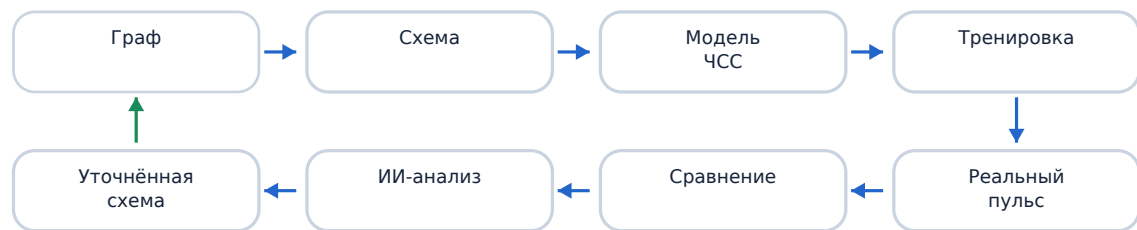


Zaruba Timer как инструмент

Граф, схема и ИИ-анализ в едином цикле тренировки

Документ объясняет, почему в Zaruba Timer совместно используются граф и язык Схема, как из них строится модель ЧСС, как таймер сверяет эту модель с реальным пульсом и почему результат становится не набором безликих сигналов, а полным циклом планирования, выполнения, анализа и уточнения тренировки.

Общая идея: тренировка как цикл модели, выполнения и анализа



каждая тренировка уточняет модель и следующий проход

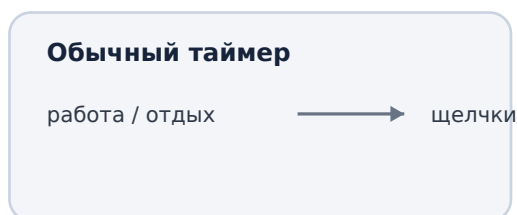
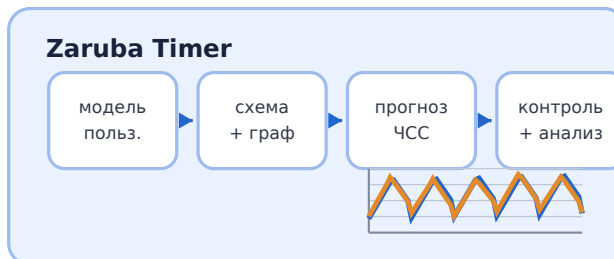
Версия с исправленной вёрсткой иллюстраций.

Содержание

1. Главное отличие: таймер щёлкает, Zaruba Timer моделирует тренировку
2. Почему граф необходим
3. Два уровня модели: пользователь и тренировка
4. Схема: язык описания сложной тренировки
5. Схема как общий язык и формат обмена
6. Трафарет: как схема и граф сливаются в итоговую модель
7. Почему в схеме часто опускаются интенсивности
8. Вариативные схемы как воспроизводимый эксперимент
9. Таймер как инструмент контроля в реальном времени
10. ИИ-файл: переход от метрик к пониманию
11. Итог: Zaruba Timer как единый цикл работы с тренировкой

Введение

Иллюстрация. Таймер и модель



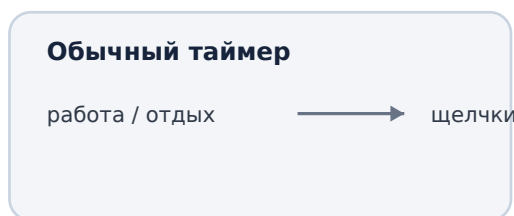
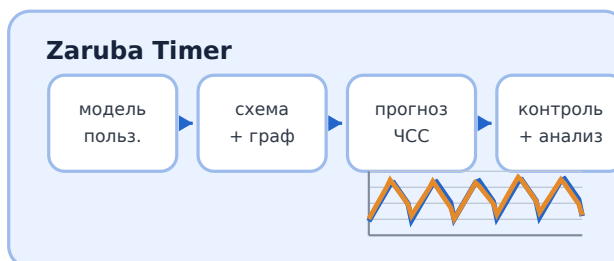
Zaruba Timer - это не просто интервальный таймер и не просто график пульса. Это инструмент, который соединяет визуальное планирование, формальное описание нагрузки, прогноз ЧСС, выполнение тренировки, контроль фактического пульса и последующий анализ результата.

Обычный таймер управляет временем. Он сообщает, когда работать, когда отдыхать и когда закончить. Zaruba Timer работает с более сложной задачей: он отражает тренировочный процесс как модель, исполнение и проверку модели.

В системе есть два главных способа описания тренировки: граф и схема. Граф даёт наглядность, быстрый старт и параметры для построения пульсовой модели. Схема даёт мощный язык описания сложной нагрузки. Вместе они образуют инструмент, который позволяет не только запустить тренировку, но и понять, как она должна протекать, как она реально прошла и что можно уточнить в следующий раз.

1. Главное отличие: таймер щёлкает, Zaruba Timer моделирует тренировку

Иллюстрация 1. Таймер и модель



Обычный интервальный таймер в своей основе выдаёт последовательность сигналов: начать работу, закончить работу, начать отдых, закончить отдых. Даже если у него удобный интерфейс, красивые круги, звуки, пресеты и история тренировок, он всё равно в основном отвечает на один вопрос: когда переключить фазу.

Zaruba Timer решает другую задачу. Он описывает и сопровождает не только внешнюю разметку времени, но и весь процесс тренировки: план нагрузки, прогнозируемую реакцию ЧСС, фактический пульс, отклонение от модели, качество совпадения и последующую корректировку параметров.

Даже простейшая модель ЧСС требует больше информации, чем обычный таймер. Для одного базового интервального атома системе нужны как минимум четыре динамических параметра:

- время работы;
- время отдыха;
- интенсивность работы;
- интенсивность отдыха.

Если тренировка повторяется, добавляется ещё и структура повторения: количество интервалов или более сложное правило развёртывания.

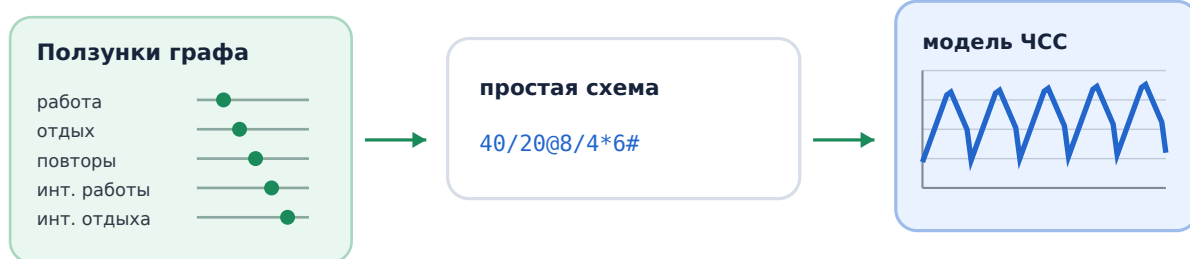
Но этим модель не исчерпывается. Чтобы прогноз ЧСС был связан не с абстрактным человеком, а с конкретным пользователем, система учитывает и персональные статические параметры: пульсовый диапазон, зоны ЧСС, параметры реакции и восстановления, τ , дрейф и другие настройки физиологической модели.

Именно здесь появляется фундаментальное отличие. Стандартный таймер может выполнить тренировку, даже ничего не зная ни об интенсивности, ни о пульсовых зонах, ни о скорости реакции сердца. Ему достаточно знать, когда подать сигнал. Но система, которая строит прогноз ЧСС и затем сравнивает его с реальным пульсом, не может обойтись одной разметкой времени. Ей нужна модель нагрузки и модель пользователя.

Обычный таймер выдаёт безликие щелчки. Zaruba Timer строит модель тренировки, ведёт по ней, сверяет её с реальностью и помогает уточнять следующий проход.

2. Почему граф необходим

Иллюстрация 2. Граф как редактор и источник параметров



Если система должна строить модель ЧСС, ей нужны параметры, которых в текстовой схеме может не быть. В схеме часто целесообразно задавать только главное: длительности, повторы, структуру, прогрессию или вариативное правило. Интенсивности могут намеренно опускаться, потому что их удобнее подбирать позже - по совпадению модельного и реального пульса.

Например, схема:

```
10/8*10#
```

задаёт рабочее время, отдых и количество повторов, но не говорит, насколько тяжёлой должна быть работа и насколько лёгким должен быть отдых. Обычный таймер такую схему мог бы просто прощёлкать. Но для построения модели ЧСС этого недостаточно.

Поэтому граф в Zaruba Timer - это не просто визуализация и не просто экран настроек. Это визуальная среда задания недостающих параметров модели. На графе пользователь может задать интенсивность работы, интенсивность отдыха и другие параметры простого атома, которые нужны для построения пульсового прогноза.

Из этого естественно возникает вторая роль графа: он становится полноценным альтернативным редактором простых схем. Если текстовая схема пуста, параметры графа сами образуют завершённую простую схему:

```
40/20@8/4*6#
```

По ней можно построить модель ЧСС, запустить таймер и выполнить тренировку без ручного написания схемы.

Если же текстовая схема задана, граф становится источником недостающих значений и одновременно презентером итогового слияния: он показывает уже не просто свои исходные настройки, а окончательную модель, по которой будет строиться ЧСС и работать таймер.

3. Два уровня модели: пользователь и тренировка

Иллюстрация 3. Два уровня модели

Персональная модель

пульсовые зоны · HR rest/max · tau · восстановление · дрейф



Тренировка

схема · граф · интервалы · интенсивности · повторы · прогрессии

модель ЧСС конкретной тренировки для конкретного пользователя

Zaruba Timer соединяет два уровня данных.

Первый уровень - персональный. Он описывает пользователя и его физиологическую модель: пульсовой диапазон, зоны ЧСС, параметры реакции и восстановления, tau, дрейф и другие настройки.

Второй уровень - тренировочный. Он описывает конкретную сессию: время работы, время отдыха, интенсивности, повторы, структуру, прогрессии, темп, вариативные правила и другие параметры нагрузки.

Обычный таймер работает почти исключительно со вторым уровнем, причём только с временем. Zaruba Timer связывает оба уровня и поэтому может не просто отсчитывать интервалы, а строить прогноз, контролировать соответствие модели и помогать уточнять тренировку после выполнения.

Эта связка особенно важна потому, что одна и та же внешняя тренировка может иметь разную физиологическую цену для разных людей или для одного и того же человека в разные периоды подготовки. Для одного пользователя заданная схема может быть лёгкой, для другого - предельной. Поэтому модель должна учитывать не только форму нагрузки, но и того, кто её выполняет.

4. Схема: язык описания сложной тренировки

Иллюстрация 4. Схема как язык

Граф: простой атом

40/20@8/4*6#



Схема: сложная форма

[dw=2 dr=2 N=10]

(dw/dr@r1 -> ... > auto@rN
-> ... -> auto@r1)*1`#



ползунками такую структуру не выразить

Если граф удобен для простой первичной прикидки, то схема нужна там, где тренировочная структура выходит за пределы одного атома. Слайдерами можно задать только одну простую форму. Но реальная тренировка может включать несколько разных атомов, прогрессии, лесенки, волны, случайно-турбулентные сценарии, повторяемые random-сессии с seed, бюджетные блоки, адаптивный темп и другие конструкции.

Именно здесь вступает в силу язык Схема. Он позволяет описывать не только длительности и паузы, но и форму нагрузки как таковую: как меняется темп, как нарастает или убывает интенсивность, как перестраивается ритм, как задаётся вариативность, как воспроизводится один и тот же сценарий для повторного сравнения.

В этом смысле схема - не просто краткая запись интервалов, а формальный язык управляемой вариативности тренировочной нагрузки.

Простая схема может выглядеть так:

```
10/8*10#
```

Она задаёт длительности и количество повторений, но не задаёт интенсивности.

Более сложная вариативная схема может выглядеть так:

```
[dw=2 dr=2 N=10]
(dw/dr@r1->[to:auto/auto@rN, delta: +dw/+dr@r+1]
->[to:auto/auto@r1, delta: -dw/-dr@r-1])*1`#title=подтягивания лесенка
```

Такая схема описывает автоматическую лесенку: от 1 повторения вверх до N и затем обратно вниз, с пропорциональным временем работы и отдыха. Подобную структуру невозможно выразить обычным набором ползунков.

Именно поэтому граф и схема не заменяют друг друга. Граф удобен как визуальный редактор простого атома и источник параметров модели. Схема нужна как язык сложной структуры.

5. Схема как общий язык и формат обмена

Иллюстрация 5. Схема как общий язык и формат обмена

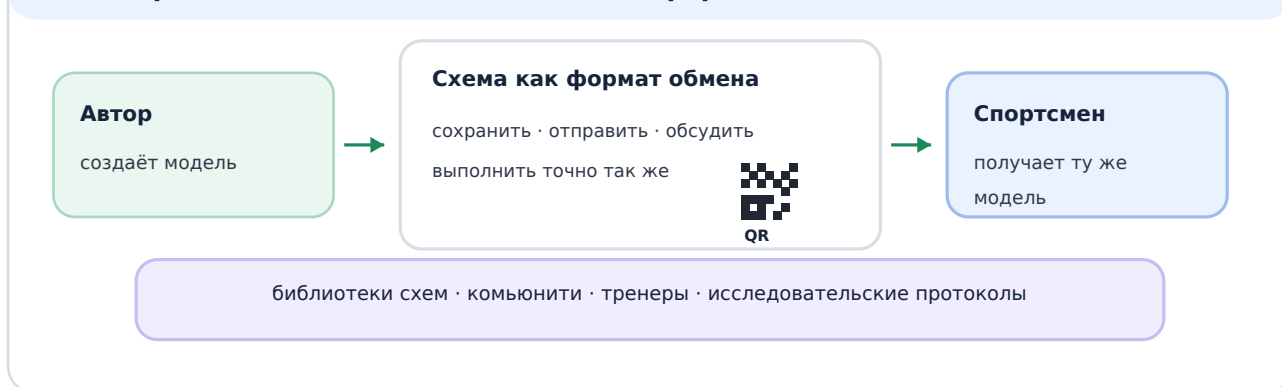


Схема нужна не только для того, чтобы описывать сложные тренировки внутри одного приложения. Она выполняет ещё одну важную роль: становится единым языком общения между спортсменами, тренерами, исследователями и авторами тренировочных моделей.

Обычное словесное описание тренировки почти всегда допускает разночтения. Один человек пишет: «сделай лесенку подтягиваний с равным отдыхом», другой понимает это немного иначе, третий меняет паузы, четвёртый по-своему считает переходы. В результате вроде бы обсуждается одна и та же тренировка, но фактически выполняются разные модели.

Схема решает эту проблему. Она задаёт тренировку формально, компактно и исполнимо. Её можно передать другому человеку и быть уверенным, что таймер развернёт и выполнит именно ту структуру, которая была задумана автором.

Например, вариативная лесенка подтягиваний может быть передана одной строкой:

```
[dw=2 dr=2 N=10]
(dw/dr1->[to:auto/auto@rN, delta: +dw/+dr@r+1]
->[to:auto/auto@r1, delta: -dw/-dr@r-1])*1`#title=подтягивания лесенка
```

Это не просто описание «примерно такой тренировки». Это переносимая модель, которую можно сохранить, отправить, опубликовать, обсудить, повторить и проверить.

Из этого естественно возникает библиотечный и комьюнити-сценарий. Пользователи могут создавать собственные схемы, сохранять их в избранные, делиться ими с другими, собирать тематические библиотеки и пополнять общедоступные коллекции: для подтягиваний, гирь, бега, гребли, паруса, единоборств, реабилитации, тестов и исследовательских протоколов.

Таймер поддерживает этот слой не только концептуально, но и практически: схема может передаваться через шаринг, а также кодироваться в QR-код и считываться обратно. Это превращает схему в удобный формат распространения тренировочных моделей: на сайте, в чате, в методичке, на соревновании, в клубе или в исследовательской группе.

Поэтому Схема - это не только язык управления таймером. Это общий язык точного обмена тренировочными моделями.

6. Трафарет: как схема и граф сливаются в итоговую модель

Иллюстрация 6. Трафаретное наложение

Нижний слой: граф

40/20@8/4*6#

Верхний слой: схема-трафарет 10/8*10#

отверстия: @ ? / ?



Итоговая модель

10/8@8/4*10#

Ключевая идея системы состоит в том, что граф и схема не конкурируют. Они образуют два слоя одной модели.

Нижний слой - это параметры графа. Они всегда могут породить простую полную схему одного атома:

40/20@8/4*6#

Верхний слой - это текстовая схема пользователя. Она может быть полной, а может быть частичной. Если схема полная, она перекрывает граф почти полностью. Если схема частичная, она работает как трафарет: заданные в ней параметры закрывают соответствующие значения графа, а пропущенные места остаются отверстиями, через которые берутся значения нижнего слоя.

Например, граф задаёт:

40/20@8/4*6#

А в редакторе написано только:

50

Это означает: пользователь задал только время работы. Остальное берётся из графа. Итоговая схема становится:

50/20@8/4*6#

Другой пример:

10/8*10#

Здесь схема задаёт время работы, время отдыха и количество повторений, но не задаёт интенсивности. Поэтому интенсивности работы и отдыха берутся из графа:

10/8@8/4*10#

Такой механизм нужен не ради сложности, а ради практической свободы. Пользователь может писать схемы настолько подробно, насколько нужно в данный момент.

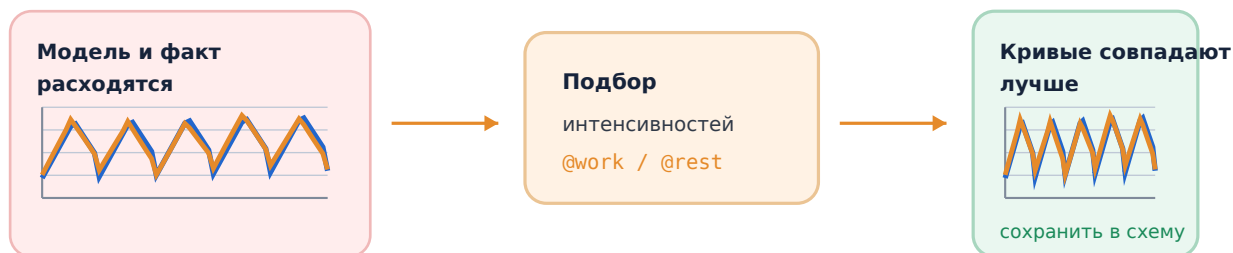
Недостающие параметры не ломают модель, потому что они пополняются из графа.

Поэтому граф выполняет сразу три роли:

- редактор простых схем;
- источник недостающих параметров для неполных схем;
- визуальный презентер итоговой объединённой модели, по которой строится ЧСС и работает таймер.

7. Почему в схеме часто опускаются интенсивности

Иллюстрация 7. Подгон модели по реальному пульсу



В тренировочной практике длительности, структура и количество повторений часто являются более стабильной частью плана, чем интенсивности. Интенсивности могут подбираться, уточняться или даже намеренно оставляться неопределёнными на этапе первоначальной отладки схемы.

Это имеет важный методический смысл. Если временно не фиксировать интенсивности в самой схеме, можно использовать граф как инструмент калибровки. После выполнения тренировки таймер сравнивает модельную кривую ЧСС с реальной кривой, полученной с пульсометра. Если совпадение недостаточно хорошее, пользователь может менять интенсивность работы и интенсивность отдыха на графе, добиваясь лучшего согласования модели с фактической физиологической реакцией.

Когда подходящие значения найдены, их можно уже закрепить в самой схеме и сохранить её как персонально откалиброванный пресет. Таким образом, схема сначала задаёт структуру, затем граф помогает подобрать численные параметры, а после этого обе стороны соединяются в более точную и устойчивую модель.

Это особенно важно для индивидуальной настройки. Одна и та же схема может требовать разных интенсивностей в зависимости от спортсмена, периода подготовки, упражнения, веса снаряда, усталости и текущего состояния. Поэтому возможность сначала оставить интенсивности открытыми, а затем подобрать их по реальному пульсу - не недостаток языка, а сильная сторона всей системы.

8. Вариативные схемы как воспроизводимый эксперимент

Иллюстрация 8. Схема как воспроизводимый эксперимент



Сильная сторона языка Схема в том, что он позволяет задавать не только работу и отдых, но и саму форму нагрузки во времени: плавные темпо-рампы, волны, прогрессии, random-сценарии, seed для повторяемости, работу относительно опорного пульса, а также сравнение повторных проходов.

Это делает схему не просто инструкцией для таймера, а воспроизводимым экспериментом над формой нагрузки и реакцией спортсмена.

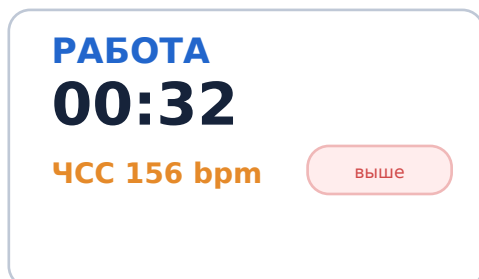
Практический смысл здесь очень велик. Если спортсмен многократно проходит одну и ту же схему или одну и ту же вариативную random-схему с фиксированным seed, становится возможным сравнивать не только результат, но и физиологическую цену результата:

- стал ли пульс ниже на той же нагрузке;
- уменьшился ли дрейф ЧСС;
- быстрее ли восстановление;
- не слишком ли рано наступает разнос пульса;
- легче ли переносится финишный участок;
- можно ли поднять темп без роста пульсовой цены.

Таким образом, тренировка становится не только выполнением предписанной нагрузки, но и средством накопления измеримого опыта. Это особенно важно и для сильных спортсменов, и для обычных пользователей: одним это даёт язык тактики и точного анализа, другим - понятный способ безопасно прогрессировать и видеть реальные изменения, а не только субъективные ощущения.

9. Таймер как инструмент контроля в реальном времени

Иллюстрация 9. Контроль в реальном времени



После того как итоговая схема сформирована, она используется не только для предварительного построения графика, но и для реального исполнения тренировки. Таймер ведёт пользователя по рабочим и восстановительным фазам, получает данные ЧСС с пульсометра и сопоставляет фактический пульс с модельной кривой.

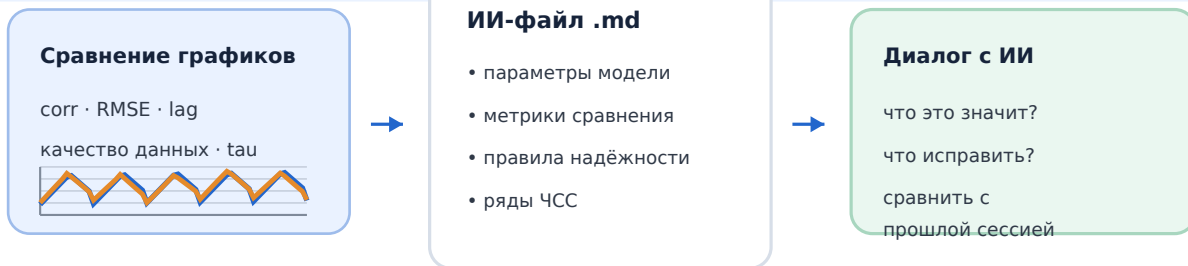
Это очень важный момент. Модель в Zaruba Timer - не декоративная картинка до начала сессии, а рабочий ориентир во время самой тренировки. Если фактический пульс заметно уходит выше или ниже ожидаемой траектории, таймер может предупредить пользователя в реальном времени.

Так можно вовремя заметить, что нагрузка оказалась слишком тяжёлой, слишком лёгкой, что восстановление идёт не так, как ожидалось, или что фактическая реакция спортсмена отличается от модели.

Именно здесь сочетаются наглядность графа, формальность схемы и практическая ценность пульсового мониторинга. Таймер становится не просто устройством отсчёта времени, а средой воспроизводимого исполнения и физиологической валидации тренировочного плана.

10. ИИ-файл: переход от метрик к пониманию

Иллюстрация 10. ИИ-файл



После сравнения модельного и реального графиков таймер может выдавать комплексный анализ: корреляцию, среднеквадратическую ошибку, лаг, перекрытие, оценку качества данных, правдоподобность параметров tau, особенности восстановления и другие показатели.

Однако такой анализ не всегда легко понять пользователю напрямую, особенно если речь идёт о нескольких взаимосвязанных метриках одновременно.

Поэтому важной особенностью системы является создание специального ИИ-файла - структурированного текстового файла, который содержит:

- параметры модели;
- параметры реальной тренировки;
- ключевые метрики сравнения;
- правила надёжности выводов;
- при необходимости - сами ряды ЧСС.

Такой файл можно передать ИИ и получить не только сухую интерпретацию чисел, но и полноценное человеческое объяснение: что произошло, насколько выводам можно доверять, что можно исправить в следующей тренировке, связано ли расхождение с моделью, физиологией или качеством данных, а также как сравнить текущую сессию с прошлой.

На основе этого же файла можно инициировать не один ответ, а целую серию разговоров с последовательным вводом результатов новых сравнений. Тогда анализ перестаёт быть разовым отчётом и становится продолжением тренировочного дневника.

Это делает ИИ не заменой аналитики, а её естественным продолжением. Сначала таймер выполняет строгий формальный расчёт, а затем ИИ помогает перевести полученные данные в осмысленное объяснение, практические рекомендации и длительное накопление понимания собственной тренировочной реакции.

ИИ-файл можно рассматривать как мост между машинным анализом и человеческой интерпретацией.

11. Итог: Zaruba Timer как единый цикл работы с тренировкой

Иллюстрация 11. Полный цикл



каждая тренировка уточняет модель и следующий проход

Смысл Zaruba Timer раскрывается не в одной отдельной функции, а в их соединении.

- Граф даёт быстрый старт, наглядность и первичную модель.
- Схема даёт язык описания тренировки любой сложности.
- Схема также является переносимым форматом обмена: её можно сохранить, передать, опубликовать, считать по QR-коду и выполнить воспроизводимо.
- Трафаретный механизм соединяет структуру схемы с параметрами графа.
- Персональная модель связывает тренировку с конкретным пользователем.
- Сравнение модельного и реального пульса помогает калибровать нагрузку.
- Таймер сопровождает выполнение в реальном времени и следит за отклонениями.
- ИИ-файл переводит аналитические метрики в понятное объяснение и открывает возможность дальнейшего диалога.

Поэтому Zaruba Timer - это не просто интервальный таймер, не просто визуализатор и не просто язык описания сессий. Это единый инструмент планирования, прогнозирования, прохождения, анализа и персонализации тренировки.

Его особая ценность в том, что он позволяет работать не только с внешней формой тренировки, но и с её внутренней физиологической логикой - от проектирования до интерпретации результата.

Краткая формула документа

Обычный таймер:

работа / отдых → звуковые сигналы

Zaruba Timer:

персональная модель + схема + граф → прогноз ЧСС → выполнение → реальный пульс → сравнение
→ ИИ-анализ → уточнённая схема

Именно поэтому связка графа и схемы не является усложнением ради усложнения. Граф нужен системе как источник динамических параметров и редактор простых моделей. Схема нужна как язык сложных и переносимых тренировочных моделей. А их совместная работа позволяет строить, выполнять, проверять и уточнять тренировку как воспроизводимый процесс.