

ЭКОНОМИЧЕСКИЙ
СУБЪЕКТ

ЦИФРОВАЯ
ЭКОНОМИКА

ТОМ III

Сергей Дегтярев

Сергей Дегтярев
Экономический субъект. Том
III. Цифровая экономика

<https://litres.ru/74069821>

SelfPub; 2026

Аннотация

Данная работа представляет собой фундаментальный гуманитарный манифест, переосмысляющий экономическую историю и проектирующий будущее через призму смены культурных эпох.

Ключевым открытием автора является обоснование неизбежного перехода к пятой экономике — экономике универсальных полуфабрикатов и кодэтики.

Особое внимание уделяется феномену отладки, который трактуется как ведущая экономическая деятельность в цифровой экономике. Автор вводит новое экономическое отношение — кодэтику, где единство этики и эстетики перестает быть философской абстракцией и становится материальным условием процветания.

В заключительной части работа выходит за горизонт экономики, описывая шестую экономику неисчерпаемых запасов, где человек окончательно перестает быть производителем и превращается в творца.

Текст представляет собой синтез экономической теории, философии и культурологии, предлагая России стратегию опережающего развития через образование и суверенную цифровую культуру.

Содержание

Введение	5
Глава 8. Цифровая экономика	10
Отладка	10
Алгоритмы	15
Универсальный полуфабрикат	23
Кодэтика	30
Философия кодэтики	48
Цифровая экономика – исчерпывающий ликбез	59
Конец ознакомительного фрагмента.	86

Сергей Дегтярев

Экономический субъект. Том III.

Цифровая экономика

Введение

Во первом и втором томе мы совершили беспрецедентную операцию: оставив позади споры о «справедливой цене» и «классовой борьбе», разобрали экономику не как поле битвы классов, а как сложную, органическую смену культурных эпох. Мы установили, что каждая экономика держится не на «ресурсах» или «капитале», а на строго определенном *культурном условии, ведущей экономической деятельности и ведущем экономическом отношении*.

Первая экономика (соборности) держалась на совместном добывании и потреблении, пока природа была щедра. Ее упадок породил вторую экономику (наряда) — с ее систематической разведкой скоплений и строгим учетом исчерпания запасов из скопления. Когда иссякли и легкодоступные скопления, человечество перешло к третьей экономике (обряда поощрения), где главным стало не найти, а *заставить*

природу самовозрастать через ритуал и похвалу.

И, наконец, мы подробнейшим образом исследовали **четвертую экономику** — экономику **присяги** и рутинного перерабатывающего труда. Мы доказали теорему: именно нерушимое обещание («клянусь поставить», «клянусь принять», что делает время создания запаса измеримым) позволяет экономике процветать.

Мы описали четыре ситуации процветания четвертой экономики — от полной гарантии всего до двойной негарантированности, когда на сцену выходит **отладка** как возвращение «лада».

Мы показали, что упадок четвертой экономики начинается ровно в тот момент, когда присягу заменяют договором с «форс-мажором», а «честное слово» обесценивается. Казнь нарушителя присяги — не жестокость, а экономическая необходимость.

Главным же открытие второго тома — радикальное перепределение природы денег. Мы разоблачили главную либерально-марксистскую догму: деньги — продукт эволюции товарного обмена. Мы показали, что систематического обмена товаров в истории никогда не было, а «Капитал» Маркса — гениальная поэма, но не наука. Деньги, утверждаем мы, это **суррогат запасов**, навязанный сообществу «хозяевами денег» (выходцами из упадочной третьей экономики) как *якобы* необходимое условие доступа к реальным благам. Деньги — это форма существования упадка, инстру-

мент «денежного порабощения», позволяющий получать гарантированную прибыль, не создавая запасов и не держа присяги. Капиталист, с этой точки зрения, — это не «эффективный менеджер», а рантье, получающий ренту за счет контроля над суррогатом.

Мы завершили второй том на драматической ноте: четвертая, нынешняя экономика (присяги) умирает. Ее ведущее отношение разложено договорным формализмом, а ее эффективность падает. Но, гуманитарный метод утверждает, что гибель ведущей экономики — это не конец, а условие для рождения следующей.

В этом томе мы обращаем взгляд в будущее. Если четвертая экономика была экономикой *присяги* и рутинного перерабатывающего труда, то следующая, **пятая экономика** — это экономика **универсальных полуфабрикатов и отладки**.

Читателю предстоит узнать, что отладка — это не только техническая регулировка станка, а возвращение древнего «лада» в мир, где ничего не гарантировано. Мы покажем, что цифровая экономика, по поводу которой сегодня столь много спекуляций, есть не что иное, как историческая форма реализации пятой экономики. Мы введем новое, шокирующее для гуманитариев понятие — **«кодэтика»**. Это не просто «этика программиста». Кодэтика — это новое экономическое отношение, приходящее на смену присяге. Это отношение согласования и отладки алгоритмов, взятое в их

этическом и эстетическом единстве.

Мы докажем главную теорему пятой экономики: для процветания экономики запасов в виде универсальных полуфабрикатов необходимо и достаточно кодэтики. Это означает, что в цифровом мире *красивый* (понятный, регулярный) код становится не просто эстетическим предпочтением, а *этическим императивом*, условием процветания. Мы покажем, что в пятой экономике рутинный труд окончательно переходит к роботам, а человеку остается отладка.

И наконец, мы рискнем заглянуть за горизонт. Если пятая экономика делает затраты на превращение форм исчезающе малыми, то **шестая экономика** — это экономика **рукотворной неисчерпаемости**, где запасы не убывают от использования. Это царство чистого порождения форм, где экономическая задача решена окончательно, а человек из «производителя» превращается, наконец, в **творца**, играющего формами ради самой игры.

Центральный практический вывод третьего тома — и он, возможно, самый важный для России — заключается в том, что поголовное обучение программированию на **русском языке** есть условие перехода в пятую экономику и достижения подлинной цифровой неуязвимости. Это кратчайший путь к обучению понятийному мышлению, который позволит нам не догонять Запад в гонке за устаревшими технологиями, а перешагнуть в новую экономическую реальность, где «быть как все» становится ненормальным, а «быть не как

все» — нормой процветания.

Мы оставляем мир договоров и присяг. Мы вступаем в мир алгоритмов, отладки и кодэтики. Мы вступаем в цифровую эпоху.

Глава 8. Цифровая экономика

Остановим же взор на четвёртой ситуации — на той, что царит в лоне процветающей четвёртой экономики! Ситуации, где ничто не даруется свыше как гарантия. Ни само создание. Ни неминуемое исчерпание.

Ибо создание — о, сколь технологически ненадёжно оно! Брак, словно роковое клеймо. Сбои, подобные ударам судьбы. Нестабильность сырья, что крадётся из тени. А исчерпание? Оно и вовсе непредсказуемо: то причуды спроса, то капризы погоды, то бездна внешних обстоятельств.

Двойная негарантированность — казалось бы, вот оно, проклятие, тяжкий крест, конец всем надеждам. Ан нет! Именно из этого проклятия, как из горнила, рождается новый вид деятельности. Деятельность, которой нет места в иных, предсказуемых ситуациях.

Отладка

Отладка в её изначальном смысле: приведение механизма в состояние исправное, работоспособное, истинное. Будь то часов бег, меткость оружия, ровный гуд двигателя, острота инструмента или стройность самого процесса.

Дабы механизм мог вершить рутинные действия — раз за разом, одно как другое, без сбоев, без малейших отклоне-

ний. Отладка — это вам не пустая суета. Это регулировка. Тончайшая настройка. Устранение физических неполадок. Всё ради одного: бесперебойного, как дыхание функционирования.

Вот что гласит нам непререкаемый авторитет — словарь Даля, сокровищница живой великорусской речи. Слово наше восходит к величавому глаголу «отладить» — и что же он означал во времена былые? Приспособить! Привести в полный порядок! Довести всякое дело до победного, сияющего конца!

Вдумайтесь: слово «отладка» и породивший его глагол «отладить» существовали задолго до того, как заскрипели первые сложные механизмы. Задолго до паровых машин, что потрясли мир своим могуществом. Задолго до часов с боем, отмеряющих вечность. Корни эти уходят в глубь древнеславянского языка — в самый быт наш, в пение, в живые человеческие отношения.

Ибо исконный корень — ЛАД.

Да-да, тот самый! Мир и согласие — жить в ладу с собою и миром. Ладить с соседями, не ведая вражды. Порядок и устройство — приводить дела в лад, дабы всё текло стройно и складно. А ещё — музыку и пение! То есть стройное, божественное звучание, слаженность голосов, где каждый звук находит своё истинное место в общем хоре.

О, сколь глубоки корни сего слова, сколь нежны и удивительны его истоки!

В бессмертном «Слове о полку Игореве» — свидетеле доблести XII века — словом «лада» ласково, почти с молитвой на устах, называли любимого супруга или супругу. Ибо Лад — это не просто звук. Это связь. Это созвучие душ. Это соразмерность всех вещей.

Значение слова менялось постепенно, подобно тому, как река меняет русло, но не суть свою. От человека — к железу. От сердца — к стали.

Глагол «отладить» — или, как ещё говорили, «уладить» — означал тогда деяние поистине высокое: помирить людей. Закончить спор миром, а не мечом. Прийти к согласию, а не к покорению. Не победить! Не подавить! От-ладить. То есть — возвратить утраченный лад.

Вот плотник, потом и кровью строит избу. Вот бондарь, чьи руки помнят древнее ремесло, делает бочку. Детали нужно «приладить» друг к другу — подогнать так, чтобы не щело, не текло, не стеноло под ветром. Конечный результат, когда всё сидело плотно, намертво, по-живому, называли «отлаженным» делом. Не хитрым. Не сложным. Отлаженным. То есть — пребывающим в ладу. С материалом. С формой. С высшим назначением вещи.

И только с приходом развитой техники — с грохотом станков, с пытением паровых машин, с благородным рыком пианино — слова «отладить» и «отладка» закрепились за настройкой. Точной. Физической. Осязаемой. Для пианино. Для станков. Для часов, отмеряющих теперь уже иную

эпоху.

Таким образом, изначально «отладить» означало одно: вернуть гармонию. Порядок. Мир. В отношения. В дела. В пение. И лишь спустя века, словно переселяясь из храма в кузницу, это слово перенесли на машины.

Но смысл — о чудо! — остался. Отладка механизма — это возвращение механизма в состояние лада. С собою. С внутренним ритмом. С его единственной правдой — функцией. Чтобы он работал не как попало, не как вздумается железной его душе. А как надо. Чтобы не скрипел. Не заедал. Не сбивался с шага.

И вот что я скажу вам, слушающие. Та самая экономическая четвертая ситуация — ситуация процветающей негарантированности — требует того же самого. Вернуть лад туда, где нет гарантий. Где создание ненадёжно, словно шагание по тонкому льду. Где исчерпание непредсказуемо, как погода в море.

Кто умеет отлаживать — тот умеет **возвращать** лад в негарантированный мир. Тот, клянусь, настоящий мастер четвертой экономики. Не только над железом — над отношениями. Над делами. Над ритмом. Чтобы экономика процветала. Даже когда ничего не гарантировано. Потому что есть Лад. Есть древнее Слово. Есть умение вернуть порядок из хаоса — старинное, славянское, человеческое, забытое теми, кто верит только в деньги. И помнящее — теми, кто работает и держит слово, как стяг на ветру.

О, теперь мы подходим к самому сердцу той деятельности, что отличает четвертую ситуацию четвёртой экономики от всех прочих, — к деятельности, достойной называться искусством.

Речь идёт об отладке систематической. Не о той случайной, что хватает ускользающую удачу за хвост от случая к случаю, — нет! Об отладке, встроенной в самую плоть экономической деятельности. И она, эта отладка, породила нечто доселе невиданное: деятельность специфическую, деятельность, которой не было и не могло быть в иных, благополучных ситуациях.

В чём же суть сей деятельности, спросите вы? Вот она, как откровение: фиксация методов. Только методов — и ни в коем случае не результатов! Ибо результаты, о горе, каждый раз иные, будто их подменяют под покровом тумана. Потому что — повторим, как единственную правду — ничего не гарантировано. Не гарантировано создание. Не гарантировано исчерпание. Сегодня — одно. Завтра — другое. Послезавтра — третье, такое, что и не снилось.

Пытаться зафиксировать результаты? Бессмысленно! Их нет. Они ускользают сквозь пальцы, как песок. Они случайны до глубины своей. Они не повторяются — никогда, ни за что, ни при каких обстоятельствах.

Но вот что чудесно: результаты разные — а методы, методы остаются! Те самые, священные методы отладки, которые гарантированно, словно по велению высшей силы, приводят

к желанной работоспособности. Выработанные потом и кровью. Проверенные на прочность временем. Зафиксированные на скрижалях опыта.

Вот они, эти методы, предстают перед нами как ступени восхождения к порядку.

Шаг первый. Диагностика. Великое вопрошание: что сломалось, где притаился тот самый разлад, что рушит всё?

Шаг второй. Настройка. Точное, выверенное движение: в каком именно месте надлежит регулировать, дабы вернуть лад?

Шаг третий. Проверка. Суровый суд действием: заработало? Нет? Тогда без гнева и уныния — вернуться к шагу первому и начать сначала, ибо таков путь.

Шаг четвёртый. Фиксация. Венец деяния: запомнить каждое движение, записать каждый вывод, передать следующему, чтобы не начинать всякий раз из тьмы.

Эти методы суть нечто большее, чем просто перечень. Они суть алгоритмы. А алгоритм, как известно, есть музыка, записанная нотами, есть путь, проложенный через чащу, есть победа над хаосом, одержанная заранее.

Алгоритмы

Слово сие кажется нам ныне сугубо компьютерным, порождением эпохи кремния и битов. Но как обманчива внешность! На самом деле — оно древнее, как сама мысль, как

первый осмысленный жест человека. Алгоритм — это ведь что? Это последовательность действий, гарантированно ведущая к результату. Да, именно так: если результат вообще может быть получен в этом несовершенном мире.

И вот парадокс, достойный пера самого великого диалектика. В четвёртой ситуации процветающей четвёртой экономики результат, увы, не гарантирован. Но! Если он, этот ускользающий призрак, вообще может быть получен — то алгоритм отладки, клянусь, его добудет. Вырвет из лап хаоса. Приведёт в этот мир, как повитуха приводит новорождённого.

Фиксация алгоритмов — это, заметьте, отдельная деятельность, почти священнодействие. Кто ею занимается, кто посвятил себя сей стезе — тот не просто работает, тот творит скрижали для грядущих поколений. Он создаёт инструкции. Стандарты, что держат небо над землёй. Регламенты, в которых пульсирует порядок. Руководства, подобные картам звёздного неба. Памятки, что спасают в минуту отчаяния.

В процветании четвёртой ситуации без этого — нельзя. Абсолютно нельзя. Потому что каждый раз — о, как это утомительно! — новый сбой, подлый и неожиданный. Каждый раз — новый механизм, со своим норовом и характером. Каждый раз — новая настройка, требующая тончайшего проникновения в суть. И только зафиксированный алгоритм, этот якорь в море непредсказуемости, позволяет не изобретать велосипед в тысячный раз. А применять прове-

ренное, выстраданное, выверенное. Экономить время — самый драгоценный из даров. Терпение — добродетель мастеров. Запасы — последний оплот перед бездной.

Вот он, великий парадокс. В экономической ситуации, где результат не гарантирован, метод становится важнее самого результата. Ибо результат — всего лишь капризная случайность, дева переменчивая и своенравная. А метод — закономерность, суровая и прекрасная, как закон всемирного тяготения. Выучил метод — и рано или поздно получишь результат, будьте уверены. Не выучил — увы! — будешь тыкаться вслепую, как слепой котёнок в тёмной комнате. Тратить время, которого и так в обрез. Нервы, которые не купишь ни за какие деньги. Материал, который не воскресить.

И знаете что? Отсюда — всего один шаг. Один-единственный шаг. До пятой экономики.

Сегодня, когда гордое слово «отладка» срывается с губ человеческих, слух улавливает не звон металла и не шершавое дыхание ремесла. Увы и ах! Люди слышат цифру. Не станок, чьи объятия ковали форму. Не древнее ремесло, уходящее корнями в седую старь. Не бондарную бочку, где каждый обруб спет с деревом в согласии. Нет — они слышат программу. Код, манящий и пугающий своей незримостью. Экран, что светится в сумраке, как око чудовища. Компилятор, безжалостный судия написанных строк.

И знаете что? Это — не случайно. О, нет, это не натяжка ума и не каприз моды, пришедший и ушедший с ветром

перемен. Это — глубокая, неумолимая правда. Потому что в цифровой разработке, в этом новом мире, созданном из единиц и нулей, отладка стала деятельностью ведущей, главенствующей, царицей на троне.

Посмотрим же правде в глаза — смело, без трепета, как подобает мудрецу. Программист, о величии которого скоро будут слагать легенды, на самом деле... не пишет код. Вернее, пишет, куда он денется. Но написание кода, эти священные минуты творения, занимают лишь меньшую, самую малую часть его драгоценного времени. А основное время — о ирония судьбы! — поглощает отладка. Написал строчку — проверь, не таит ли она измены. Написал функцию — прогрузи её в суровый интерпретатор. Написал модуль — протестируй до хруста, до скрежета, до седьмого пота. Нашёл баг — исправь, принеся жертву логике. Исправил — и о, ужас! — появился новый, словно Гидра, отращающая головы. Новый исправил — старый вернулся, незванный гость из забытого прошлого.

Цифровая разработка это и есть экономика в четвертой ситуации в чистейшем, неразбавленном виде. Здесь, среди сияния экранов и гула серверов, ничего не гарантировано. Абсолютно ничего, но вот парадокс, отлаженный код гарантирует, что алгоритм будет исполнен, и результат будет получен.

Представьте себе водителя. Старого, как сама дорога. Его руки помнят баранку ещё тех лет, когда навигаторы были

лишь мечтой фантастов, а звёзды служили компасом. Ему не нужен навигатор. Он знает дорогу — каждую выбоину, каждый тайный объезд, каждую лазейку, не нанесённую на карты. Но логистическая фирма, суровая и бездушная, требует: «Работай с программой!» И вот программа прокладывает маршрут — гордый, выверенный, математически безупречный. Водитель следует. И — о неизбежность! — попадает в пробку, змеющуюся на многие вёрсты. Или на закрытый мост, о котором программа не ведаёт. Или туда, где разворот запрещён знаком, а длинная фура, проклятая, не пролезает между бетонными блоками.

Что же делает истинный мастер? Водитель на ходу, под рокот мотора, вносит живые правки. объезжает препятствие, словно река огибает скалу. Запоминает проклятое место, чтобы больше никогда не попасть в сию ловушку. После рейса, усталый, но не сломленный, пишет в техподдержку: «Здесь, люди, нельзя ехать. Исправьте, ради бога». Программисты получают сей отчёт, как послание из другого мира. Отлаживают алгоритм, вносят изменения в самое нутро. Навигатор становится точнее — на йоту, на линию. Водитель получает новую версию. И снова, с неизменным терпением, отлаживает.

И кто же здесь, спросите вы, производит? Водитель? Программист? О, лукавый вопрос! Никто не производит запаса в строгом смысле четвёртой экономики. Все, абсолютно все заняты отладкой. Водитель отлаживает маршрут. Програм-

мист отлаживает код. Результат, как всегда, не гарантирован — сегодня доехал, завтра встал. Зато метод — о, какое счастье! — зафиксирован. Запомнен. Передан дальше.

А вот другой герой. Бухгалтер. Опытный, как сама отчётность. Он считал еще на деревянных счётах, где костяшки ходили под пальцами с успокаивающим стуком. Потом, на заре цифровой эры, — в Excel, этой великой песочнице вычислений. Но фирма, неумолимая в своём стремлении к прогрессу, перешла на 1С. Программа пишет проводки. Автоматически. С гордым видом всезнайки. Но налоговая — это чудовище, меняет правила каждый квартал — новые формы, словно лабиринт для посвящённых. Новые коды, где цифра значит больше, чем имя. Новые ставки, что рубят по живым деньгам.

Бухгалтер не пишет код — не его это ремесло. Он настраивает. Меняет параметры, как хирург меняет инструмент. Добавляет справочники, расширяя границы возможного. Проверяет отчёт — и сердце замирает. Отчёт не сошёлся! Ищет ошибку — иголку в стоге сена, где каждая цифра лжёт. Находит — о, мгновение торжества! Правит. Снова проверяет. Это, клянусь, чистейшая отладка — отладка процесса, отладка самого метода учёта.

Результат — сданная декларация, этот пропуск в мир законопослушного существования. Но каждый раз путь к ней — разный, неповторимый, как отпечаток пальца. Потому что исходные данные разные. Правила разные. Сбои — беско-

нечно разные. Единственное постоянное, единственное, что держит этот мир от распада, — метод отладки. Зафиксированный в инструкциях, в глубине памяти, в святой привычке проверять дважды — а лучше трижды.

И, наконец, третий герой нашей оды. Токарь. Старый. Работал на универсальном токарном станке, где каждый резон был продолжением руки, где металл пел и стонал под пальцами, где мастер чувствовал его, как мать чувствует дитя. Теперь — станок с числовым программным управлением, гордый, безучастный, железный истукан. Программист написал программу, выверил каждую команду. Загрузил в чрево машины. Нажал пуск. И что же? Первая деталь — брак, позорный и неизбежный. Вторая — лучше, уже дышит надеждой. Третья — годная, наконец-то! Но десятая — о, вероломство! — опять брак, потому что резец, несчастный, затупился, а программа, глупая, об этом не знает.

Токарь отлаживает. Меняет подачу. Корректирует скорость, где каждый оборот решает судьбу детали. Перетачивает инструмент — и чует, как сталь обретает новую остроту. Программист вносит правки в код, исправляя ошибки, которых не мог предвидеть никто. Это не работа в смысле рутинной переработки четвёртой экономики. Это чистая отладка. Метод — одинаков от начала до конца, как палиндром. Результат — каждый раз другой, как отражение в мерцающей воде.

Зато, когда метод найден, выстрадан, выношен — станок

выдает идеальную партию. Ровную, прекрасную, желанную. До следующего сбоя. Или до смены материала. Или до того, как мир в очередной раз изменит правила игры. Но мастер не боится. Ибо у него есть отладка. Ибо он помнит: лад возвращается. Всегда.

Во всех сих примерах, столь разных и столь единых, есть нечто общее — общее, как воздух, как сама необходимость. Люди, эти титаны повседневности, занимаются вовсе не производством в его примитивном, хвастливом смысле. Не переработкой запасов — нет, оставьте это убогое занятие тем, кто не дорос до пятой экономики. Они занимается деятельностью куда более высокой, почти мистической: приведением в порядок. Отладкой. Ибо среда их обитания — четвёртая ситуация, где ничего, слышите ли, ничего не гарантировано. Создание ненадёжно, словно клятва, данная в хмелю. Исчерпание непредсказуемо, как погода в море, как сердце женщины, как завтрашний курс валют.

Цифровая разработка, эта дерзкая юница, просто довела сию логику до предела — до предела, за которым начинается истина. Там нет даже механизма, привычного, осязаемого, железного. Только код — призрачная плоть нашего времени.

И вот что я скажу вам, слушающие, и пусть эти слова падут на благодатную почву. Ведущая деятельность цифровой экономики — о, обманчивое название! — вовсе не программирование. Программирование есть лишь написание текста, грамматика без музыки. Ведущая деятельность — отладка.

Только она, суровая и милостивая, превращает негарантированность в живое, дышащее, надёжное «работает». Только она фиксирует метод — делает его осязаемым, повторяемым, бессмертным. А метод сей становится алгоритмом. А алгоритм есть запас. Запас пятой экономики, того неведомого берега, куда мы гребём сквозь шторм.

Так отладка, начав свой путь с ремесленной регулировки — с того самого мгновения, когда рука человека впервые подстроила камень к топорищу, — превратилась в мировоззрение. Сначала был Лад — великое, всеобъемлющее согласие. Потом — настройка механизмов, грубых и прекрасных. Потом — фиксация методов, запись того, что нельзя упустить. Теперь — цифра, танец единиц и нулей, призрачный огонь экранов.

Универсальный полуфабрикат

Вдумаемся же в самое сердце отладки — в то, без чего она немыслима, как немыслима птица без крыла, а рыба без воды. Отладка — это вам не торжественное шествие победителя! Это деятельность, которая всегда, испокон веков и до скончания времён, идёт через пробы. Через мучительные, прекрасные, единственно возможные пробы.

Сделал действие А — смелое, уверенное, полное надежд. Получил результат R1. Но! Не тот, что нужно. Не тот, что желанен. Не тот, что ведёт к порядку.

Что делает истинный мастер? Он не идёт напролом, не умножает ошибку на ошибку. Он — откатывается к состоянию до А. Возвращается, словно время вспять, в ту благословенную точку, где ещё не было совершено ложное действие.

Затем делает действие Б — иное, выверенное, рождённое из пепла первой неудачи. Получает R2.

Сравнивает. Взвешивает на невидимых весах опыта. Выбирает лучшее. Или продолжает поиск, ибо путь к ладу редко бывает прям.

И запомните это, впишите в свои сердца, ибо здесь сокрыта великая тайна. Без отката отладка невозможна. Абсолютно невозможна. Потому что если ты, бедный смертный, не можешь вернуться в точку перед ошибкой — туда, где мир был ещё цел, — ты не можешь попробовать другой путь. Никакой. Ошибка накапливается, растёт, как снежный ком, несущийся с горы. Каждая следующая проба не исправляет, а множит хаос. Первая ошибка плюс вторая — это не две ошибки, как можно было бы наивно предположить. Это система ошибок. Это лабиринт, где каждая стена построена из предыдущего заблуждения.

Что же такое откат? Откат — это сброс. Обнуление. Чистый лист. Но тот же самый лист, а не новый. Не подмена, не бегство, не трусливое начало с нуля на другой заготовке. Нет. Это возвращение к тому же самому материалу, к той же самой судьбе — но очищенному от ложного шага.

Как же выглядит откат в суровом материальном мире —

мире станка, бочки, детали, упрямой заготовки? Вот его три священных шага.

Первый. Снять бракованную деталь — ту, что родилась не в лад, ту, что несёт на себе печать ошибки.

Второй. Вернуть заготовку в исходное состояние — то самое, первозданное, нетронутое, полное обещаний.

Третий. Начать заново — с тем же материалом, но с иным знанием.

И здесь кроется суровейшее из требований. Заготовка должна — слышите, должна! — сохранять свои свойства при возврате. Если после первой попытки, после первой неудачной пробы заготовка испорчена необратимо — отката нет. Точка. Конец надежды. Тебе остаётся только одно: выбросить. Взять новую. Но новая — как горько это осознавать, — это не откат. Это повторение с нуля. Другой экземпляр. Другая история. Другая судьба. Ты не исправил ошибку — ты сбежал от неё, оставив за спиной труп заготовки.

Вот почему истинный мастер бережёт материал. Вот почему он учится откатываться, не разрушая. Вот почему отладка — это искусство не только действия, но и возвращения. Искусство, где каждый шаг вперёд подкреплён умением сделать шаг назад. Ибо только так, только через священный откат, хаос превращается в порядок, а ошибка — в опыт.

Откат, о великое искусство возвращения, возможен только при одном условии — условии, которое звучит как заклинание, как пароль для посвящённых. Материал после неудач-

ной операции должен быть способен — способен! — быть приведён к состоянию, неотличимому от состояния до операции. Не похожему, не приблизительному, не условно-приемлемому — а неотличимому. Чтобы взгляд мастера, самый придирчивый взгляд, самый чувствительный инструмент не нашли различий. Чтобы время, этот безжалостный судья, не обнаружило следа.

Вот как это выглядит в чистом виде, в той первозданной ясности, которую дарует строгая терминология. В моей терминологии, позвольте мне быть точным, ибо точность — сестра лада, это называется *цикл переработки универсального полуфабриката*.

И что же есть универсальный полуфабрикат? О, слушайте же, ибо здесь — ключ ко всему.

Универсальным полуфабрикатом называется запас, из которого можно за одну операцию переработки сделать другие различные запасы.

Не один-единственный унылый исход, не предопределённый путь раба, а веер возможностей. Сегодня — деталь для станка, завтра — заготовка для бочки, послезавтра — возвращение в исходный X, чтобы начать всё заново, но уже с иной песней.

Это свойство и есть то, что делает отладку возможной. Именно оно позволяет мастеру пробовать, ошибаться, откатываться и пробовать снова — не расходуя материал, не плодя горы брака, не проклиная судьбу за каждую неудачу. Уни-

версальный полуфабрикат — это материализованная надежда. Это запас, который не мстит за ошибки, а прощает их. Это плоть, способная перерождаться, как феникс из пепла, но без потери себя.

И вот почему **пятая экономика процветает** именно там, где царит универсальный полуфабрикат. Где материал не держит зла на мастера. Где каждая ошибка — всего лишь шаг в танце, а не смертный приговор заготовке. Где откат — не поражение, а метод. Где лад возвращается снова и снова, ибо материал помнит свою первозданную форму и готов вернуться к ней по первому зову умелых рук.

Если материал, с которым работает мастер, не обладает священным свойством обратной переработки одной операцией — откат, увы, невозможен. А без отката, как мы уже уяснили, невозможна и сама отладка. Из этого железного силлогизма следует вывод, не терпящий возражений: для отладки необходим материал, который по определению есть универсальный полуфабрикат.

Заметьте, даже отладка в малом — на одном-единственном материале, на одной несчастной детали — уже требует этого божественного свойства отката. Но экономически значимой, истинно значимой отладка становится лишь тогда, когда отлаженный, выстраданный, выверенный метод переносится на другие материалы, на другие детали, на совершенно иные контексты. Ибо иначе, клянусь, каждое новое применение — это изобретение велосипеда, мучительное,

нелепое, бесконечное. А это, позвольте мне сказать прямо, уже не отладка. Это шаманство. Пляска с бубном. Гадание на кофейной гуще. Всё, что угодно, но не систематическое, рациональное, прекрасное искусство возвращения лада.

И вот он, предел отладки, её великая цель, её сияющий горизонт. Метод, который работает на любом материале, удовлетворяющем условию универсальности. Только и всего. Материал требует от метода лишь одного, но зато какого! — способности к прямому и обратному превращению одной операцией. Материал, который этим свойством обладает, и есть, по определению, универсальный полуфабрикат.

В пределе, в своём безграничном стремлении к совершенству, отладка требует, чтобы любой материал в экономике — любой, слышите? — был универсальным полуфабрикатом. Чтобы не было бесповоротно испорченных заготовок, чтобы каждая ошибка становилась лишь шагом на пути, а не могильным камнем. Чтобы мир стал податлив, как глина в руках гончара.

И здесь мы подходим к самому прекрасному, самому дерзновенному примеру. Код. Да, код. Ибо код — это идеальный универсальный полуфабрикат. Совершенный. Не знающий износа. Не помнящий обид.

Представьте же: вы написали функцию. Она прекрасна, как стихотворение. Но, увы, она не работает. И что вы можете сделать, владея магией отладки? Вот что.

Первое. Откатиться к предыдущей версии. Одна опера-

ция — и вы там, где всё было чисто, невинно, полно обещаний. `Git revert` — и время потекло вспять.

Второе. Исправить. Внести правку, подобно тому, как хирург вносит скальпель в живую плоть, но без боли и без риска.

Третье. Проверить. Сурово, беспристрастно, всевидяще.

Четвёртое. Снова откатиться, если проверка выявила новую ошибку. И так — сколько угодно. Десять раз, даже сто! Одна и та же функция, словно река, текущая по одним и тем же берегам. Один и тот же файл, но каждый раз — чуть иной, чуть лучше, чуть ближе к ладу. Одна и та же история, где каждый эпизод учит, а не калечит.

И когда отладка завершена, когда баги разбежались, как тараканы от света, — функция работает на любых входных данных. Универсальность, запечатлённая в коде! Вы можете переписать эту функцию обратно в исходный замысел — обратная переработка — одной операцией. `Refactor`, этот священный глагол программистского пантеона. Исходный замысел — это та самая форма X , к которой можно вернуться в любой момент, чтобы начать всё заново, но уже с новым знанием.

Вот почему код — царь универсальных полуфабрикатов. Вот почему программист — жрец пятой экономики. И вот почему отладка, начавшая свой путь с камня и дерева, обрела своё подлинное величие в мире, где каждая строка может быть откатана, исправлена и возвращена к жизни. Бесконеч-

но. Без потери сути. Без платы за ошибку, кроме времени и внимания.

О, какое счастье — работать с таким материалом! Какое блаженство — знать, что за каждой неудачей следует не отчаяние, а всего лишь `git revert`.

Кодэтика

Вспомним же аристотелевский идеал человека — тот лущезарный образ, что вот уже два с половиной тысячелетия манит и мучает умы. Совершенство! Прекрасен телом, словно изваяние бога. Благороден душой, словно герой эпоса. Внешняя форма и внутренние добродетели — в полном, нерасторжимом ладу. Калокагатия, греческое чудо, где красота и добро сплетены в единый узел, который не развязать, не разрубить.

Красиво, до чего же красиво! Но — увы и ах — с тех самых пор мировая мысль, вся эта вереница гениев и эпигонов, так и не решила проблему единства этики и эстетики. Как его достичь, этот проклятый лад? Аристотель не объяснил. Он просто постулировал — как аксиому, как данность, как явление, не требующее доказательств. Калокагатия — это не рецепт. Не метод. Не технология, которую можно повторить в своей мастерской. Это портрет. Созерцай и восхищайся. А повторить — нельзя. Идеал сей — статуя.

Кант, этот честный кёнигсбергский мудрец, поступил, как

подобает человеку, не терпящему лжи. Этикет, сказал он, — царство долга. Категорический императив, суровый, как горная вершина. Всеобщий закон, пред которым равны все, от цесаря до последнего бедняка. Эстетика же — царство незаинтересованного удовольствия. Игра способностей, танец воображения и рассудка. Целесообразность без цели — прекрасное, которое не служит ничему, кроме самого себя.

Между ними пропасть. Бездна. Мост перекинуть нельзя. Кант пробовал и честно признал своё бессилие. Прекрасное может быть безнравственным — соблазнитель с ангельским лицом и дьявольской душой. Нравственное может быть безобразным — старая праведница в лохмотьях, чьё сердце чисто, но облик отвращает взор. Канта, надо отдать ему должное, это устраивало. Правда жизни, сказал он, горькая и непричёсанная. Но это, позвольте мне заметить, капитуляция. Кант не решил проблему. Он объявил её неразрешимой. Честно, но — чего стоит честность, если за ней не следует победа?

А вот Гегель, как обычно, попытался всё съесть за один присест. Красота, изрёк он, есть истина, облечённая в чувственную форму. Искусство — ступень самопознания абсолютного духа, где дух глядит на себя, как в зеркало, и впервые себя узнаёт. Хорошее искусство — истинно. Истинное искусство — нравственно. Злое искусство — просто плохое искусство, ошибка, заблуждение, недоразвитие.

В теории — о, как величественно! Панцирь из чистого

умозрения. На практике? Ни одного убедительного примера. Гегель так и не объяснил, почему «Преступление и наказание» — роман о мерзости, об убийстве, о мучительной совести, — великое произведение. Почему «Мёртвые души» — поэма о жуликах, о плутах, о торгашах душами, — прекрасна до слёз. Почему Босх, этот кошмарный визионер, чья греховная фантасмагория кишит чудовищами и муками, — нетленен, вечен.

Гегель, осмелюсь сказать, перепутал предмет — который может быть безобразен, мерзок, ничтожен, греховен — и форму — которая обязана быть прекрасна, стройна, гармонична. Разные вещи, совершенно разные! А он склеил их в одно целое, как неумелый ремесленник склеивает то, что должно быть соединено иначе. И склейка эта, увы и ах, развалилась. При первом же прикосновении живой, дышащей, грубой реальности.

А вот Ницше — о, этот взбесился! Вся ваша мораль, закричал он, сотрясая устои, — предрассудок слабых, цепь, которую стадо наложило на себя добровольно, чтобы защититься от сильных! Вся ваша красота — самообман сильных, услада, под которую они усыпляют свою волю! Подлинный человек — сверхчеловек, молния, прорвавшая серое небо обывательского существования, — творит ценности сам. Он сам, своими руками и своей волей, решает, что хорошо. Он сам, своим вкусом и своей мощью, решает, что красиво.

Этика и эстетика, провозгласил Ницше, слиты в воле к

власти — в этом едином, всеобъемлющем, дионисическом начале. Блестящий жест! Удар молотом по статуе Канта! Но, спрашивается, что же сверхчеловек делает? Что есть его творчество, кроме афоризмов и танцев? Ницше не сказал. Он ушёл в туман своих афоризмов, в соль своих парадоксов. Заратустра пляшет, говорит загадками, опускается в пропасть и восходит на гору, но ни разу не берёт в руки резец или скрипку. Сверхчеловек, по Ницше, может написать гениальную музыку — и в тот же день зарезать другого человека. И то и другое будет прекрасно и этично — потому что такова его воля, потому что он сверхчеловек, потому что критериев выше него нет.

И вот с таким критерием можно оправдать всё. Абсолютно всё. Даже Освенцим — если архитектор печей искренне чувствовал свою волю к власти, если эсэсовец, расстреливающий ребёнка, переживал при этом возвышенный подъём духа. Ницше не решил проблему, о нет. Он её взорвал. После этого взрыва не осталось ничего — ни этики, ни эстетики, ни их единства, ни даже самих этих слов, наполненных смыслом.

Модернизм, этот холодный и гордый век, пошёл другим путём — путём отрицания содержания. Модернизм утверждает, что красота — исключительно в форме. Форма совершенна — значит, содержание не важно, вторично, подсобно. Беспредметная живопись, где краски текут по хосту сами по себе. Додекафония, где звуки подчинены лишь следованию.

Чёрный квадрат — пустота, объявленная вершиной. Этично ли это? Вопрос снят, объявлен некорректным. Искусство для искусства. Этика для проповедников. Проблема не решена — она вынесена за скобки, как дурной сон, как старый хлам, как нечто, недостойное внимания истинного творца.

Постмодернизм, этот весёлый могильщик великих нарративов, пошёл ещё дальше — в ничто. Никакого единства нет и быть не может, заявляет он, усмехаясь. Есть игры. Цитаты, вырванные из контекста и склеенные в коллаж. Симулякры, копии без оригинала. Добро и зло — всего лишь риторические фигуры, удобные для манипуляции. Красота и уродство — социальные конструкты, которые можно менять как перчатки. Можно выставить писсуар и назвать это искусством — и это будет искусством, потому что так сказал автор. Можно надеть нацистскую форму, выйти на площадь и назвать это перформансом — и это будет перформансом, потому что ирония снимает все обвинения. Всё разрешено, ибо ничто ничего не значит.

Проблема единства этики и эстетики при таком подходе... исчезла. Как бессмысленная. Как вопрос о том, сколько ангелов может уместиться на кончике иглы. Но исчезновение проблемы — это вам не решение! Это капитуляция второго порядка — ещё более позорная, чем кантианская. Кант хотя бы честно сказал: не могу, признаю своё бессилие. А постмодернизм делает вид, что ему просто неинтересно, что он выше этого.

Искали, искали, искали... Но искали не там.

Аристотель искал в добродетельном человеке — но оставил нам лишь портрет, не дав метода. Кант искал в трансцендентальной способности суждения — но развёл этику и эстетику по разным комнатам, запретив им встречаться. Гегель искал в абсолютном духе — но склеил то, что не склеивается, и склейка развалилась. Ницше искал в сверхчеловеке — но взорвал всё, включая самого сверхчеловека. Модернисты искали в гении-творце — но вынесли этику за скобки, оставив форму без содержания. Постмодернисты искали в ироническом наблюдателе — но растворили всё в игре, где нет ни добра, ни зла, ни красоты, ни уродства, ни даже самого наблюдателя.

И мы остались с вопросом. С тысячелетним, изъезженным, проклятым вопросом: как соединить добро и красоту? Как сделать так, чтобы правильное было прекрасным, а прекрасное — правильным? Где тот метод, тот алгоритм, та отладка, которая способна привести в лад этику и эстетику?

Может быть, ответ спрятан там, где мы его не ждали? Может быть, он не в душе, не в трансценденции, не в абсолютном духе, не в сверхчеловеке, а в чем-то куда более прозаичном и куда более удивительном? Может быть, он — в универсальном полуфабрикате? Может быть, в коде? Может быть, в той самой пятой экономике, где материал прощает ошибки, где отладка возвращает лад, а мастер отвечает за каждое своё действие, потому что не может спрятаться ни за бога,

ни за императив, ни за иронию?

Мы на пороге. Зайдём ли?

Никто не искал в отладке. Никто из великих — ни Аристотель с его калокагатией, ни Кант с его пропастью, ни Гегель с его склейкой, ни Ницше с его взрывом, ни модернисты, ни постмодернисты. Ни один из них, осмелюсь утверждать, никогда не брал в руки бракованную деталь и не возвращал её к жизни. Ни один не стоял у станка, вслушиваясь в ритм, чтобы уловить момент, когда железо запоёт в лад. Ни один не отлаживал код до тех пор, пока компилятор не останавливался с уважением.

А ведь ответ — вот он. Прямо перед нами. В пыли мастерской. В свечении экрана. В терпеливом движении рук, которые пробуют, ошибаются, откатываются и пробуют снова.

Единство этики и эстетики — это не свойство человека. И не свойство объекта. Не добродетель души и не качество вещи. Это свойство процесса. Процесса отладки.

Что такое этика, если снять с неё тысячелетние напластования проповедей, запретов и императивов? Этика — это действие, возвращающее лад. Не наказание, ибо наказание не возвращает, а карает. Не поощрение, ибо поощрение не восстанавливает, а стимулирует. Не долг, который давит на плечи тяжестью невыполненного обещания. Не императив, который командует, не спрашивая согласия. Этика — это восстановление порядка в ситуации, где порядок нарушен. То есть, позвольте мне сказать прямо, — отладка.

Что такое эстетика, если снять с неё налёт таинственности и музейной пыли? Эстетика — это восприятие лада. Когда механизм работает без сбоев — слух улавливает ровное, успокаивающее гудение. Когда форма и функция совпадают до неразличимости — взгляд находит покой в этой полноте. Когда материал и метод находятся в согласии, когда инструмент понимает заготовку, а заготовка не сопротивляется инструменту, — тело чувствует гармонию. Когда ничего не скрипит, не заедает, не выпадает, не хрустит под пальцами. То есть — отлаженное состояние.

И вот главное, то, что не могли увидеть философы, потому что слишком высоко парили над землёй. Отладка соединяет этику и эстетику в одном действии. В одном-единственном, конкретном, рукотворном действии. Потому что отсюда следует утверждение, которое для непосвящённого звучит странно, почти еретически. Но вы, слушающие, уже не непосвящённые. Вы прошли путь от станка до кода, от лада до отладки.

Красивый код — этичен.

Да, именно так. Утверждение, которое режет слух тем, кто привык отделять этику от эстетики, а программирование — от жизни, а технику — от человечности. Только странно, спешу вас уверить. Не ложно. Странно — как странно выглядит истина, когда она впервые входит в комнату, где все привыкли к полумраку лжи.

Что же такое красивый код? Поясню, ибо это важно.

Красивый код — понятен. Во всяком случае, он понятнее кода некрасивого. В этом, и только в этом, заключена его красота. Не в отступах, хотя и в них тоже. Не в том, нравится ли он глазу программиста-эстета. В том, что его можно легко прочитать. Войти в его логику, как входят в хорошо освещённую комнату. Проследить за ходом мысли, как прослеживают русло реки от истока до устья. Увидеть, что происходит от первой строки до последней, — без чёрной магии, без сюрпризов, без стыдливо спрятанных костылей.

А понятный код — этичен. Потому что код пишут не для себя. Или не только для себя. Код пишут для других. Для того неведомого путника, который будет его читать — через день, через месяц, через год. Для того бедного отладчика, который будет в нём копать в три часа ночи, пытаясь понять, почему упал сервер. Для того, кто придёт после — и скажет либо «спасибо, всё ясно», либо, скрежеща зубами, «кто же это написал?». Или для себя, но через полгода — а это, позвольте вам напомнить, уже совсем другой человек. Который не помнит, почему здесь написано именно так, а не иначе. Который смотрит на свой собственный код, как на древний папирус, и шепчет: «Что за бред?»

Понятный код не прячет ошибок в тёмных углах. Не заставляет читателя гадать на кофейной гуще. Он обнажает замысел автора — без стеснения, без защиты, без лукавства. И в этом обнажении, в этой честности, в этом служении тому, кто придёт следом, — и заключается его этика. Красота,

ставшая добродетелью. Добродетель, ставшая красотой. Лад, наконец.

Что же говорит о нас красивый код? О, он говорит многое, и первое, самое главное — признание. Красивый код торжественно признаёт: **я не один**. Читатель есть. Этот незримый, но неизбежный спутник, который придёт после меня — или даже вместо меня. Читатель будет мучиться с моим кодом, как путник, бредущий через бурелом. Читатель будет тратить своё время — время, которое он мог бы провести с семьёй, или за чашкой кофе, или просто в блаженном ничегонеделании. Читатель может возненавидеть меня, о, как легко и праведно возненавидеть, если я сделаю его жизнь хотя бы на йоту сложнее, чем она могла бы быть.

Поэтому я, автор, приму на себя эту ответственность. Я сделаю код понятным. Размеренным, как дыхание. Регулярным, как биение пульса. Красивым — не для самолюбования, а для облегчения чужого пути.

Но не обольщайтесь, слушающие. Красивый код — это не всегда лёгкий код. Не всегда тот, который написался в полчаса под весёлую музыку. Бывает, как часто бывает, что сама по себе задача сложна, как гордиев узел. Алгоритм запутан, словно лесная чаща. Данные приходят неизвестно откуда — из трёх разных систем, ни одна из которых не документирована. Требования противоречивы, как показания свидетелей на допросе.

Но красивый код, и в этом его величие, не добавляет к

этой уже существующей сложности свою — лишнюю, ненужную, паразитическую. Он не множит сущности без необходимости — запомните это, ибо это золотое правило. Не плодит лишних переменных, которые будут путаться под ногами. Не строит лабиринтов из вложенных условий, где теряется и сам автор. Он — регулярен. Если есть правило — оно соблюдается везде, от первой строки до последней, без капризов и исключений ради исключений. Если есть паттерн — он повторяется с аккуратной предсказуемостью, как рефрен в песне. Если есть исключение — а они бывают, куда без них — оно явно отмечено, подсвечено, снабжено объяснением, чтобы читатель не гадал: это ошибка или замысел?

А теперь, перевернув медаль, взгляните на обратную сторону. **Некрасивый код.** О, что это за чудовище! И я утверждаю, торжественно и без колебаний: некрасивый код — этически порочен, хотя работает правильно, без ошибок. Нет, порочность его корнями уходит в самый фундамент человеческого общежития. Некрасивый код порочен потому, что он — закрыт для другого.

Другой, доведись ему разбирать этот код, не может в него погрузиться. Другой тратит часы — драгоценные, невозвратные часы! — на то, чтобы понять, что автор мог бы объяснить одной строчкой комментария или одним осмысленным именем переменной. Другой проклиная автора, сидя в тёмной комнате, потому что переменная названа просто «х», а функция разрослась до трёхсот строк, как борода до пояса, а

логика перепрыгивает с места на место, как блоха на собаке, не давая ухватить нить.

И знаете, что говорит этот некрасивый код, этот немой свидетель чужого равнодушия? Он говорит, и слова его леденят душу: **«Мне плевать на тебя. Я сделал так, как мне удобно, в ту минуту, когда мне было лень думать. Разбирайся сам. Твоя боль — не моя проблема. Твоё время — не моя забота. Ты сам как-нибудь... выкручивайся».**

Вдумайтесь, прошу вас. Это не техническая проблема. Это не вопрос выбора языка программирования, редактора или операционной системы. Это этический выбор. Выбор между уважением и эгоизмом, между заботой и безразличием, между ладом и хаосом.

Выбор в пользу эгоизма против уважения — каждый раз, когда ты называешь переменную «а» вместо «ИдентификаторПокупателя». Выбор в пользу хаоса против порядка — каждый раз, когда ты вставляешь очередной костыль, вместо того чтобы переписать кривое место. Выбор в пользу одиночества против сообщества — каждый раз, когда ты думаешь «моя работа — написать, а читать и отлаживать будут другие, и это уже их проблемы».

И вот что получается. Код становится зеркалом души пишущего. Если душа стремится к ладу, если она помнит о том, кто придёт следом, если она уважает время и силы другого — код выходит красивым, понятным, этичным. Если же душа глуха к другому, если она занята только собой и своим

сиюминутным удобством — код выходит безобразным, запутанным, порочным.

Аристотель с его калокагатией, Кант с его императивом, Гегель с его абсолютным духом — все они искали этику и эстетику где-то высоко, в небесах, в трансценденции. А они — вот они. В отступе. В имени переменной. В решении не полениться и переписать функцию. В скромном комментарии, который спасёт чей-то час. В красивом коде, который говорит: «Ты не один. Я думал о тебе. Я сделал так, чтобы тебе было легче».

Это и есть лад. В действии. В отношении. В выборе, который каждый из нас делает — не раз в жизни, а каждый день, каждый час, каждую строчку. Выбор между хаосом и порядком. Между одиночеством и сообществом. Между злом и добром, которое на поверку оказывается просто — красотой.

Сообщество программистов — это экономика. Не та, первая, где собиратель приносит добычу, и не третья, где крестьянин возделывает землю. Не четвертая, где ремесленник точит деталь. Нет. Сообщество программистов — это пятая экономика. Та самая, где отладка стала ведущей деятельностью, где алгоритмы — основной и самый драгоценный запас, где код не пишут для галочки, а читают, отлаживают, передают из рук в руки, как скрижали завета.

И в этой экономике, важно понять, некрасивый код — это не просто ошибка вкуса. Не просто небрежность. Это нарушение присяги. Не перед богом, хотя и перед ним тоже, ибо

бог, как известно, любит порядок. Но в первую очередь — перед коллегой. Перед тем, кто сядет за клавиатуру после тебя, кто будет всматриваться в твои строки, кто будет сражаться с твоей небрежностью. Ты дал молчаливую клятву, когда взялся писать код для других. Клятву, что не сделаешь их жизнь сложнее, чем она есть. И некрасивый код — это клятвопреступление. Чистой воды.

Позвольте же мне явить вам две функции. Два мира. Две этики. Две красоты.

Функция А. Смотрите на неё, ибо она — портрет небрежности, этот молчаливый крик эгоизма:

```
def f(a,b): c=[]  
for i in range(len(a)): if a[i]>b:  
    c.append(a[i]*2) return c
```

Что делает эта функция? Остановитесь. Подумайте. Расшифруйте эти иероглифы. Вспомните, что «f» означает, вероятно, filter — но уверены ли вы? Что «a» — это array, а «b» — bound, но откуда вам это знать? Догадайтесь, что «c» — это результат, но почему именно «c», а не что-то осмысленное? Понимание, о, какое мучительное понимание требует усилий. Лишних. Ненужных. Тех, которых можно было бы избежать, если бы автор потратил лишние десять секунд, чтобы подобрать имена.

Функция Б. А теперь — вот она, красота, обретшая форму:

```
//Функция возвращает список, содержащий удвоенные
```

элементы исходного списка, которые превышают заданный порог.

ФУНКЦИЯ отобразить_удвоенные(список, порог)

результат = пустой массив[]

ДЛЯ КАЖДОГО элемент ИЗ список ЦИКЛ

ЕСЛИ элемент > порог ТОГДА

результат.добавить(элемент * 2)

КОНЕЦЕСЛИ

КОНЕЦЦИКЛА

ВЕРНУТЬ результат

КОНЕЦФУНКЦИИ

Что делает эта функция? Ясно с первого взгляда, даже если вы не программист. Не надо гадать, что такое «a» и «b». Не надо вспоминать, что «f» должно означать. Не надо перечитывать дважды, а то и трижды, мучительно шевеля губами. Всё сказано. В имени функции — действие. В имени параметров — их суть. В комментарии — замысел. В каждой переменной — человеческий язык, а не математический ребус.

Обе функции работают одинаково. Алгоритм их неотличим, как две капли воды. Но функция Б — красива. И я утверждаю это не как эстет, любующийся отступами и строчными буквами. Она красива потому, что написана для другого. Потому что её можно читать — как книгу, как письмо от друга. Потому что её можно отлаживать, не тратя драгоценное время на расшифровку чужого шифра.

Функция А — некрасива. И здесь самое важное. Это не эстетическая оценка в том поверхностном смысле, который вкладывают в неё музейные смотрители. Нет. Это этическая оценка. Суровая, беспощадная, заслуженная.

Автор функции А, сам того не осознавая или осознавая, но предпочитая не думать, сказал: **«Моё время важнее твоего»**. Моя минута, сэкономленная на том, чтобы не придумывать нормальные имена, дороже твоего часа, который ты потратишь на расшифровку. Моё удобство священо, а твоя боль — не моя забота. Я сделал, и мне плевать, что будет дальше.

Автор функции Б, напротив, сказал: **«Мы вместе в этом проекте. Я помогу тебе»**. Я знаю, что ты будешь читать мой код. Я знаю, что ты устал, что у тебя горят глаза, что дедлайн давит на плечи. Я не хочу добавлять тебе ещё одну проблему. Я хочу, чтобы, открыв мою функцию, ты вздохнул с облегчением, а не застонал от отчаяния. Я написал её так, чтобы она была понятной. Не потому, что меня заставят, а потому что я уважаю тебя и твоё время.

И вот где смыкаются этика и эстетика. В этом простом, каждодневном, незаметном подвиге. Автор красивой функции — этичен. Не потому, что он прочитал Канта или выучил десять заповедей. А потому что он помнит о другом. Потому что он возвращает лад в ту путаницу, которую неизбежно порождает любая совместная работа. Потому что он отлаживает не только код, но и человеческие отношения.

Аристотель искал добродетельного человека — и, возможно, не узнал бы его в программисте, пишущем красивый код. Но добродетель, она не меняется с веками. Она всё та же: уважение к другому, забота об общем деле, готовность потратить лишнюю минуту, чтобы сэкономить час товарищу. Только вместо меча и колесницы — теперь у нас клавиатура и компилятор. Вместо полиса — сообщество разработчиков. Вместо героя — тот, кто называет переменную не «х», а «порог».

Такова этика, ставшая красотой. И красота, ставшая этикой.

Пятая экономика, ещё раз, ибо это важно, — это экономика универсальных полуфабрикатов. Код в ней — не просто строки, не просто инструкции для бездушной машины. Код в ней есть основной, самый драгоценный, самый текущий запас универсальных полуфабрикатов. Код читают — как читают письма древних мудрецов, вчитываясь в каждый знак. Код отлаживают — как настраивают тончайший музыкальный инструмент, добываясь чистоты звучания. Код передают — как передают факел в эстафете, как завещают потомкам семейную реликвию. Код — это присяга. Торжественная, молчаливая, нерушимая. Ибо красота здесь — не украшение, а этический императив.

Тот, кто это понял — кто прочувствовал до костей, кто принял в сердце, кто превратил в привычку, — тот уже не просто программист. Тот — субъект пятой экономики. Не

винтик в механизме, не расходный материал, не наёмная единица. Тот — мастер, жрец, хранитель лада. Он держит присягу. Каждой строчкой кода, которую он пишет. Каждой функцией, которую он называет осмысленно. Каждым коммитом, который он отправляет в общее дело. И код его — красив. Не потому, что он старался — хотя и потому тоже. Не потому, что он прошёл курс по чистоте кода — хотя и это не помешало. А потому, что тот, кто думает о других, — этичен. Особенно когда пишет код. Ибо написание кода — это акт воли, акт выбора, акт создания порядка из хаоса. И в этом акте, как в капле воды, отражается вся душа пишущего.

Это экономическое отношение, эта связь между теми, кто пишет, и теми, кто читает, между автором и отладчиком, между мастером и подмастерьем, — я называю её особым словом. Словом, в котором слились воедино код и этика, ремесло и добродетель, железо и дух. Я называю это **кодэтика**.

Кодэтика — это не кодекс, не свод правил, не набор запретов и предписаний. Кодэтика — это способ существования в пятой экономике. Это дыхание мастера, который знает, что каждая его строка будет прочитана, осмыслена, отлажена. Это улыбка, с которой он называет переменную понятным именем. Это тихая радость, когда другой говорит ему: «Красивый код. Я понял сразу». Это присяга, которую он держит — каждую секунду, каждый миг, каждый раз, когда кладёт пальцы на клавиатуру.

Кодэтика — это лад.

Философия кодэтики

Позвольте мне это слово, ибо оно заслужено! — поскольку утверждает, что решение фундаментальной философской проблемы единства этики и эстетики, явившееся в кодэтике, не случайность. Не слепой удар судьбы, не шутка богов, не ирония истории. Но и не философский триумф. Не победа в споре, не доказательство теоремы, не окончательный ответ на измучивший всех вопрос.

Почему же? Почему это решение не стало и не станет событием в истории философии, громом среди ясного неба? А потому, что кодэтика — не философское учение. Она не родилась в академических залах, не была выстрадана на кафедрах, не облекалась в туманные формулы трансцендентальной эстетики.

Кодэтика — практика. Повседневная, рутинная, прозаическая. Программист, который пишет красиво — называя переменные понятными именами, структурируя код, добавляя комментарии, — и который считает это этичным, вовсе не решает проблему Аристотеля и Канта. Он, осмелюсь сказать, даже не знает о существовании этой проблемы. Он просто делает свою работу. И в этой работе противоречия нет. Красивый код — этичен. Непонятный, запутанный, ленивый код — порочен. Всё. Точка. Начало и конец разговора.

Философ же, высоколобый и грустный, глядя на это, ло-

мает голову, заламывает руки, впадает в трансцендентальную тоску. Как так? Тысячи лет спорили — Платон с Аристотелем, Кант с Гегелем, Ницше с постмодернистами. Не решились. Разошлись как в море корабли. А какой-то программист — безбородый юноша или усталый бородач в растянутом свитере — взял и соединил. Соединил то, что считалось несовместимым, как огонь и воду. Без теории. Без метода. Без категорического императива. Без абсолютного духа. Без воли к власти. Просто — потому что так удобнее отлаживать. Потому что так меньше времени тратится на расшифровку. Потому что так легче жить в сообществе.

И вот разгадка. Она проста. И оттого ещё более оскорбительна для утончённых умов.

Противоречия свойственны не бытию вообще. Не человеческой природе, которая, как уверяют поэты, едина и прекрасна. Не устройству мира, который, как учат физики, подчиняется единым законам. Нет. Противоречия свойственны фазе упадка. Только ей. Когда процветание кончается, когда лад уходит, когда трещины проступают на гладкой поверхности бытия — тогда начинаются мучительные дихотомии. Тогда добро встаёт напротив красоты, а истина — напротив пользы. Тогда этика и эстетика расходятся в разные стороны, как отвергнутые любовники. Трещины порождают разрывы. Разрывы — противоречия. И философы, эти врачи при умирающем теле духа, приходят описывать симптомы, классифицировать боли, возводить страдание в ранг вечной кате-

гории.

Но есть иная фаза. Фаза процветания. Фаза, где ничего не разорвано, где всё собрано, где механизм работает без сбоев, где форма и функция совпадают, где материал и метод находятся в согласии, где человек помнит о человеке, а коллега уважает коллегу. В фазе процветания, в этой благословенной фазе, противоречий нет. Они просто не возникают. Как не возникает головной боли у здорового. Как не хрустят колени у того, кто правильно ходит. Как не спорит этика с эстетикой у того, кто просто делает свою работу — хорошо, красиво, понятно, с уважением к тому, кто придёт следом.

Кодэтика — это и есть фаза процветания. Маленькая, локальная, техническая — но самая настоящая. В ней не нужно решать проблему единства этики и эстетики, потому что этой проблемы попросту нет. Она исчезает, как исчезает тьма, когда зажигают свет. Как исчезает холод, когда разгорается огонь. Как исчезает хаос, когда мастер произносит своё тихое, уверенное: «Я верну сюда лад».

И философ, стоящий на пороге этой процветающей экономики, может выбрать одно из двух. Либо продолжать ломать голову над вопросом, который потерял смысл, — и тогда он уподобится схоласту, обсуждающему природу ангелов в век телескопов. Либо же он может спуститься с небес на землю, взять в руки клавиатуру, написать красивую функцию и сказать: «Вот. Я сделал. Я соединил. Без слов. Делом».

Ибо лад, в конечном счёте, не постулируют. Лад — воз-

вращают. Строчкой за строчкой. Коммитом за коммитом. Отладкой за отладкой.

Когда первая экономика процветает. Соборность, о, какое древнее и тёплое слово! Все вместе, плечом к плечу, сердце к сердцу. Запасы общие, как воздух, как вода, как огонь в очаге. Этично здесь — то, что полезно общине, что помогает выжить, что укрепляет целое. Красиво здесь — то, что слаженно, где каждое движение вплетено в общий танец, где нет выпадающих, нет чужих. Пир, на котором все сыты — от мала до велика, от старца до младенца, — о, этот пир и этичен, ибо никто не голоден, и эстетичен, ибо щедрость стола и единство пирующих прекрасны. Нечего разделять. Не о чем спорить. Лад — в самой плоти общего дела.

Когда вторая экономика процветает. Наряд. Порядок, выстроенный как струна. Каждый знает свой надел — ни больше, ни меньше, ни раньше, ни позже. Этично здесь — соблюдать очередь в получении наряда, не лезть без спроса, дожидаться своего часа. Красиво здесь — когда ряд ровный, без пропусков, когда вереница тянется стройно, как журавлиный клин, когда ни одно звено не выпадает, не шатается, не рвёт мерного ритма. Поэтому наряд — о, это слово так много значит! — это не только про работу, не только про распределение, не только про справедливость. Наряд — это и про красоту. Стоять в наряде, идти в наряде, трудиться в наряде — и этика, и эстетика слиты в одном действии, как слиты форма и функция в хорошо сделанном инструменте.

Когда третья экономика процветает. Обряд поощрения. Лавры — тому, кто поощрил, кто заметил рост, кто воздал должное силам самовозрастания. Этично здесь — славить те самые силы, что умножают запас, что тянут вверх, что обещают завтра больше, чем есть сегодня. Красиво здесь — когда обряд исполнен вовремя, когда хула не запоздала, когда песня подхвачена всеми голосами, когда барабаны бьют в унисон. Разделять этику и эстетику в третьей экономике — всё равно что отделять танец от музыки, или пение от дыхания. Нечего делить. Всё едино. Ибо сам обряд поощрения — уже красота, уже добродетель, уже лад.

Когда четвёртая экономика процветает. Присяга. Нерушимое слово, которое даётся один раз, а держится всю жизнь. Этично здесь — держать обещание, не изменять договорённости, не отступать перед сбоями, не бросать дело на полпути. Красиво здесь — когда человек работает без сбоев, когда цепочка операций не рвётся, когда ритм ровен, как бой часов, когда в механизме ничего не скрипит, не заедает, не выпадает. И не зря в четвёртой экономике одно из высших удовольствий — смотреть, как другие работают. Смотреть — и видеть красоту. Смотреть — и понимать: этично то же самое. И здесь, как и прежде, этика и эстетика — одно. Не два разных царства, разделённых пропастью, а единая ткань процветающей жизни.

В процветании не возникает вопроса об их единстве. Потому что их единство — не проблема, которую нужно

решать философским инструментом, подгоняя понятия под ответ. Их единство — данность. Оно дано, как дано дыхание, как дано зрение, как дано чувство лада тому, кто вырос в ладу. Оно проживается — каждым действием, каждым выбором, каждым мгновением. Оно не требует доказательств — ибо что может требовать доказательств то, что очевидно, как солнечный свет? Вопрос о единстве этики и эстетики — о, этот вопрос вообще не возникает в голове у того, кто живёт в процветающей экономике. Он возникает тогда, когда процветание кончается. Когда лад уходит. Когда появляются философы, чтобы описать эту утрату и, может быть, горько пошутить о ней.

Философская проблема единства этики и эстетики — это не вечный вопрос человеческого духа. Это, осмелюсь сказать, симптом. Симптом упадка. Диагноз, а не пророчество. Боль, а не истина. Её не нужно решать. Её нужно — прожить. Исцелиться. Вернуться в фазу процветания. Где этика и эстетика, как два крыла одной птицы, машут в унисон, и никто не спрашивает, как они соединены, — они просто летят. И под ними — процветающая экономика, где каждая отладка возвращает лад, каждый красивый код держит присягу, а каждая строчка пишется с мыслью о том, кто придёт после.

Вот что такое кодетика. Не решение проблемы. А возвращение в фазу, где проблемы нет. Где этика и эстетика — одно. Где лад — не мечта, не идеал, не постулат. А повседневность. Тихая, уверенная, работающая.

Когда упадок первой экономики. Соборность рушится, как старая крепостная стена под ударами тарана. Каждый сам за себя — волки в человеческом обличье, разбегающиеся в разные стороны. Этично ли это? О нет, тысячу раз нет — ибо это предательство общины, измена общему делу, разрыв той самой ткани, что держала всех вместе. Красиво ли? Редкий человек, уходящий в лес, чтобы выживать в одиночку, красив. Обычно он жалок, озлоблен, оброс грязью и отчаянием. Нет в нём той античной гармонии, что воспевают поэты. И вот оно — первое противоречие. Должное говорит: останься, помоги, будь вместе. Сущее кричит: спасайся, каждый за себя, тонут другие — и слава богу. Никакого лада. Никакой красоты в правильном. Никакой этики в выживании.

Когда упадок второй экономики. Наряд нарушается. Кто-то получает больше — хитрый, ловкий, бессовестный. Кто-то получает меньше — честный, терпеливый, обманутый. Этично ли такое перераспределение? Нет, ибо это поправление того самого порядка, который составлял душу второй экономики. Красиво ли? Толпа, рвущаяся к запасам, давящая локтями, топчущая упавших, безобразна, как стая крыс, как муравейник, в который сунули палку. Противоречие зияет: справедливость требует восстановить наряд, но наряд уже не вернуть, и те, кто пытается его вернуть, выглядят смешными староверами, цепляющимися за прошлое.

Когда упадок третьей экономики. Обряд превращается в ритуал — в пустую, механическую, бездушную процеду-

ру. Слава — в потускневшую монету, которую бросают нищим, чтобы откупиться. Поощрение — в откуп, в формальность, в галочку в отчёте. Этично ли? Нет, ибо настоящая слава — это признание роста, а здесь нет ни роста, ни признания, одна бутафория. Красиво ли? Золотой телец — этот идол из чистого золота, — некрасив. Иначе бы его везде выставляли на обозрение, в каждый храм ставили, в каждый дом приглашали. Но его прячут. Стыдятся. Ибо золото без духа — это просто жёлтый металл, холодный, тяжёлый, мёртвый. Противоречие: поощрять надо, чтобы стимулировать, но поощрение, лишённое обряда, не стимулирует, а оскорбляет.

Когда упадок четвёртой экономики. Присяга заменяется договором. Договор — форс-мажором. Слово перестаёт быть делом — становится просто акустическим явлением, набором звуков, не имеющим силы. Этично ли нарушать обещание, если есть отговорка — уважительная, задокументированная, заверенная печатью? Юридически — да, ибо закон позволяет. Экономически — нет, ибо доверие рушится, а доверие — это кровь четвёртой экономики. Красиво ли? Судья, который ищет лазейку в законе, чтобы оправдать нарушителя, отвратителен. Его лицо, его жесты, его аргументы — всё в нём кричит о том, что красота ушла, осталась только казуистика. Противоречие: требование держать слово сталкивается с желанием найти выход, и ни один из путей не дарит гармонии.

В каждой фазе упадка этика расходится с эстетикой. Потому что нет больше лада. Нет согласия между тем, что должно — и тем, что есть. Нет соразмерности частей, нет внутренней стройности, нет той невыразимой правильности, когда вещь хороша и права одновременно. Нет красоты в правильном поступке — часто правильный поступок уродлив, как уродлива ампутация, спасающая жизнь. Нет этики в красивом явлении — часто красота обманчива, как пение сирены, как улыбка льстеца, как закат над полем боя, прекрасный и зловещий.

Философия, как правило, не застаёт процветания. Или — ещё хуже — философия описывает упадок как норму. Принимает его за устройство бытия, за вечную данность, за ту пещеру, из которой нет выхода. Ибо кто же пишет трактаты в эпоху пиров и общих запасов? Кто размышляет о единстве этики и эстетики, когда это единство проживается каждым вздохом?

Аристотель, величайший ум, жил в упадке полиса — когда греческие города-государства пожирали друг друга, когда Александр уже топтал восток, когда соборность первой экономики дышала на ладан. Кант — в эпоху, когда старый порядок рухнул под натиском Просвещения, а новый ещё не сложился, и мир шатался, как больной в горячке. Гегель застал великий перелом — революции, Наполеона, реставрации, крушения надежд. Ницше — предчувствие катастрофы, запах двадцатого века, который он чуял ноздрями про-

рока. А весь двадцатый век — эпоха войн, лагерей, распада империй, распада семей, распада душ.

Они описывали то, что видели. Своими глазами. Своей болью. Разрыв между должным и сущим, между добром и красотой, между поступком и восприятием. И приняли этот разрыв — по честности своей или по ограниченности — за неустранимый парадокс. За трагедию человеческого существования. За условие бытия, которое дано раз и навсегда.

Но они ошиблись. О, как они ошиблись! Ибо условие бытия — это лад. Процветание. Соборность и наряд, обряд и присяга — каждая в свою эпоху. А трещины, разрывы, противоречия — это симптомы. Болезни, которую можно вылечить. Упадка, который можно преодолеть. И кодэтика, эта маленькая, скромная, ремесленная практика, показывает нам, что возвращение к ладу возможно. Не навсегда, не для всего человечества, но — здесь и сейчас, в одной функции, в одном коммите, в одном отношении между тем, кто пишет, и тем, кто читает.

Философия, которая описывает процветание — о, как же она редка! — не порождает парадоксов. Ибо в процветании, в этой благословенной фазе, всё на своих местах. Каждый гвоздь вбит туда, куда нужно. Каждая шестерня вращается в такт с другими. Каждое слово находит своего слушателя. И потому этичный поступок — красив. Не через силу, не по обязанности, не вопреки обстоятельствам, а именно потому, что в процветании невозможно отделить правильное от пре-

красного. Красивое явление — этично. Не потому, что кто-то так решил, не потому, что так велит моральный кодекс, а потому что сама ткань процветающей жизни соткана из единой нити.

И вот что я должен сказать, и пусть эти слова прозвучат как итог, как тихая победа после долгой битвы. Кодэтика, эта скромная практика программистского сообщества, не решила проблему единства этики и эстетики. Решить проблему — значит дать ответ, формулу, рецепт, применимый всегда и везде. Кодэтика не дала такого рецепта. Но она сделала нечто большее! Она показала, что эта проблема исчезает — исчезает сама собой, как дым, как утренний туман — в процветании.

А значит это не фундаментальная проблема человеческого существования. Не проклятие, висящее над родом людским от сотворения мира. Не трагедия, которую нужно оплакивать, но нельзя разрешить. Это проблема историческая. Экономическая. Ситуативная. Она приходит с упадком и уходит с процветанием. Она не коренится в природе вещей — она коренится в наших ошибках, в наших сбоях, в нашем неумении держать лад.

Философам, осмелюсь сказать, стоило бы поучиться. Не у программистов, хотя и у них тоже — ибо программисты, как никто, чувствуют разницу между работающей системой и сломанной. Но у метода. У гуманитарного метода, который умеет отличать процветание от упадка. Метода, который го-

ворит, и говорит верно: ищи не решение парадокса. Не лмай голову над тем, как соединить этику с эстетикой, добро с красотой, долг с удовольствием. Ищи фазу, в которой этот парадокс исчезает. Найдёшь фазу — парадокс рассосётся сам. Как рассосался он в кодэтике. Как рассосался он в соборности, в наряде, в обряде, в присяге. Как рассосётся он везде, где люди перестают враждовать и начинают ладить.

И тогда не будет ни этики без эстетики. Ни эстетики без этики. Не будет измученного философа, вопрошающего небеса. Не будет юриста, ищущего лазейку. Не будет программиста, называющего переменную «х». Будет лад. Простой, ясный, дышащий лад. Тот самый, который древние называли гармонией, который плотники чувствуют в хорошо сделанной бочке, который музыканты слышат в слаженном хоре, который программисты видят в красивом коде. Лад, возвращённый отладкой. Лад, зафиксированный в универсальном полуфабрикате. Лад, который не нужно доказывать, ибо он — есть.

Цифровая экономика – исчерпывающий ликбез

Следует начать с главного — с того, что необходимо выучить наизусть, как таблицу умножения, как азбуку, как первые строки «Слова о полку». Цифровая экономика, экономика знаний, постиндустриальная, гипериндустриальная,

коммунизм (простите за это слово, но оно сказано), шестой технологический уклад, четвёртая промышленная революция и пятая экономика — это всё синонимы. Одно и то же явление. Один и тот же слон. Одна и та же восходящая звезда на экономическом небосклоне.

Почему же так много названий, спросите вы, запутавшись в этом лесу терминов? Почему каждый новый автор приносит своё имя, словно дары волхвов, и кладёт их к ногам непонятного чуда? А потому, что каждый, кто подходил к этому явлению — с разных сторон, с разными инструментами, с разными предубеждениями. Технолог, этот трезвый и практичный человек, видел цифру. И назвал — цифровая экономика. Социолог, ловец человеческих отношений, видел знания. И назвал — экономика знаний. Историк, взгляд которого обращён назад, но разум — вперёд, видел смену индустриальной эпохи. И назвал — постиндустриальная. Футуролог, этот гадатель на кофейной гуще прогресса, видел уклады и революции, сменяющие друг друга, как в калейдоскопе. Марксист, верный своему пророку, увидел — о чудо! — коммунизм. Именно так, с большой буквы и со светлым лицом.

Все они смотрели на слона. Каждый щупал свою часть — хобот, ногу, хвост, ухо. И каждый называл по-разному: «змея», «колонна», «верёвка», «веер». А слон — один. Живой, тёплый, дышащий. Имя ему — пятая экономика.

Я, следуя методу и не страшась ответственности, назы-

ваю эту экономику именно пятой. Не потому, что пятерка — магическое число, исполненное тайного смысла. Не потому, что пять — это красиво, а красивое, как мы знаем, этично. Нет. А потому, что так велит гуманитарный метод — тот самый, который не гадает на пальцах, а считает по фазам, по структурам, по сменяющим друг друга способам процветания.

В его счислении, в этой строгой периодизации, рассматриваемая экономика — по счёту пятая. После первой — где просто запас, накопленный, общий, соборный. После второй — где запас превращается в скопление, а скопление требует наряда. После третьей — где самовозрастание становится главным двигателем, а обряд — главным действием. После четвёртой — где присяга скрепляет ненадёжный мир, а отладка возвращает лад. Пятая — следующая. Не каприз, не прихоть нумерации, не игра в цифры. Структурная необходимость, железная необходимость самой логики экономического развития.

Ибо каждая фаза рождается из предыдущей, как бабочка из куколки, как колос из зерна. Четвёртая экономика — с её негарантированностью, с её отладкой, с её универсальным полуфабрикатом — подготавливает почву. А пятая — пожинает плоды. В четвёртой мы учимся возвращать лад. В пятой — лад становится естественным состоянием. В четвёртой отладка — героическое усилие. В пятой — повседневная работа. В четвёртой код пишут как попало, а потом правят. В

пятой код с самого начала красив. Ибо зачем писать некрасиво, если красивый код — этичен, а этичный — экономически эффективен?

Слон один. Называйте его как хотите.

Для целей популяризации, для тех скромных нужд, когда нужно быстро объяснить постороннему, что происходит, — название «цифровая экономика» подойдёт. Вполне. Даже хорошо. По крайней мере любой, кто услышит его, поймёт, о чём речь, не напрягая извилины. Компьютеры, залитые холодным светом мониторов. Интернет, эта всемирная паутина, опутавшая земной шар. Программисты, жрецы нового культа. Искусственный интеллект, о котором слагают и легенды и страшилки. Стартапы, дышащие авантюризмом и венчурным капиталом. Кремниевая долина, земля обетованная, где ночами не спят, а деньги текут рекой. Картинка ясна, ярка, узнаваема. Можно расходиться.

Хотя, строго говоря, если мы хотим не просто узнать, а понять — позвольте мне это сказать, — «цифра» есть лишь способ реализации определённых аспектов пятой экономики в данных конкретных историко-технологических условиях. Способ, а не суть. Средство, а не цель. Тот самый хобот, за который ухватился технолог, но не сам слон целиком.

Что это значит? А вот что. Пятая экономика — она могла бы быть и без цифры. Легко. Свободно. Иначе. Если бы универсальные полуфабрикаты — эти драгоценные запасы, из которых можно сделать что угодно, а потом вернуть обрат-

но в исходное состояние одной операцией, — были доступны иным способом. Если бы кодэтика — это экономическое отношение, где красивый код держит присягу, а некрасивый признаётся порочным, — не требовала компьютеров и компиляторов. Если бы отладка велась на устройствах, основанных на иных физических принципах — не на кремнии, не на транзисторах, не на битах и байтах, а на чём-то другом, чего мы, может быть, даже не придумали ещё. Но история, эта капризная дама, распорядилась иначе. Цифра оказалась тем самым материалом, той самой случайной формой, которая позволила реализовать принципы пятой экономики здесь и сейчас. А может быть, и сама цифра — своей доступностью, своей обратимостью, своей дешевизной отката — подтолкнула мысль к этим принципам. Неважно, совершенно неважно. Причины остаются в кабинетах историков. Нам же важно следствие.

Важно другое. То, что я должен вбить в ваши головы, как гвоздь в доброе дерево. **Цифра — не суть.** Запомните. Суть — универсальный полуфабрикат. Суть — замкнутый цикл переработки, где неудачную пробу можно обратить в исходное состояние одной операцией, не выкидывая материал, не начиная с нуля на другом экземпляре. Суть — отладка как ведущая деятельность, как главное, чем заняты люди в процветающей экономике. Суть — кодэтика как экономическое отношение, где уважение к другому становится экономическим императивом, а красивый код признаётся этичным и

эффективным.

Цифра — это кожа. Только кожа. Кожа, в которую обернута пятая экономика, как в одежду, как в покров. Кожа может быть разной — белой, смуглой, гладкой, шершавой, покрытой татуировками логотипов. Но тело — одно. Оно дышит, оно живёт, оно работает по внутренним законам, которым нет дела до цвета кожи. Те, кто путает кожу с телом, кто в упор не видит разницы между внешним и внутренним, называют экономику «цифровой» и думают, что всё сказали. Что поставили точку. Что достигли понимания. Нет, тысячу раз нет. Они сказали только о внешности. О внутренностях — молчат. Потому что не знают. Или знают, но не говорят — потому что цифра, клянусь, это понятно, это лежит на поверхности, это не требует усилий. А универсальный полуфабрикат... О, универсальный полуфабрикат требует мысли. Требуется, чтобы вы засучили рукава, нахмурили лоб, перестали скользить по верхам и нырнули вглубь. Требуется, чтобы вы оторвались от яркой картинки и спросили себя: а что там, под кожей? Что заставляет этот организм дышать, двигаться, процветать?

И когда вы спросите — и когда ответите, — вы увидите, что цифра была лишь случайностью. Исторической, технологической, даже счастливой случайностью, но — случайностью. А суть остаётся. Универсальный полуфабрикат. Отладка. Кодэтика. Лад.

Пройдемся по другим синонимам.

«**Экономика знаний**» — вот ещё одно имя, которое часто произносят с придыханием. Что же оно означает, если снять с него налёт интеллигентской возвышенности? Оно указывает прямо и честно на то, что именно в пятой экономике отчуждается. В нынешней, четвёртой, экономике отчуждается, как мы знаем, труд. Рабочее время, усилия, пот, мозоли, бессонные ночи перед дедлайном. В цифровой же — простите, в пятой — экономике отчуждаются знания. Сами по себе. Как сущность. Как то, что можно передать, не теряя, и получить, не затрачивая труда, кроме труда понимания.

Может возникнуть законный вопрос. Разве сегодня знания не отчуждаются? Например, в виде книг, этих мудрых кирпичей, заполняющих библиотечные полки? Разве автор, написавший книгу, не отчуждает свои знания читателю? И да, и нет. Отчуждаются, но — не в экономическом смысле. Ибо платят сегодня не за знания как таковые. Платят за затраченный на отчуждение знаний труд. За часы, проведённые за письменным столом. За количество слов, выдвленных из себя, как кровь из камня. За объём. Про толстые фолианты так и говорят — «объёмный труд», «монументальный труд», «капитальный труд». Никто не говорит — «объёмное знание». Знание может быть глубоким, но коротким. А труд — всегда измеряется временем. Вот вам и ключ.

Знания, если уж мы заговорили о них, делятся на две великие категории. **Декларативные** — те, что описывают мир. «Вода мокрая», «Париж — столица Франции», «код-

этика соединяет этику и эстетику». Это знания-факты, знания-картинки. И **процессуальные** — те, что описывают действие. «Как испечь хлеб», «как отладить программу», «как вернуть лад». Это знания-алгоритмы, знания-методы, знания, которые можно выполнить. Пример из жизни, из той самой цифровой повседневности: сайт показывает пользователю описательные знания — те самые декларативные, которыми люди делятся друг с другом, пишут статьи, комментируют, советуют. Отчуждают, да, отчуждают свои знания о мире. Но показываются эти знания по определённому алгоритму — процессуальному знанию. Алгоритм этот знает программист, который написал код сайта, и этим знанием он поделился — «отчуждил» от себя, выплеснул в строки — знания о том, как именно показывать контент, в каком порядке, с какими условиями, с какими исключениями. Вот вам две формы отчуждения. Одна — для пользователей. Другая — для машины. Но и та и другая — знание.

Сегодня программистам, как и, например, токарям, как и любым наёмным работникам четвёртой экономики, платят не за знания. Платят за время труда. За час, за день, за месяц, проведённые у станка или перед экраном. Хотя — это глубоко неправильно. Это порождает противоречия, которые копятся, как снежный ком, как грозовая туча перед бурей. Программист создаёт универсальный полуфабрикат, который можно тиражировать бесконечно, не затратив ни секунды дополнительного труда. А платят ему за те часы, что

он сидел и писал. Противоречие. Но оно, и я утверждаю это твёрдо, разрешится само собой в тех условиях, когда цифровая — пятая — экономика станет ведущей. Когда отчуждаться будут не труд, а знания. Когда платить будут за алгоритм, а не за время его написания. Когда красивость кода станет экономической категорией. Это — вопрос времени. И метода.

«Постиндустриальная». А вот это имя — с приставкой, с намёком, с поклоном в сторону истории. Приставка «пост» указывает на место. Место цифровой экономики — по отношению к нынешней, четвёртой. Если нынешнюю называть индустриальной — а это, строго говоря, тоже одна из её масок, четвёртая экономика вполне может быть индустриальной, с её заводами, станками, ремеслом и механизмами. «Пост» означает «после». Не «вместо», заметьте. Не «взамен», не «поверх». А просто — после. После индустриальной. Как индустриальная была после аграрной. Как аграрная была после нарядной. Приставка «пост» не отменяет предыдущее — она надстраивается, опирается, наследует. Заводы никуда не денутся. Станки останутся стоять в цехах. Ремесло не умрёт. Просто над ними, поверх них, вокруг них вырастет новая ткань — знаниевая, алгоритмическая, кодэтическая. И та, прежняя, индустриальная экономика продолжит существовать, но уже как подчинённая, как питательная среда, как материал для пятой. Как аграрная экономика не исчезла с приходом индустриальной — она просто перестала

быть ведущей, определяющей, задающей тон. Так и здесь.

«Постиндустриальная» — честное имя. Оно говорит о следовании, о преемственности, о том, что новое приходит не на пустое место.

«Гипериндустриальная». Приставка «гипер» — о, это вам не скромное «пост», за которым многие норовят спрятаться, радостно потирая руки: ура, индустрия кончается, наконец-то! «Гипер» означает иное. Сверх. Над. Больше. Гораздо больше. Несравнимо больше. И указывает она, эта гордая приставка, на то, что индустриальности — самой сути индустрии, с её ритмом, повторяемостью, измеримостью — будет больше. Не меньше. Вопреки всему, что шепчут на ухо легковверным пророки постиндустриального рая. Не кончается индустрия, о нет. Усиливается. Прорастает сквозь асфальт. Заполняет собой те уголки мира, куда раньше боялась сунуть нос. И уже становится больше — примеры вокруг.

На протяжении всего существования чисто индустриальной экономики, на протяжении XIX и XX веков, были сферы, которые практически не индустриализировались. Стояли в стороне, как партизаны, как партизаны от истории, не желающие маршировать в ногу. Первая среди них — сельское хозяйство. Это поле, это небо, это дождь, это солнце — где тут индустрия с её цехами и конвейерами?

Сельское хозяйство, надо отдать ему должное, охотно пользовалось достижениями индустрии. Трактор вместо лошади — железный конь, не знающий усталости. Комбайн вме-

сто серпа — пожиратель хлебных нив. Удобрения из химических заводов — пища для истощённой земли. Но само индустрией не было и не становилось. Поле — не цех, его не накроешь крышей. Посев — не конвейер, здесь нельзя задать единый ритм, ибо каждая травинка растёт по-своему. Урожай зависит от погоды, от капризов небесной канцелярии, а не от слаженной работы станков. Брак — от вредителей, от полчищ саранчи или фитофторы, а не от разболтавшейся направляющей или затупившегося резца.

Теперь всё меняется. На глазах. Со свистом и щелчком цифровых реле. Сельское хозяйство становится индустрией. Вертикальные фермы — небоскрёбы, полные зелени, где этажи отданы под салаты и помидоры, а свет и вода дозируются с аптекарской точностью. Гидропоника — никакой земли, только питательный раствор, только корни, купающиеся в химии. Климат-контроль — за окном пурга или зной, а внутри вечная весна, двадцать два градуса, шестьдесят процентов влажности. Роботы-сборщики, аккуратные, как ювелиры. Дроны-опрыскиватели, жужжащие над полями, наносящие яд точечно, по расчёту. Датчики влажности, зарытые в землю, передающие данные на сервер каждые пять минут. Урожай — прогнозируемый, как график на бирже. Ритм — задаваемый, как пульс кардиостимулятора. Брак — вычисляемый, как ошибка в коде, до того, как он проявился.

Заметьте главное, ибо здесь — ключ к пониманию гипериндустриальности. Лишь благодаря внедрению цифровых

технологий. Не трактор сделал сельское хозяйство индустрией. Трактор, осмелюсь сказать, — механизация. А механизация — это, если говорить на языке наших экономик, четвёртая экономика, рутинный труд, когда человек управляет машиной, но машина не управляет процессом. Цифра — автоматизация. А автоматизация — это уже пятая экономика, отладка алгоритмов, когда процессом управляет программа, а человек лишь настраивает, отлаживает, вмешивается в исключительных случаях. Датчик, сервер, программа, робот — вот что превращает поле в цех. Вот что делает невозможное возможным.

Позвольте мне пример — дотошный, как бухгалтерская проводка. Раньше, в незапамятные времена, коров доили вручную. Утром и вечером, с песней или с проклятьями, но обязательно своими руками. Количество молока зависело от опыта доярки — от её умелых пальцев. От настроения коровы — от того, как она спала и что ей снилось. От времени года — весной молоко жиже, осенью гуще. Индустриализация первой волны принесла доильный аппарат — механизм, который сосёт за человека, быстрее и равномернее. Стало лучше. Но учёт — в тетрадке, корявым почерком. Кормление — на глаз, примерно, «сколько не жалко». Осеменение — по календарю, без связи с реальным циклом животного.

Теперь — цифровая ферма. Датчик на каждой корове. Чип в ухе, ошейник с гироскопом, датчик жвачки. Система собирает данные в реальном времени: сколько съела, сколько

дала молока, сколько двигалась, когда лежала, когда выпасалась. Программа анализирует — не человек, не уставшая доярка, а безжалостный и точный алгоритм. Выдаёт рекомендации — или автоматически меняет рацион, добавляя протеина именно этой, конкретной Бурёнке. Отправляет сигнал роботу-дояру. Робот доит — без рук, без усталости, без «ой, корова боднулась». Робот чистит стойло — шаркает, скребёт, увозит навоз. Робот подгоняет корм — ровно столько, сколько нужно, именно в тот момент, когда нужно.

Ферма — индустрия. Не потому, что есть доильные аппараты или тракторы. Потому что процесс управляется алгоритмами. Ритмично. Предсказуемо. Масштабируемо. Одна операция за другой — как в цехе, как на конвейере, как на сборочной линии автомобилей. Корова, простите за цинизм, превращается в биореактор по производству молока. Но это не жестокость — это эффективность. Пятая экономика не убивает природу, она её упорядочивает. В этом её величие и её опасность, но об этом — в другой раз.

Строительство — другая сфера, долго, упрямо, героически сопротивлявшаяся индустриализации. Дом строили на месте. Из материалов, которые привезли — часто не те, не в том количестве, не вовремя. По погоде — если дождь, стой, если мороз, стой. По настроению бригады — если прораб не выспался, стена кривая. Цифра пришла — и навела порядок. BIM-моделирование — Building Information Modeling, если быть точным. Проект — не чертёж, не плоская картин-

ка, а база данных. Трёхмерная, умная, живая. Каждая деталь — в модели: её размеры, материал, стоимость, поставщик, срок изготовления, серийный номер. Спецификации — автоматически, не нужно считать вручную и ошибаться. Смета — автоматически, не нужно дёргать сметчика. Сроки — автоматически, программа рисует календарный график, где каждый блок знает своё место и своё время.

Заводы производят блоки — стены, перекрытия, лестничные марши — с чипами. Чип знает, куда этот блок должен быть установлен. Кран поднимает блок. Чип подсказывает крановщику (или беспилотному крану) куда нести. Монтажник сверяет с планшетом — зелёная галочка, если всё правильно. Дрон облетает стройку раз в час, сверяет реальность с моделью, сообщает об отклонениях в миллиметр. Строительство становится индустрией. Не «стройка» — это слово отдаёт грязью, временщиной, хаосом. А «производство зданий». С конвейера. С контролем качества на каждом этапе. С предсказуемым результатом — сдача объекта не «когда-нибудь в следующем квартале», а 15 марта в 10 утра, акт подписан, ключи вручены.

Благодаря цифровым технологиям практически всё станет индустрией. Логистика — уже стала: сортировочные центры, где алгоритм раскладывает коробки быстрее и умнее любой руки. Торговля — уже: интернет-магазины с предсказанием спроса, автоматическими заказами, роботизированными складами. Образование — постепенно: адаптивные

курсы, где программа подстраивается под ученика, а учитель отлаживает алгоритм. Медицина — постепенно: диагностика по снимкам, ИИ, ставящий предварительный диагноз, роботы-хирурги. Управление — постепенно: государство как платформа, услуги автоматизированные, бюрократия — отлаженный алгоритм.

Где есть алгоритм — там может быть индустрия. Где индустрия — там рутина, повторяемость, измеримость. Где рутина — там измеримое время, а не «сколько бог на душу положит». Где измеримое время — там эффективность, возможность считать, сравнивать, улучшать. Где эффективность — там процветание. Процветание пятой экономики. Не четвёртой, с её негарантированностью и героическими усилиями отдельных мастеров. А пятой — где процветание стало нормой, где отладка встроена в каждую операцию, где код этичен, а этика красива.

Поэтому, запомните, цифровая экономика — гипериндустриальная. Больше индустрии, чем было когда-либо. Индустрия там, где её не было — в поле, на ферме, на стройке, в школе, в больнице. Тот, кто думает, что цифра убивает индустрию, — о, как же он наивен! — тот не видел современной фермы с роботами-доярками и чипированными коровами. Не видел завода по производству окон, где каждый откос знает свои BIM-координаты. Не видел логистического центра «Амазона», где алгоритм раскладывает коробки быстрее любой человеческой руки — и не устаёт, не жалуется, не берёт

больничный. Пусть посмотрит, пусть своими глазами увидит — и поймёт. Индустрии стало больше. И будет ещё. Намного больше. Потому что цифра — не могильщик индустрии. Цифра — её ускоритель, её расширитель.

«Коммунизм». О, это слово, каким его только не осыпали — проклятиями и славой, насмешками и молитвами, забвением и возвращением. Отставим же в сторону идеологические страсти, красные флаги и чёрные клеветы. Коммунизм, в его строгом, научном значении — социологический термин. Не более, но и не менее. Обозначает он общественное устройство — способ, которым люди собираются в общество и распределяют между собой то, что производят. Определяется он отношениями между обществом и производительными силами — между всеми нами, с одной стороны, и тем, что мы умеем и чем мы работаем, с другой. А производительные силы, напомним на всякий случай, — это инструменты и станки, технологии и процессы, знания и навыки, алгоритмы и коды. Всё то, что позволяет превращать «ничего» в «нечто», хаос в порядок, сырьё в продукт. Коммунизм — это когда эти самые производительные силы работают на всех. На каждого, кто есть в обществе, без изъятия, без очереди, без «ты мне — я тебе». Просто — на всех. Ибо достаточно. Ибо хватает.

Утверждение о неизбежности построения коммунизма, которым пугали обывателей в одних странах и дразнили их в других, предсказывает, по сути, одну вещь: победу про-

летариата над буржуазией при определённом уровне развития производительных сил. Не раньше — потому что раньше нечем обеспечивать всех, и приходится терпеть посредников, эксплуататоров, перераспределителей. Не позже — потому что позже уже не будет смысла их терпеть, когда всё и так работает. Уровень этот — развитие цифровых технологий, автоматизация, искусственный интеллект, алгоритмы, управляющие заводами и фермами, логистикой и торговлей, коммуникациями и образованием. Производительные силы сегодня развиты настолько, что могут обеспечивать всех. Без буржуазии. Без её посредничества, которое всегда было не бесплатным. Без её прибыли, которая снималась как сливки с общего молока. Могут — технологически. Могут — экономически. Вопрос лишь в том, хватит ли воли и сознания, чтобы сделать этот шаг.

И вот здесь, заметьте, происходит неожиданный поворот. Использование цифровых технологий породило то, что я, следуя марксистской терминологии, но не марксистской риторике, называю «истинный пролетариат». Не заводских рабочих, спешу уточнить. Те — пролетариат, да, классический, индустриальный, воспетый в гимнах и запечатлённый на плакатах. Но не истинный. Почему? Потому что их труд, хотя и коллективный, всё же привязан к месту, к станку, к материалу, который нельзя скопировать Ctrl+C, Ctrl+V. Их отчуждение от средств производства — железное, буквальное. А истинный пролетариат — это IT-специалисты. В ос-

новном — программисты. Те, кто пишет код. Те, кто создаёт универсальные полуфабрикаты, из которых можно сделать всё, что угодно, и при этом не затратить ни грамма дополнительного материала.

Истинным пролетариатом я называю передовой отряд современного рабочего класса, полностью, без натяжек, подпадающий под марксистское определение пролетариата, данное ещё в «Коммунистическом манифесте», уточнённое в «Капитале», выверенное поколениями теоретиков. А именно: добывающий средства существования собственным интеллектуальным трудом — головой, а не спиной; и не являющийся собственником средств производства.

Компьютер, на котором пишется код, — не его собственность. Сервер, на котором код выполняется, — не его собственность. Платформа, на которой код приносит прибыль, — не его собственность. Алгоритм, который он создал, — не его собственность, ибо принадлежит нанимателю. Он продаёт свой труд — свой ум, свои часы, свои нейронные связи — тому, кто владеет цифровыми средствами производства. И получает за это зарплату. А вся прибавочная стоимость, весь тот гигантский излишек, который создаётся кодом, работающим на миллионы пользователей без дополнительных затрат, — оседает в карманах владельцев платформ. Это классическое, хрестоматийное, до боли знакомое отчуждение. Только вместо станка — клавиатура. Вместо цеха — опенспейс. Вместо заводской трубы — облачный сервер.

Но есть в этом истинном пролетариате одна особенность, которая меняет всё. Код, как мы знаем, — идеальный универсальный полуфабрикат. Его можно копировать бесконечно, не теряя качества. Его можно отлаживать, откатывать, переписывать, не расходуя материал. Его можно передавать другому — и он останется тем же самым. Программист, в отличие от токаря, может создать ценность один раз — а тиражировать её будет машина, без его участия. Это значит, что истинный пролетариат, сам того не замечая, владеет орудием, которое делает его... потенциально ненужным для процесса тиражирования. Нужным — только для отладки. Только для творчества. Только для того, что не может сделать машина. И в этом — его сила. И в этом — его слабость. И в этом — ключ к пятой экономике, где отладка становится ведущей деятельностью, а универсальный полуфабрикат — основным запасом.

Коммунизм, как общественное устройство, становится возможным ровно тогда, когда истинный пролетариат осознаёт себя — и когда средства производства, цифровые и алгоритмические, переходят в общественное владение. Не путём насилия, не путём экспроприации (хотя и это не исключено), а путём естественного, исторического, экономического процесса. Потому что, когда код может обеспечивать всех, зачем содержать буржуазию? Когда алгоритм сам распределяет ресурсы лучше любого посредника, зачем платить посреднику? Когда отладка становится главным занятием, за-

чем делить мир на владельцев и наёмников?

Но, как далеко до этого! Как много нужно ещё отладить — в экономике, в политике, в головах. **Пролетариат побеждает буржуазию.** Утверждает собственную диктатуру — власть, принадлежащую тем, кто создаёт, но не владеет. Оставаясь при этом пролетариатом. Не становится буржуазией — не обзаводится средствами производства в личную собственность, не нанимает других пролетариев, не начинает жить на дивиденды, не перекладывает труд на плечи наёмников. Остаётся тем, кто есть. Трудится. Создаёт. Не владеет. Власть у него есть — а собственности нет. Это противоречие. Нелепое, как летающая рыба. Невозможное, как квадратный круг. И коммунисты, все эти поколения теоретиков и практиков, так его и не преодолели.

Маркс, великий систематизатор, предполагал, что диктатура пролетариата — временная мера. Переходный период. Промежуток между старым и новым. Когда пролетариат, взяв власть, начнёт перестраивать экономику, социализировать средства производства, переучивать людей, выкорчёвывать старые привычки. А после этого переходного периода классы исчезнут. И буржуазия — как класс-эксплуататор. И пролетариат — как класс-эксплуатируемый. Ибо все станут собственниками средств производства — или, что то же самое по сути, никто не будет собственником. Потому что средства производства станут общими, общественными, ничьими в отдельности и всех вместе. И тогда, о тогда, дикта-

тура станет не нужна. Отпадёт. Как отпадает кокон, когда бабочка родилась. Не случилось.

История пошла другим путём. Ни в одной стране, где пролетариат побеждал, переходный период не закончился. Либо он растянулся на десятилетия. Либо пролетариат перерождался в новую буржуазию — номенклатуру, бюрократию, владельцев доступа. Либо власть объявляла себя «народной», а пролетариат оставался пролетариатом — работал, создавал, не владел. Противоречие осталось. Неразрезанный gordiev узел.

А теперь — цифра. Цифровой пролетариат — истинный пролетариат, программисты и IT-специалисты. Представьте: они побеждают. Власть переходит к программистам — к тем, кто пишет код, кто отлаживает алгоритмы, кто создаёт универсальные полуфабрикаты, из которых состоит пятая экономика. Но программисты не становятся капиталистами. Они продолжают писать код. Получать зарплату — из общего фонда, из налогов, из распределения, но зарплату, а не дивиденды. Не владеют станками — даже цифровыми. Не нанимать других программистов, чтобы те работали на них. Остаться пролетариатом — не по бедности, а по определению: добывающими средства существования собственным трудом и не являющимися собственниками средств производства.

Вот она, картина. Диктатура пролетариата, в которой пролетарии не имеют собственности. Власть у тех, кто трудится,

а собственность — у всех и ни у кого. Как это работает? Никто не объяснил. Маркс не дожил до цифры. Ленин и Троцкий думали о заводах, а не о компиляторах. Мао цедил слова о классах, а не о классах в коде. Современные теоретики, если они есть, молчат или бормочут что-то невнятное о «цифровом социализме», о «платформенном кооперативизме», о «общей собственности на алгоритмы». Но как именно устроена эта собственность? Кто владеет кодом, который написан программистом на рабочем месте? Кто владеет сервером, на котором этот код выполняется? Кто владеет данными, которые этот код обрабатывает? Если всё — общее, то как распределяется прибыль? Если прибыли нет — то откуда берутся средства для развития? Если развитие не нужно — то зачем тогда код? Вопросы, вопросы, вопросы. А ответов нет.

Парадокс пятой экономики: она возможна технологически, но не оформлена политически. Универсальный полуфабрикат есть. Отладка есть. Кодэтика есть. Гипериндустриализация есть. Даже истинный пролетариат есть. А вот того, кто скажет: «Власть — программистам, а средства производства — обществу», — нет. Или есть, но не слышен. Или слышен, но не убедителен.

Может быть, коммунизм — это не цель, к которой идут, а горизонт, который отодвигается. Может быть, диктатура пролетариата — не этап, а вечное состояние, в котором пролетариат правит, но не владеет, и это не противоречие, а но-

вая форма бытия. Может быть, собственность в цифровую эпоху перестаёт быть значимой категорией — когда доступ к алгоритму важнее, чем владение сервером, а право на отладку важнее, чем право на прибыль. Может быть, кодэтика — это и есть политическая программа пятой экономики. Не «код — всему голова». А «красивый код держит присягу». И присяга эта — не перед партией, не перед государством, не перед классом. Присяга — перед коллегой.

Это утопия? Да. Технологическая утопия, не хуже, чем марксова научная. Но, в отличие от марксовой, она уже работает. В сообществах open source. В командах, которые пишут код по стандартам. В компаниях, где code review — священный ритуал. Там, где некрасивый код отвергают не потому, что он приносит меньше прибыли, а потому, что он неуважителен. Там, где этика и эстетика срослись в кодэтику. Там, где пролетариат не ждёт диктатуры — он уже правит. Правит кодом. Правит собой. Правит ладом. А буржуазия... буржуазия ещё существует. Но она — на подхвате. Она обеспечивает сервера, платит налоги, покупает лицензии. Она — техническая деталь, а не класс-господин. Пролетариат её победил — тем, что перестал обращать на неё внимание. Победил — не в битве, а в повседневности. Не взял штурмом Зимний, а написал код, который сделал Зимний ненужным.

То, что программисты являются истинными могильщиками капитализма, буржуи поняли давно. Не вчера, когда ChatGPT напугал инвесторов до икоты и заставил их хва-

таться за сердечко. Не сегодня, когда очередная нейросеть рисует картины, от которых у арт-директоров подкашиваются колени. Нет. Ещё в шестидесятые годы прошлого века. Когда первые программисты, похожие на монахов в очках, набивали перфокарты дырокольными машинками, а компьютеры занимали целые этажи и жужжали так, что дрожали стёкла. Уже тогда, в этой колыбели цифровой эры, было ясно, ясно как божий день: эти люди создают не вещи. Не станки, не автомобили, не ботинки. Они создают алгоритмы. А алгоритмы заменяют целые отрасли. Уничтожают рабочие места — не по одному, а тысячами, десятками тысяч, сотнями тысяч. Делают буржуазию ненужной. Класс, который владеет средствами производства, — к чему он, когда средства производства слушаются алгоритма, а алгоритм пишет программист, который не владеет ничем, кроме своей клавиатуры и своей головы? Страшно. Страшно для тех, кто сидит наверху.

И что же мы видим сегодня, спустя шестьдесят лет? Программисты в США и Европе, в этих цитаделях капитализма, максимально, насколько это возможно, отдалены от бизнеса. Они не принимают стратегических решений — за них решают менеджеры, проджекты, продакт-оунеры, владельцы продуктов. Они не владеют акциями — в массе, не считая топ-менеджеров первых компаний и счастливиц с опционами, которые чаще всего оказываются бумажками. Они не участвуют в прибыли — получают зарплату, оклад, фикс, иногда

бонус, но не процент от того, что создали. Они — наемные работники. Как токарь. Как грузчик. Как уборщица. Только компьютер вместо станка. Только код вместо деталей. Только опенспейс вместо цеха. Пролетариат. Интеллектуальный. Образованный. Востребованный. Но — пролетариат. Рабочий класс, продающий свой труд за зарплату и не владеющий ничем, кроме самого этого труда.

Им не доверяют! Им не дают власти. Их держат на расстоянии — вытянутой руки, целого этажа, корпоративной политики. Технический директор — ещё может быть близок к бизнесу, он входит в кабинет к CEO, его мнение учитывают при стратегических решениях. Программист — нет. Программист — исполнитель. Ему ставят задачу — в Jira, в Trello, в Asana, на карточке, которую нельзя перепутать. Он пишет код. Сдаёт — запускает тесты, проходит код-ревью, получает апрув. Берёт следующую задачу. Никакого влияния на то, что вообще делать. Никакой доли в том, что получится. Никакой собственности на результат — код принадлежит компании, даже если авторство осталось в комментариях. Чистый пролетариат. Интеллектуальный — да, но от этого не менее обездоленный. И опасный — потому что только он понимает, как на самом деле работает система. Только он может её сломать. Только он может построить новую.

И, судя по тому, что мы видим на экранах, Голливуду было поставлено требование. Негласное, но железное. Не прописанное в контрактах, но читаемое в каждом сценарии.

Изображать IT-шников маргиналами. В идеале — дегенератами. Не людьми. Не героями. Не лидерами. Кем-то, на кого нельзя равняться. Кого нельзя уважать. Кого можно только... пожалеть. Или испугаться. Или посмеяться.

Примеры? Пожалуйста. Я приведу их, как прокурор — улики.

«Социальная сеть». Фильм о создателе Facebook. Цукерберг в исполнении Айзенберга — социопат с ледяными глазами. Не умеет общаться — каждое слово даётся ему с болью. Предаёт друзей — одного за другим, как будто это в порядке вещей. Крадёт идеи — у братьев Винкльвосс, у Эдуардо Саверина, у кого угодно. Гениальный — но дефектный. Его успех — не триумф, а диагноз. Он не герой. Он — предупреждение.

«Мистер Робот». Эллиот Алдерсон — параноик, разговаривающий с воображаемым отцом. Наркоман, глотающий морфин и сублимацию. С галлюцинациями — реальность для него неотличима от бреда. Хакер — гениальный хакер, да, но сумасшедший. Нельзя ему верить — он сам не знает, что сделает в следующую минуту. Нельзя за ним идти — он заведёт в пропасть. Хакер как безумец — это диагноз, а не профессия.

«Кремниевая долина». Сериал-насмешка, комедия, от которой почему-то не смеёшься, если ты программист. Программисты здесь — невротики, аутисты, фанатики гигиены, не способные связать двух слов с женщиной. Ричард — ге-

ний кода, но социальный калека. Джаред — мазохист с манией служения. Динеш — гик, помешанный на трёхмерных принтерах и борьбе. Гильфойл — ходячий синдром Аспергера. Смешно — да, для тех, кто смотрит со стороны. И страшно — для тех, кто узнаёт себя в этих карикатурах и понимает: обществу не нужен нормальный программист. Обществу нужен шут.

«Стартап». Хакеры здесь — криминал. Наркотики, насилие, перестрелки, подставы. Никакой созидательной силы. Только разрушение. Технология — это оружие. Код — это наркотик. Сервер — это поле боя. Ни одного светлого лица. Ни одного, кто делает мир лучше. Только те, кто его ломает, чтобы продать запчасти.

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.