства, не выводимые одно из другого. Изменение позиции по оси X не должно автоматически предопределять изменение по оси Y. Если оси зависимы, объекты выстроятся в диагональ, и сетка превратится в ранжир, а не в типологию. Пример нарушения: оси «Опыт» и «Стаж» — они почти всегда растут вместе. Пример правильной пары: «Опыт» и «Обучаемость» — они независимы, возможны все четыре сочетания.

Вопрос ортогональности — один из центральных во всей книге. Мы будем возвращаться к нему неоднократно, особенно в главе 4, посвящённой выбору осей.

## 1.2. Шкалы и деления — структура оси

Если ось задаёт направление смысла, то шкала определяет, с какой точностью мы будем в этом направлении различать объекты.

Шкала — это способ превратить непрерывное или многозначное свойство в набор различных позиций. Деления — это градации на шкале, пороги, которые отделяют одно состояние от другого.

Выбор шкалы — не техническая мелочь. Он определяет, какие различия будут видны, а какие — смазаны.

### Типы шкал.

*Бинарная шкала* (да/нет, есть/нет, высокий/низкий). Простейший случай: два состояния по оси. Принуждает к определённости — объект либо здесь, либо там. Сильная сторона: быстрота и ясность. Слабая: огрубление реальности. Большинство явлений спектральны, и насильственное разделение на два лагеря может исказить картину.

**Порядковая шкала** (низкий/средний/высокий). Наиболее ходовой вариант в управленческих сетках. Три градации дают достаточную чувствительность без переусложнения.  $3 \times 3 = 9$  ячеек — число, которое глаз удерживает легко. Слабость — субъективность границ: где кончается «средний» и начинается «высокий»?

**Количественная шкала** (числовая, с шагом). Использует числа: проценты, баллы, натуральные единицы. Дает точность позиционирования и позволяет измерять расстояния между объектами. Опасность — иллюзия объективности: число маскирует субъективность измерения.

**Логика шкалы.** Помимо типа, важно, как внутри шкалы распределены значения.

— **Линейная шкала:** равные отрезки соответствуют равным приращениям свойства (1, 2, 3, 4, 5). Подходит, когда важны абсолютные различия.

— **Нелинейная шкала:** каждый шаг означает порядок величины (1, 10, 100). Полезна, когда распределение объектов сильно скошено — например, размер компаний (много малых, мало гигантских). Линейная шкала соберёт почти всех в нижних делениях и не даст различий. Логарифмическая раздвинет их.

— **Психологическая шкала:** построена на восприятии (оценки удовлетворённости, NPS). Здесь равные шаги не обязаны быть равными математически, но отражают внутреннюю градацию ощущений.

Выбор логики шкалы — не математическое упражнение. Это решение о том, что считать «существенным различием». Для стартапа разница между нулём и одним клиентом может быть огромной, и шкала должна это отражать.

**Количество делений.** Правило, проверенное практикой: 2—5 делений на ось. Меньше двух — нет оси. Два деления — для быстрых решений. Три — золотая середина для типологии. Четыре-пять — верхняя граница обозримости. Семь и больше — сетка перестанет схватываться глазом и превращается в таблицу. Это не запрет, но предупреждение: вы теряете навигационную силу.

**Определение границ.** Для бинарных и порядковых шкал необходимо провести линии, разделяющие «низкий» и «средний». Это можно сделать экспертно («считаем высокой результативностью выполнение плана от 100%»), квантильно (верхние 25% — высокий), нормативно (относительно стандарта должности) или порогово (выручка > X). Метод выбирается под задачу.

### 1.3. Ячейки — кванты смысла

Ячейка — результат пересечения деления одной оси с делением другой. Минимальная единица карты, в которую попадает объект.

Функция ячейки — присвоить объекту тип или статус. «Звезда» в BCG, «кризис-менеджер» в матрице компетенций, «чемодан без ручки» в портфельном анализе — всё это имена ячеек. Имя закрепляет смысл в памяти и облегчает коммуникацию.

Ячейки могут быть плотно заселёнными, пустыми или содержать единичные объекты.

**Пустая ячейка** — не дефект сетки, а сигнал. Если ячейка пуста, возможны два объяснения. Первое: она «красная» — объекты с такими характеристиками физически или логически невозможны. Высокое качество при нулевой цене, низкий риск при астрономической доходности — таких сочетаний не бывает, и пустота здесь не упущение, а граница реальности. Второе: она «зелёная» — объекты возможны, но никто их не создал. Это либо упущенная возможность, либо будущая инновация. Пустая ячейка «Пенсионеры — цифровой продукт» в матрице клиентов и продуктов двадцать лет назад была зелёной пустотой, из которой вырос целый рынок.

**Переполненная ячейка** — сигнал о том, что сетка не работает. Если почти все объекты сбились в одну ячейку, оси не различают выборку. Нужно либо увеличить число делений, либо признать, что выбранные оси нерелевантны.

## **1.4. Зоны — области с общим решением**

Зона — группа ячеек, для которой мы задаём единое правило, стратегию или вывод. Если ячейки отвечают на вопрос «где находится объект», то зоны — на вопрос «что с этим делать».

Классический пример — матрица Эйзенхауэра. В ней четыре ячейки, но они же являются и зонами: «Делай сразу», «Планируй», «Делегируй», «Отложи». Однако в более крупных сетках зоны могут объединять несколько ячеек. В матрице 3×3 рисков мы можем выделить три зоны: «Приемлемо», «Требуется контроль», «Недопустимо» — и провести границы не по линиям ячеек, а диагонально или ступенчато.

Зонирование — это перевод позиции в действие. Именно оно превращает аналитическую карту в управленческий инструмент. Без зон сетка остаётся описанием. С зонами она становится предписанием.

Важно: зоны не выводятся автоматически из данных. Это волевое решение. Одну и ту же сетку можно разбить на зоны по-разному, в зависимости от стратегии, аппетита к риску и контекста.

## **Итог главы**

Четыре элемента — оси, шкалы, ячейки, зоны — образуют алфавит метода. Оси задают измерения. Шкалы структурируют каждое измерение. Ячейки фиксируют типы и фигуры. Зоны связывают позицию с решением. Каждый элемент настраивается, и каждая настройка содержательна. Сетка понятий — не готовая форма, а конструктор, собранный под задачу.

В следующей главе мы расширим оптику: рассмотрим, какие формы может принимать сетка — от классических 2×2 до радиальных и иерархических структур.

## Глава 2. Типология сеток

В предыдущей главе мы разобрали элементы, из которых состоит сетка: оси, шкалы, ячейки, зоны. Эти элементы можно собрать в разные формы — так же, как из одних и тех же строительных материалов получаются и хижина, и собор.

Форма сетки — это не вопрос эстетики. Это вопрос соответствия задаче. Одна задача требует жёсткого выбора «или/или» и ложится в  $2 \times 2$ . Другая — плавного позиционирования и просит координатную плоскость. Третья — сравнения профилей и легче всего решается радаром. Четвёртая — вложенных категорий и естественно укладывается в дерево.

Ошибка — использовать одну и ту же форму под все задачи. Это как забивать шурупы молотком: технически возможно, но результат не тот.

В этой главе мы рассмотрим пять типов сеток: дихотомическую, координатную, радиальную, иерархическую и многомерную (последнюю — обзорно, поскольку ей посвящена вся Часть 3). Для каждого типа определим, как он устроен, в чём его сильные и слабые стороны и для каких задач он подходит.

### 2.1. Дихотомическая сетка ( $2 \times 2$ , $3 \times 3$ , $N \times M$ )

Это самая узнаваемая форма сетки. Две оси, каждая разбита на дискретные деления — чаще всего два или три. Объекты попадают в ячейки на пересечениях.

Классическая  $2 \times 2$  даёт четыре ячейки. Она принуждает к выбору: каждый объект занимает одну из четырёх позиций. Никаких «чуть-чуть», никакого «посередине» (если вы не сделали среднее значение осознанным делением). Именно эта жёсткость — её сила. Когда вам нужно принять решение,  $2 \times 2$  отрезает путь к отступлению: «Ты либо здесь, либо там. Выбери».

Вариант  $3 \times 3$  даёт девять ячеек — достаточно, чтобы смягчить бинарность без потери обозримости. Девять ячеек глаз охватывает легко, имена им придумываются без труда.

Общий случай  $N \times M$  — любое количество делений на осях. Но практический предел, как мы уже говорили, — около пяти делений на ось. Дальше сетка превращается в таблицу и теряет навигационную силу.

**Сильные стороны.** Простота восприятия. Лёгкость именования ячеек. Принуждение к определённости. Хорошая коммуникационная форма:  $2 \times 2$  понятна любой аудитории за десять секунд.

**Слабые стороны.** Огрубление реальности. Не все объекты хотят укладываться в прокрустово ложе дискретных ячеек. Потеря нюанса: два объекта в одной ячейке выглядят одинаковыми, но могут быть весьма различны внутри неё.

**Когда применять.** Когда нужно быстрое решение, типология или приоритизация. Когда важнее определённости и коммуникационная ясность, чем высокая точность позиционирования.

### 2.2. Координатная (непрерывная) сетка

В дихотомической сетке объект попадает в ячейку. В координатной — он размещается в точке на плоскости, где оси непрерывны или имеют достаточно мелкий шаг.

Никаких ячеек в явном виде здесь нет. Есть пространство, и в нём — точки. Расстояние между точками несёт смысл: два близких объекта похожи по выбранным измерениям, два далёких — различны.

Типичный пример — пузырьковая диаграмма: ось X, ось Y, а размер пузырька добавляет третье измерение. BCG в её классическом виде — именно координатная сетка: доля рынка и темп роста непрерывны, продукты размещаются точками.

**Сильные стороны.** Точность позиционирования. Возможность видеть расстояния и близость. Хорошая форма для выявления кластеров и выбросов визуально: глаз сразу замечает скопления точек и одинокие фигуры.

**Слабые стороны.** Сложнее интерпретировать без подготовки. Нельзя сказать «это звезда» — нужно описать: «доля 0,8, рост 12%». Коммуникация требует перевода с языка координат на язык смыслов.

**Когда применять.** Когда объекты сравнимы количественно. Когда их много (десятки и сотни). Когда важны не только типы, но и расстояния между объектами. Когда вы хотите увидеть кластеры и выбросы, а не наложить заранее заданную сетку ячеек.

## 2.3. Радиальная (центр-лучевая) сетка

Это отход от привычного прямоугольника. Структура: центр и лучи-оси, расходящиеся из него. Значения откладываются на лучах. Объект представлен точками на каждом луче, которые соединяются замкнутой линией — получается многоугольник, «профиль» объекта.

Классический пример — радарная диаграмма компетенций: пять-семь лучей, каждый — компетенция, и на каждом отмечен уровень владения. Другой пример — колесо жизненного баланса, где лучи — сферы жизни, а значения — удовлетворённость.

Важное свойство радиальной сетки: она предназначена для сравнения профилей нескольких объектов. Вы накладываете два-три многоугольника на одну диаграмму и сразу видите, где один «шире» другого.

**Сильные стороны.** Исключительно наглядна для сравнения нескольких объектов по многим измерениям одновременно. Даёт целостный образ профиля.

**Слабые стороны.** Лучи не обязаны быть ортогональны, и это часто приводит к проблеме: можно взять два сильно пересекающихся критерия и создать иллюзию многомерности там, где её нет. «Ответственность» и «Дисциплина» как отдельные лучи — это одно и то же, названное разными словами. Кроме того, порядок лучей влияет на форму фигуры, что может сбивать с толку. Площадь фигуры не имеет прямого аналитического смысла, хотя глаз пытается её интерпретировать.

**Когда применять.** Когда нужно сравнить несколько объектов по единому набору измерений. Когда аудитория привыкла к визуализации профилей. Хороша для оценки сотрудников, сравнения продуктов, анализа конкурентов по множеству параметров.

## 2.4. Иерархическая сетка (дерево)

Дерево обычно противопоставляют сетке. Но с точки зрения нашего метода это частный случай: иерархия — это последовательное применение бинарных или множественных делений по разным осям на каждом уровне.

Пример: сначала делим всё живое по оси «живое/неживое». Затем живое — по оси «животное/растение». Затем животных — по оси «позвоночные/беспозвоночные». На каждом шаге это дихотомия, просто вложенная в предыдущую.

Формально дерево можно представить как сетку, где каждая развилка — это ось, а ветви — деления. Преимущество такого взгляда в том, что мы перестаём видеть в дереве нечто принципиально иное и начинаем видеть в нём конструктор из тех же элементов: оси, деления, ячейки.

**Сильные стороны.** Привычность. Хорошо работает для классификаций, онтологий, любых систем, где есть отношение «общее — частное».

**Слабые стороны.** Жёсткость. Реальный объект может принадлежать к нескольким ветвям одновременно, а дерево принуждает выбрать одну. Дерево плохо работает с перекрёстными связями.

**Когда применять.** Когда объекты естественно вложены друг в друга (категории товаров, организационная структура, таксономии). Когда важна иерархия наследования свойств.

## 2.5. Многомерная сетка (более двух измерений) — анонс

Два измерения визуализируются на плоскости легко. Три — с трудом. Четыре и больше — глаз не видит вовсе. Это физиологическое ограничение: сетчатка двумерна.

Но реальные задачи часто требуют учёта трёх, четырёх, пяти переменных одновременно. Сотрудник — это не только результативность и потенциал. Клиент — не только чек и частота. Проект — не только риск и доходность. Возникает необходимость в многомерной сетке.

Решение этой проблемы — предмет всей Части 3. Здесь мы лишь зафиксируем: многомерная сетка не является отдельным типом в том же смысле, что предыдущие четыре. Это способ работать с несколькими измерениями, упаковывая их в обозримые формы: серии двумерных проекций, цветовые и размерные кодировки, параллельные координаты, радары, кубические разложения.

Многомерная сетка — следующий уровень сложности после двумерной. Но логика остаётся той же: оси, шкалы, ячейки, зоны. Просто осей больше двух.

## Итог главы

Пять типов сеток — это не альтернативы, из которых нужно выбрать одну на всю жизнь. Это палитра. Одна и та же задача на разных этапах может потребовать разных форм: начали с дерева для первичной классификации, перешли к 2×2 для принятия решения, затем к радару для сравнения финалистов.

Выбор типа сетки — такое же содержательное решение, как и выбор осей. Хороший проектировщик знает возможности каждого типа и подбирает форму под задачу, а не наоборот.

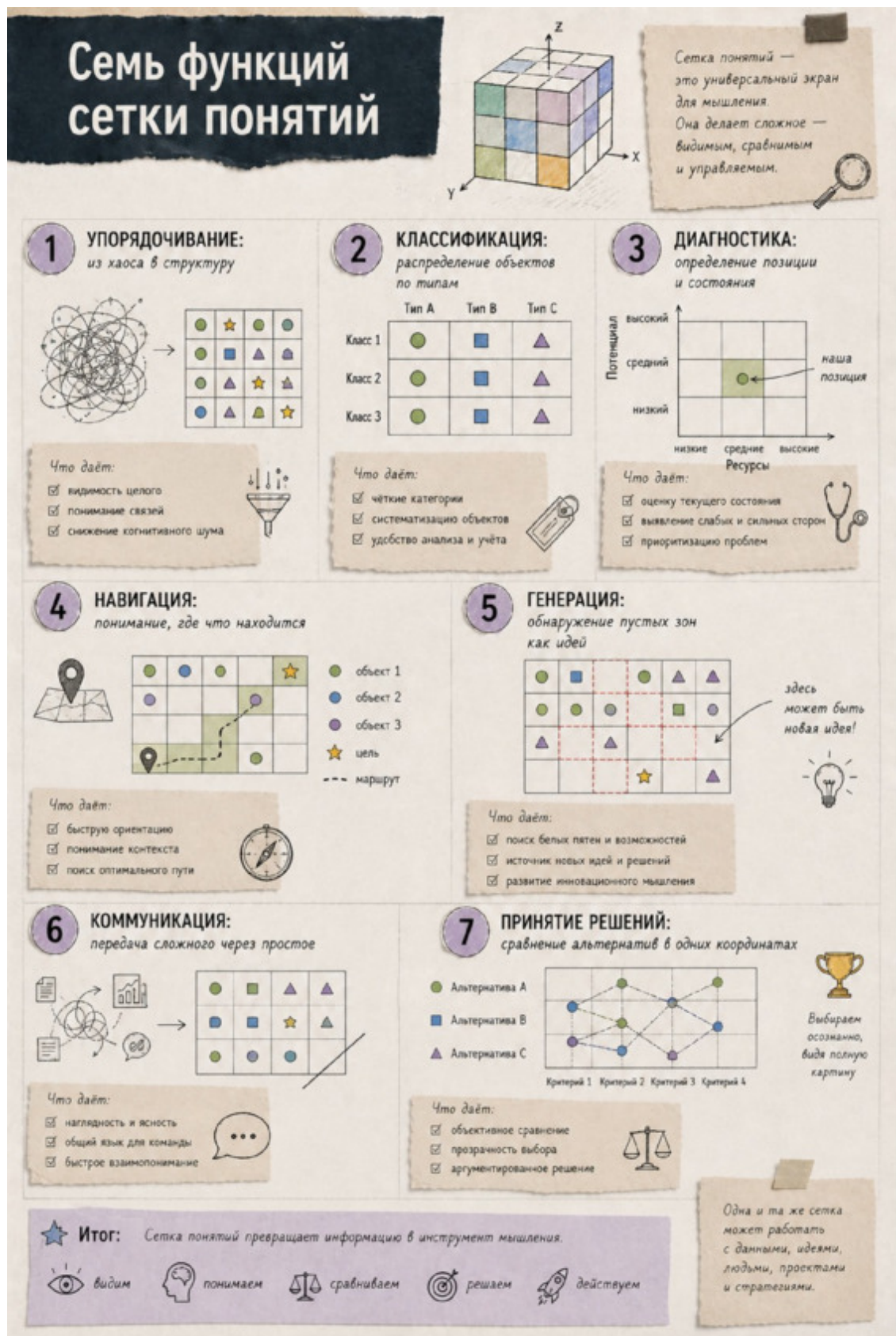
Теперь, когда мы понимаем, из чего сетка состоит (Глава 1) и какие формы может принимать (Глава 2), мы можем спросить: зачем вообще её строить? Какие функции она выполняет в нашем мышлении? Об этом — Глава 3.

### **Глава 3. Зачем нужна сетка понятий: семь функций**

Сетку можно разобрать на элементы (Глава 1) и собрать в разные формы (Глава 2). Но остаётся главный вопрос: что именно она делает? Зачем тратить время на её построение?

Практики, работающие с матрицами, редко отвечают на этот вопрос развёрнуто. Обычно говорят: «Чтобы структурировать информацию» — и переходят к делу. Но «структурировать» — это слишком общо. Сетка выполняет не одну, а несколько разных функций, и каждая из них решает свою когнитивную или управленческую задачу. Если вы не различаете эти функции, вы рискуете построить сетку, которая хорошо выполняет одну (например, классифицирует), но проваливает другую (например, не помогает принять решение).

В этой главе мы выделим семь функций, которые сетка понятий способна выполнять. Для каждой — суть, пример и вопрос для проверки: «Как понять, что эта функция не сработала?»



### 3.1. Упорядочивание: из хаоса в структуру

Это первая, самая очевидная функция. На входе — список из  $N$  объектов, которые никак не соотносятся друг с другом в сознании. На выходе — та же совокупность, но размещённая в ячейках так, что видна общая архитектура.

Упорядочивание само по себе снижает когнитивную нагрузку. Мозг, который держал пятнадцать разрозненных сущностей, теперь держит четыре-девять групп и может оперировать ими осмысленно.

**Пример.** Менеджер по работе с клиентами имеет список из двадцати компаний — текущих и потенциальных заказчиков. В голове они перемешаны. Менеджер накладывает оси «Потенциальная выручка» и «Вероятность сделки» — и двадцать безымянных названий превращаются в четыре группы: «Фокус», «Наращивать», «Наблюдать», «Отложить». Хаос стал структурой.

**Как понять, что функция не сработала.** Вы по-прежнему не можете назвать группы объектов. Сетка есть, но вы продолжаете обсуждать каждый объект по отдельности, не апеллируя к его позиции. Значит, упорядочивание формально, а не содержательно.

### 3.2. Классификация: распределение объектов по типам

Упорядочивание отвечает на вопрос «где что находится». Классификация идёт дальше — она присваивает объектам имена типов на основании их позиции. Каждая ячейка получает имя, и объект, попавший в неё, приобретает этот тип.

Классификация — мощный когнитивный инструмент. Она позволяет перейти от перечисления конкретных объектов к рассуждению о типах. Вместо «Иванов, Петров, Сидоров» — «звёзды», «рабочие лошади», «проблемные».

**Пример.** Сотрудники размещены на сетке «Результативность — Потенциал». Ячейки получают имена. Верхний правый угол (высокая результативность, высокий потенциал) — «Ключевые сотрудники» или «Звёзды». Нижний левый (низкая результативность, низкий потенциал) — «Балласт» или «Зона замены». Имена сразу подсказывают, как работать с каждой группой.

Опасность классификации — навешивание ярлыков без понимания контекста. Объект на границе ячеек может формально получить тип, который не вполне ему соответствует. Хорошая классификация оставляет пространство для пограничных случаев.

**Как понять, что функция не сработала.** Ячейки есть, а имена не присвоены — они остаются «верхним левым» и «нижним правым». Это значит, что классификация не завершена: типы не родились, смысл не упакован в слова.

### 3.3. Диагностика: определение позиции и состояния

Классификация говорит «кто есть кто». Диагностика говорит «что это означает для объекта». Она связывает позицию в сетке с состоянием: норма, тревога, успех, кризис.

Сетка не ставит диагноз сама — она задаёт правильный вопрос. Если продукт оказался «собакой» в BCG, это не приговор, а вопрос: «Почему мы здесь и что с этим делать?» Диагностическая ценность сетки именно в том, чтобы высветить отклонения от желаемого и запустить расследование.

**Пример.** Компания наносит свои проекты на сетку «Стратегическая важность — Отклонение от плана по срокам». Проект в зоне «высокая важность, большое отклонение» — это

диагноз «стратегический провал». Сетка не объясняет причин, но точно указывает, куда смотреть в первую очередь.

**Как понять, что функция не сработала.** Вы смотрите на сетку и не можете сказать, какие объекты в порядке, а какие требуют вмешательства. Позиции определены, но оценка состояния не возникла. Вероятно, зоны не связаны с управленческими выводами.

### 3.4. Навигация: понимание, где что находится

Навигация — это способность сетки служить картой пространства решений. Вы можете указать: «Мы находимся здесь, конкуренты — там, желаемая цель — там». Сетка даёт вид сверху, в котором появляется возможность планировать маршрут.

Функция навигации тесно связана с понятием зон. Зоны — это регионы на карте с разными свойствами. Навигация — это движение по карте с пониманием этих свойств.

**Пример.** Портфель проектов компании на матрице «Риск — Доходность». Проекты сконцентрированы в нижнем левом углу: мало риска, мало доходности. Навигационный вывод: «Мы застряли в безопасной посредственности. Нужны проекты в верхнем правом квадранте». Сетка показала не только текущее положение, но и направление движения.

**Как понять, что функция не сработала.** Сетка описывает текущее состояние, но не даёт ответа на вопрос «куда дальше?». Карта есть, а маршрута нет. Значит, навигационная функция не реализована — сетка осталась статичным снимком.

### 3.5. Генерация: обнаружение пустых зон как идей

Это самая творческая функция сетки, и её часто упускают. Пустая ячейка — не обязательно ошибка. Часто это возможность, которую никто не заметил.

В Главе 1 мы ввели различие «красной» и «зелёной» пустоты. Красная пустота — сочетание, которое физически, экономически или логически невозможно. Высокое качество при нулевой цене — красная пустота. Высокая доходность при нулевом риске — тоже. Тратить время на «освоение» таких ячеек бессмысленно.

Зелёная пустота — сочетание, которое возможно, но по каким-то причинам не реализовано. Никто не подумал. Или существовал технологический барьер, который только сейчас пал. Или рынок был слишком мал, а теперь вырос. Зелёная пустота — источник инноваций.



Рисунок 4 Пустая ячейка — это сигнал. Зелёная пустота: возможность, которую никто не занял. Красная пустота: граница реальности.

**Пример.** Матрица «Тип клиента — Тип продукта». Ячейка «Малый бизнес — Корпоративное решение» пуста. Красная или зелёная? Традиционное рассуждение: малый бизнес не может платить корпоративные деньги. Но если продукт сделать модульным, с оплатой по подписке и без внедрения? Возможно, это зелёная пустота, и здесь скрыт новый сегмент.

**Как понять, что функция не сработала.** Все ячейки заняты, и вы не задали вопрос «почему эта пуста?» ни для одной из пустых. Или, наоборот, все пустые сочли невозможными, не проверив гипотезу «а что, если?». Генерация не работает, когда сетку читают без любопытства.

### 3.6. Коммуникация: передача сложного через простое

Сетка — это высокоплотная упаковка смысла. Одна диаграмма заменяет несколько страниц текста. Десять минут доклада можно сжать в один взгляд.

Но коммуникационная сила сетки не только в сжатии. Она меняет характер обсуждения. Когда перед участниками совещания лежит сетка, они спорят не о «правильности подхода», а о позициях объектов и границах зон. Это более предметный, более конструктивный спор.

**Пример.** Вместо отчёта о состоянии рынка на пятнадцати страницах аналитик показывает матрицу «Темп роста — Концентрация отрасли» с нанесёнными на неё конкурентами. За три секунды видны лидеры, нишевые игроки и аутсайдеры. Обсуждение начинается не с «расскажите, что вы имели в виду», а с вопроса: «Почему игрок X расположен здесь? Он точно не левее?» — и это продуктивно.

**Как понять, что функция не сработала.** Сетку показывают, но аудитория либо не задаёт вопросов, либо вопросы касаются оформления, а не смысла позиций. Или сетку приходится долго объяснять устно — значит, она не самодостаточна и коммуникация провалилась.

### 3.7. Принятие решений: сравнение альтернатив в одних координатах

Это функция, ради которой часто строят сетку в первую очередь. Все альтернативы оцениваются по единым осям, и вместо вкусового спора возникает сопоставление позиций.

Сетка не принимает решение за вас. Она формализует аргументы. Если два человека смотрят на одну сетку и спорят, куда поставить точку, — это содержательный спор, потому что он идёт о фактах и прогнозах. Если они спорят без сетки, они часто просто обмениваются предпочтениями: «мне кажется, этот вариант лучше». Сетка переводит обсуждение в плоскость критериев.

**Пример.** Стратегический комитет выбирает между пятью вариантами развития. Оси: «Ожидаемая доходность» и «Вероятность успеха». Каждый вариант получает позицию на сетке. Спор переходит от абстрактного «вариант Б лучше» к конкретному: «Почему вы поместили вариант Б правее А? На чём основана оценка вероятности?» Это уже не лозунги, а анализ.

**Как понять, что функция не сработала.** Сетка построена, но решение принимается так же, как без неё — интуитивно или голосованием. Или, наоборот, итоговый балл матрицы решений воспринимается как диктат, без анализа чувствительности и обсуждения пограничных случаев. Сетка либо не повлияла, либо заменила мышление, вместо того чтобы его дисциплинировать.

## Итог главы

Семь функций — это не семь разных применений сетки, а семь слоёв, которые могут присутствовать в одной и той же сетке одновременно. Хорошо спроектированная сетка упорядочивает, классифицирует, даёт диагноз, показывает маршрут, подсвечивает упущенные возможности, сжимает коммуникацию и поддерживает решение. Все семь — в одной конструкции.

Если какая-то функция не работает, это не значит, что метод плох. Это значит, что проектировщик не настроил соответствующий элемент. Упорядочивание не работает — проблема в осях. Классификация не работает — не присвоены имена ячейкам. Навигация не работает — не определены зоны. Генерация не работает — пустые ячейки проигнорированы.

Такой взгляд на функции даёт вам инструмент самопроверки: построив сетку, пройдитесь по семи пунктам и спросите себя — эта функция реализована? Если нет — что именно нужно донастроить?

## Итог Части 1

Мы начали с того, что разобрали сетку понятий на составные части. Оси задают измерения, шкалы структурируют каждое измерение, ячейки фиксируют типы, зоны связывают позицию с решением. Эти четыре элемента — алфавит метода: из них собирается любая матрица, любой радар, любое дерево.

Затем мы вышли за пределы привычного 2×2. Сетка бывает дихотомической и координатной, радиальной и иерархической, а при необходимости — многомерной. Выбор формы — такое же содержательное решение, как и выбор осей: форма должна соответствовать задаче, а не наоборот.

Наконец, мы увидели, что построенная сетка способна выполнять не одну, а несколько функций одновременно — от упорядочивания до поддержки решений. Каждая функция описывается на конкретный элемент конструкции, и это даёт инструмент самопроверки: если что-то не работает, мы знаем, что именно настраивать.

Теперь, зная устройство сетки, мы готовы к главному — её проектированию. Об этом — Часть 2.

## ЧАСТЬ 2. ПОСТРОЕНИЕ ДВУМЕРНЫХ СЕТОК

В Части 1 мы разобрали сетку понятий как конструкцию: оси, шкалы, ячейки, зоны; пять типов форм, от дихотомической до многомерной; семь функций, которые сетка способна выполнять. Это знание необходимо, но недостаточно. Знать анатомию — не значит уметь строить.

Построение сетки — это не заполнение готового шаблона, а последовательность содержательных решений. Каждое из них влияет на результат. Пропуск одного шага или бездумное следование привычному образцу часто приводит к «сетке-пустышке»: нарисовано аккуратно, понимания не прибавилось.

В этой части мы пройдем весь путь от формулировки задачи до проверки готовой сетки. Глава 4 посвящена самому ответственному шагу — выбору осей. Глава 5 — настройке шкал и делений. Глава 6 — размещению объектов в сетке. Глава 7 даёт разбор десяти классических матриц, чтобы вы видели не магию, а чужие проектировочные решения. Глава 8 собирает всё в пошаговый алгоритм и проводит через него на сквозном кейсе.

Двумерная сетка — фундамент. Даже когда мы перейдем к многомерным построениям в Части 3, мы будем сводить их к серии двумерных проекций. Освоив эту часть, вы получите главное: способность спроектировать работающую сетку под любую задачу, не подгоняя её под готовый шаблон.

Начнем с самого важного шага.

## Глава 4. Выбор осей: самый важный шаг

Если бы меня попросили назвать единственное решение, которое определяет судьбу сетки, я бы без колебаний сказал: выбор осей. Всё остальное — шкалы, ячейки, зоны, интерпретации — можно поправить потом; неправильные оси не исправляются ничем.

Почему это так — разберём в первом разделе главы. Затем я дам четыре принципа, которыми должны удовлетворять хорошие оси. После этого опишу пошаговую технику поиска двух измерений — как из пятнадцати факторов получить работоспособную пару. Завершим разбором типичных ошибок.

### 4.1. Почему выбор осей решает всё

Оси — это линзы, через которые мы смотрим на реальность. Выбор осей определяет, что мы увидим, а что останется невидимым.

Представьте, что вы анализируете команду из десяти сотрудников. Если вы возьмёте оси «Стаж — Образование», вы увидите картину опыта и квалификации. Если «Результативность — Инициативность» — картину сегодняшней и будущей эффективности. Если «Лояльность — Стрессоустойчивость» — картину надёжности в кризис. Данные одни и те же, объекты одни и те же — но сетки дают три принципиально разных взгляда на команду. Каждый из них по-своему верен и каждый что-то исключает.

Это фундаментальное свойство двумерной сетки: она одновременно удерживает только два измерения. Выбор этих двух — акт стратегической фокусировки. Вы не просто говорите: «Эти критерии важны». Вы говорите: «Эти критерии настолько важны, что ради них я готов на время отставить всё остальное».

Поэтому выбор осей — не аналитический, а проектировочный шаг. Вы не находите оси в данных. Вы решаете, что будет главным, — и только потом проверяете, можно ли по выбранным осям осмысленно разместить объекты.

Когда выбор осей сделан неправильно, последствия не компенсируются ничем. Если оси зависимы — сетка превращается в диагональную линию. Если оси не различают объекты — все сбиваются в одну ячейку. Если оси не содержательны — сетка выглядит красиво, но не даёт инсайта. Поэтому следующий раздел — о принципах, позволяющих избежать этих ловушек.

### 4.2. Четыре принципа хороших осей

Хорошая пара осей должна удовлетворять четырём требованиям. Три из них относятся к каждой оси по отдельности, четвёртое — к их соотношению.

#### **Принцип 1. Независимость (ортогональность)**

Две оси должны измерять разные свойства, не выводимые одно из другого. Изменение позиции объекта по оси X не должно автоматически предопределять изменение по оси Y.

Проверочный вопрос: «Может ли объект иметь высокое значение по X и низкое по Y — и наоборот?» Если ответ «нет» для обоих сочетаний, оси зависимы. Вы измеряете одно и то же двумя способами. Классические примеры нарушения: «Опыт» и «Стаж» — в большинстве профессиональных сред они растут вместе; «Размер компании» и «Бюджет на рекламу» — сильно коррелируют; «Качество» и «Цена» — в некоторых категориях связаны почти жёстко.

Последствия зависимости: объекты выстраиваются вдоль диагонали. Ячейки вне диагонали пустуют не потому, что невозможны, а потому что в данной выборке таких сочетаний не оказалось. Сетка перестаёт быть типологией и становится ранжиром.

#### **Принцип 2. Полнота**

Все объекты, которые вы собираетесь анализировать, должны иметь определённое значение по каждой из осей. Не должно быть объектов, про которые вы говорите: «По этой оси его оценить невозможно».

Если вы анализируете эффективность каналов продаж и берёте ось «Проходимость торговой точки», то онлайн-каналы не имеют этой характеристики. Их нельзя разместить в сетке, они выпадают. Либо ось нужно заменить, либо честно ограничить область анализа.

Неполнота часто возникает, когда ось формулируют слишком узко, привязав её к конкретной метрике, которая есть не для всех объектов. Лечение: формулировать ось как общее измерение, а метрики подбирать под неё для разных групп объектов отдельно.

### **Принцип 3. Различимость**

Объекты должны различаться по каждой оси. Если все объекты имеют примерно одинаковое значение, ось не работает — она не разводит их по разным позициям.

Представьте, что вы анализируете стартапы на самой ранней стадии и берёте ось «Прибыльность». Почти все убыточны. Ось формально есть, а содержательно — все находятся в одной точке. Для этой выборки различность нулевая.

Практический приём: прежде чем утверждать оси, сделайте грубую прикидку. Возьмите пять-десять объектов и мысленно разбросайте их по шкале. Если все скопились в одной половине, ось, вероятно, неразличительна для данной задачи.

### **Принцип 4. Содержательность**

Оси должны иметь смысл для решаемой задачи. Это не статистическое, а смысловое требование. Можно построить сетку сотрудников по осям «Размер обуви» и «Любимый цвет» — она будет и независимой, и полной, и, скорее всего, различительной. Но для задачи управления персоналом она бессодержательна.

Проверочный вопрос: «Если объект находится в крайней левой точке оси X — это о чём-то важном для задачи говорит?» Если крайние значения не несут управленческого смысла — ось не содержательна. «Любимый цвет — синий» не говорит о сотруднике ничего, что помогло бы принять кадровое решение. «Результативность — низкая» говорит.

Четыре принципа работают как фильтр. Выдвинули гипотезу осей — проверьте каждую по отдельности на содержательность, полноту и различность, а пару — на независимость. Если какая-то проверка не пройдена — возвращайтесь и ищите другую комбинацию.

## **4.3. Техника «Поиск двух измерений»**

Принципы — это фильтр. Техника — это способ сгенерировать кандидатов на фильтрацию. Она превращает выбор осей из озарения в последовательную процедуру.



Рисунок 5 Техника «Поиск двух измерений»: от разрозненных факторов к ортогональным осям.

### Шаг 1. Выписать все факторы

Сформулируйте задачу. Затем выпишите все факторы, которые могут иметь значение для её решения. Без цензуры, без оценки важности — сплошным списком. Для задачи «Оценить сотрудников для кадрового резерва» список может выглядеть так: результативность, стаж, обучаемость, инициативность, лояльность, возраст, образование, коммуникабельность, стрессоустойчивость, опыт управления, знание отрасли, текучесть проектов, зарплата, рекомендации руководителя. Обычно набирается от десяти до двадцати пяти факторов.

На этом шаге главное — не останавливать себя соображениями «это неважно» или «это невозможно измерить». Сначала соберите всё, потом отфильтруете.

### Шаг 2. Сгруппировать родственные факторы

В длинном списке почти всегда есть факторы, которые говорят об одном и том же разными словами. «Обучаемость» и «Гибкость мышления». «Лояльность» и «Срок работы в компании». «Инициативность» и «Проактивность». Объедините их в смысловые кластеры. Дайте каждому кластеру рабочее имя — не «Фактор 3», а «Гибкость», не «Кластер А», а «Результативность».

Обычно из пятнадцати-двадцати факторов получается от четырёх до семи кластеров. Это уже обозримое число кандидатов на роль осей.

### Шаг 3. Выбрать два максимально ортогональных кластера

Теперь — попарное сравнение. Для каждой пары кластеров задайте проверочный вопрос: «Может ли объект быть высоким по первому и низким по второму — и наоборот?» Ищите пару, где ответ «да» максимально очевиден для обоих сочетаний.

Пример. Кластер «Результативность» и кластер «Потенциал». Может ли сотрудник давать высокие результаты сейчас, но иметь низкий потенциал роста? Да, «рабочая лошадка». Может ли иметь высокий потенциал, но низкую текущую результативность? Да, «новичок» или «недоиспользованный талант». Оси ортогональны.

Контрпример. Кластер «Результативность» и кластер «Опыт». В конкретной профессиональной среде они могут быть связаны почти жёстко: результативность растёт с опытом. Тогда ответ «нет», и эту пару отмечаем.

#### **Шаг 4. Проверить выбранную пару по четырём принципам**

Прогоните кластер-кандидат через фильтр: содержательность, полнота, различность для каждого кластера и независимость для пары. Если одна из осей не прошла проверку — вернитесь к шагу 3 и выберите следующую пару.

#### **Шаг 5. Дать осям рабочие имена**

Кластер — это временное рабочее понятие. Ось должна получить имя, которое будет понятно тем, кто увидит сетку. Имя должно быть содержательным и кратким. «Результативность», а не «Кластер эффективности текущей деятельности». «Потенциал», а не «Способность к росту и развитию в обозримой перспективе».

Имя оси — это не формальность. Оно задаёт рамку интерпретации. Два разных имени для одной и той же шкалы могут повести обсуждение в разные стороны.

**Упражнение.** Ниже — десять факторов, описывающих сотрудника. Найдите два наиболее ортогональных кластера и предложите оси.

Список: стаж работы; выполнение КПЭ за последний год; скорость освоения новых инструментов; коммуникабельность; число успешно завершённых проектов за последние два года; зарплата; возраст; лояльность компании; инициативность; текучесть задач.

Попробуйте сделать это прямо сейчас, прежде чем читать разбор.

**Разбор.** Сначала группируем факторы в кластеры. Выполнение КПЭ и число успешных проектов тяготеют к одному смыслу — «Результативность». Скорость освоения нового, инициативность, коммуникабельность — к «Потенциалу» или «Обучаемости». Стаж, возраст — к «Опыту». Лояльность — сама по себе. Зарплата — сама по себе, и для задачи кадрового резерва не содержательна.

Теперь попарно. Пара «Результативность — Потенциал» ортогональна: возможны все четыре сочетания. Пара «Результативность — Опыт» — под вопросом: в конкретной среде они могут коррелировать. Пара «Потенциал — Лояльность» — тоже ортогональна, и это возможный альтернативный выбор, если задача — не кадровый резерв, а, скажем, формирование антикризисной команды.

Здесь ключевой мета-урок: выбор осей зависит не только от факторов, но и от задачи. Для кадрового резерва содержательны «Результативность — Потенциал». Для удержания ключевых людей в трудные времена — «Результативность — Лояльность». Факторы одни, задачи разные — оси разные. Это не ошибка, а суть метода.

### **4.4. Типичные ошибки при выборе осей**

Разберём пять ошибок, которые встречаются чаще всего. Каждая из них способна обесценить всю последующую работу.

#### **Ошибка 1. Зависимые оси (псевдоортогональность)**

Симптом: объекты выстраиваются в диагональ от нижнего левого к верхнему правому. Причина: оси измеряют одно и то же свойство, названное разными словами. Пример: «Знания» и «Опыт» в конкретной среде растут одновременно, и сетка не создаёт типологии — она ранжирует. Последствие: иллюзия анализа. Кажется, что мы что-то разложили, а на деле просто расставили объекты по ранжиру. Лечение: задать проверочный вопрос из раздела 4.2.

#### **Ошибка 2. Пустые оси**

Симптом: по одной из осей все объекты находятся примерно в одном и том же месте. Причина: ось в принципе содержательна, но для данной выборки не различает. Пример: «Прибыльность» для стартапов на pre-seed стадии. Лечение: заменить на ось, по которой есть разброс в данной выборке, либо изменить состав объектов.

#### **Ошибка 3. Слишком много делений на раннем этапе**

Симптом: проектировщик сразу задаёт шкалу от одного до десяти. Причина: желание точности. Последствие: при малом количестве объектов сетка становится разреженной пустыней — три объекта на сто ячеек. Интерпретировать нечего. Лечение: начинать с 2×2 или 3×3, а детализацию добавлять позже, когда станет ясна структура.

#### **Ошибка 4. Смещение оси и деления**

Симптом: вместо оси «География рынка» проектировщик сразу задаёт «Россия/Зарубежье». Причина: мышление готовыми категориями. Последствие: теряется возможность переключить шкалу. «Россия/Зарубежье» может оказаться слишком грубо, но ось «География» позволяет уточнить: регионы, кластеры, страны. Лечение: сначала называть ось как измерение, потом — как её делить.

#### **Ошибка 5. Игнорирование задачи**

Симптом: выбраны интересные, ортогональные, но бесполезные для решения оси. Пример: анализировать поставщиков по осям «Возраст компании» и «Количество сотрудников» — это ортогонально и различительно, но для выбора поставщика содержательнее «Надёжность» и «Цена». Причина: проектировщик пошёл от данных, а не от задачи. Лечение: всегда начинать с вопроса «Какое решение мы принимаем?», а не «Что у нас есть из данных?».

### **Итог главы**

Выбор осей — это не анализ данных, а проектировочное решение. Данные помогают проверить оси на различность, но не подсказывают их. Оси выбираются из понимания задачи. Четыре принципа — независимость, полнота, различность, содержательность — дают фильтр для любого набора кандидатов. Техника «Поиск двух измерений» превращает отбор в процедуру, которая не требует озарений. Пять ошибок предупреждают о главных ловушках.

Когда оси выбраны, следующий шаг — решить, как именно мы будем различать объекты по каждой из них. Это предмет следующей главы.

## Глава 5. Определение шкал и делений

Оси выбраны. Мы знаем, по каким критериям будем оценивать объекты. Теперь следующий вопрос: с какой точностью?

Если оси — это направления смысла, то шкалы и деления — это структура каждого направления. Они определяют, насколько подробно мы различаем объекты и где проходят границы между позициями. Этот шаг часто воспринимают как технический — «разбить на низкий/средний/высокий, чего тут думать». Но именно здесь решается, будут ли объекты разведены по разным ячейкам или сольются в одну; возникнут ли пустые зоны; можно ли будет проследить траекторию движения.

### 5.1. Типы шкал

Шкала — это способ превратить неопределённое или непрерывное свойство в набор различаемых позиций. Типов шкал три, и каждый создаёт свою логику восприятия.

**Бинарная шкала (да/нет, есть/нет, высокий/низкий).**

Простейший случай — два состояния. Объект либо здесь, либо там. Никакой середины.

Сила бинарной шкалы в том, что она принуждает к определённости. Вы не можете сказать «умеренно» — вы должны занять сторону. Это ценно для принятия решений: делать/не делать, инвестировать/выходить, свой/чужой. Бинарная шкала отсекает промедление.

Слабость — огрубление реальности. Большинство явлений спектральны: сотрудник не «результативен» или «нерезультативен», он где-то на continuum'e. Жёсткая бинаризация может исказить картину и спровоцировать спор о границе: «Почему это „низкий“, когда тут скорее средний?»

Бинарную шкалу стоит применять, когда нужно волевое решение, когда серая зона вредна, когда объектов немного и они действительно поляризованы.

**Порядковая шкала (низкий/средний/высокий).**

Наиболее частый случай в управленческих сетках. Три градации дают достаточную чувствительность без переусложнения. Девять ячеек (3×3) — число, которое глаз охватывает легко, а смысловые различия между ячейками уже заметны.

Сила — гибкость. Появляется середина, куда можно поместить «типичные» объекты, не натягивая их на крайности.

Слабость — субъективность границ. Где заканчивается «средний» и начинается «высокий»? Два разных эксперта могут разместить один и тот же объект по-разному. Лечение — задать критерии границ заранее (об этом ниже, в разделе 5.3).

**Количественная шкала (числовая, с шагом).**

Здесь ось получает числовые значения: проценты, баллы, натуральные единицы. Объект размещается не в ячейке, а в точке с координатой.

Сила — точность позиционирования. Можно измерить расстояние между двумя объектами, а не только их ранг. Можно использовать пузырьковые и точечные диаграммы.

Слабость — иллюзия объективности. Число маскирует субъективность измерения. Если методика расчёта балла сомнительна, количественная шкала только придаёт видимость точности ненадёжным данным. Опасность «ложной точности»: шкала от 1 до 100, где никто не может объяснить разницу между 63 и 64, дезориентирует, а не проясняет.

Количественная шкала хороша, когда данные действительно измеряются в числах, методика измерения прозрачна, а аудитория готова читать координаты, а не только имена ячеек.

**Логика шкалы: линейная и нелинейная.**

Помимо типа шкалы, важно, как внутри неё распределены значения. Здесь есть содержательный выбор, который часто упускают.

*Линейная шкала* устроена так, что равные отрезки соответствуют равным приращениям свойства. Шаг 1, 2, 3, 4, 5. Переход от 1 к 2 такой же по смыслу, как от 4 к 5. Подходит, когда важны абсолютные различия между объектами.

*Нелинейная шкала* (чаще всего логарифмическая или степенная) устроена иначе: каждый следующий шаг означает порядок величины. Например, 1, 10, 100, 1000. Переход от 10 к 100 и от 100 к 1000 — это разные шаги в абсолютных числах, но одинаковые в логике порядка.

Зачем это нужно? Представьте, что вы анализируете компании по размеру выручки. Малых компаний с выручкой до 10 миллионов — сотни, средних — десятки, крупных — единицы. Если вы возьмёте линейную шкалу с шагом 10 миллионов, почти все объекты соберутся в первом делении, и ось потеряет различительность. Если возьмёте логарифмическую — они распределятся равномерно.

Общее правило: если распределение объектов по свойству сильно скошено (много маленьких значений, мало больших), линейная шкала не даст различий. Подумайте о нелинейной.

*Психологическая шкала* — особый случай. Она основана на восприятии, а не на физической метрике. Шкалы удовлетворённости (1—5), NPS (0—10), оценки по Лайкерту. Здесь равные шаги не обязаны быть равными математически — они отражают внутреннюю градацию ощущений. Переход от «нейтрально» к «скорее удовлетворён» может быть психологически больше, чем от «скорее удовлетворён» к «полностью удовлетворён».

Выбор логики шкалы — не математическое упражнение. Это решение о том, что считать «существенным различием» в данной задаче. Для инвестора разница между стартапом с нулевой выручкой и стартапом с первым миллионом может быть важнее, чем разница между ста миллионами и ста двадцатью. Шкала должна это отражать.

## 5.2. Правило «золотой середины»: количество делений

Сколько делений должно быть на оси? Правило, проверенное практикой: **от двух до пяти**. Меньше двух — нет шкалы, одно деление — это константа, а не измерение. Больше пяти — сетка перестаёт схватываться глазом и превращается в таблицу.

Два деления — для быстрых решений. Четыре ячейки в 2×2 заполняются и интерпретируются мгновенно.

Три деления — золотая середина. Девять ячеек — обозримо, но уже есть нюанс. Большинство управленческих матриц именно таковы.

Четыре деления — 16 ячеек. Ещё обозримо, но требует аккуратной проработки имён и зон.

Пять делений — 25 ячеек. Верхняя граница комфортного восприятия для неподготовленной аудитории. Дальше начинается аналитическая таблица, а не навигационная сетка.

Почему перегружать деления вредно? Три причины.

Первое — когнитивная нагрузка. Рабочая память человека удерживает  $7 \pm 2$  объекта. Сетка из 6×6 (36 ячеек) уже превышает этот порог, и вы теряете главное преимущество метода — быструю обозримость.

Второе — разреженность. Если объектов двадцать, а ячеек сто, большинство ячеек будут пусты. И не потому, что это сигнал (как мы обсуждали в разделе о зелёной пустоте), а просто потому, что делений больше, чем объектов. Вы теряете способность отличать содержательную пустоту от технической.

Третье — ложная точность. Разбивая шкалу на десять градаций, вы создаёте иллюзию, что можете различать объекты с такой точностью. Часто не можете — и тогда точность работает против вас, провоцируя споры о десятых долях, которые не имеют содержательного смысла.

Когда правило можно нарушить? Если объектов действительно много (сотни), и сетка — чисто аналитический инструмент, а не средство коммуникации на совещании. В этом случае 7 или 10 делений могут быть оправданны. Но для навигационных целей держитесь правила 2—5.

### 5.3. Определение границ делений

Для бинарной и порядковой шкал необходимо провести линии, разделяющие «низкий» и «средний», «средний» и «высокий». Это содержательное, а иногда и политическое решение. Где проходит граница — вопрос не техники, а смысла.

Четыре основных метода определения границ.

**Экспертный.** Руководитель или команда договариваются: «Высокой результативностью считаем выполнение плана от 100 процентов». Сила — простота и прозрачность. Слабость — субъективность: другой эксперт мог бы провести черту иначе.

**Квартильный.** Верхние 25 процентов выборки получают статус «высокий», нижние 25 — «низкий», остальные — «средний». Сила — объективность относительно данной совокупности. Слабость — зависимость от состава выборки. Если все сотрудники сильные, квартальный метод всё равно отправит четверть в «низкие», что может быть несправедливо содержательно.

**Нормативный.** Границы привязываются к внешнему стандарту: «Соответствует требованиям должностной инструкции — средний. Превосходит — высокий». Сила — привязка к объективному (хотя бы в теории) стандарту, а не к соседям по выборке. Слабость — нужен сам стандарт, и он должен быть актуален.

**Пороговый.** Задаётся конкретное числовое значение: «Выручка выше 100 миллионов — высокий уровень». Сила — чёткость. Слабость — порог нужно обосновать, и он может устареть.

Ни один метод не является универсально лучшим. Выбор зависит от задачи. Для стратегических решений чаще подходит экспертный и нормативный подходы (они ближе к смыслу). Для аналитики больших выборок — квартильный и пороговый (они ближе к данным). Важно одно: метод должен быть выбран и объявлен до размещения объектов. Иначе возникнет соблазн подогнать границы под желаемый результат.

### Итог главы

Шкала и деления — не раскраска готовой сетки, а содержательный выбор. Тип шкалы (бинарная, порядковая, количественная) задаёт точность и жёсткость позиционирования. Логика шкалы (линейная или нелинейная) определяет, что будет считаться существенным различием. Количество делений (от двух до пяти) удерживает баланс между чувствительностью и обзорностью. Границы делений, определённые заранее и прозрачно, предотвращают манипуляцию позициями.

Теперь у нас есть оси (Глава 4) и шкалы (Глава 5). Следующий шаг — взять реальные объекты и разместить их в построенной системе координат. Об этом — Глава 6.

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.