

Добро пожаловать обратно в software engineering

Alisa Lafoks



Хорошая новость: мы снова становимся software-инженерами. А не кодерами.

Software engineering — это про решение задач. Как в математике: есть проблема — её нужно решить. Чем ты её решишь, на каком языке, каким инструментом — по большому счёту всё равно. Важен результат: задача решена или нет.

Вот это и было интересно. Сесть и думать. Найти решение, которого до тебя не было. Придумать что-то новое, заставить это работать. Интеллектуальная и творческая работа — за этим в неё мы и пришли. А не за тем, чтобы полдня воевать с конфигом, неделю с фреймворком или месяцы с инфраструктурой.

Содержание

Как инженеры превратились в кодеров	2
Брукс назвал это ещё в 1986-м.	2
AI забирает грязную работу	2
С возвращением	3

Как инженеры превратились в кодеров

А потом, со временем, software engineering незаметно превратился в кодинг. Не потому что инженеры поглупели — а потому что фреймворков и инфраструктуры развелось до неприличия. Каждую библиотеку надо знать, каждый сервис — подключить, каждый пайплайн — настроить.

И инженеры погрязли. Погрязли в кодинге и в настройке всего подряд: поднимать окружение, возиться с деплоем, чинить сборку, обновлять зависимости, гуглить, почему версия А не стыкуется с версией В. Погрязли в этом — а не в решении интеллектуальных задач, с чем, в общем-то, и связана наша профессия. Ни сил, ни внимания у них на творчество не осталось.

Вот и вышло, что инженер превратился в кодера. В человека, который целыми днями пишет код вокруг задачи, почти не доходя до самой задачи. Решение задач стало редким, почти праздничным состоянием — в короткой паузе между двумя порциями кодинга.

Брукс назвал это ещё в 1986-м

У этого есть имя. Фредерик Брукс разделил сложность софта на два вида: сущностную и привнесённую.^[1]

Сущностная — это трудность самой задачи. Её не убрать: она и есть инженерия, те самые интеллектуальные задачи. Привнесённая — это всё, чторосло вокруг: языки, фреймворки, инфраструктура, обвязка, инструменты. Кодинг ради того, чтобы всё это состыковать между собой.

Брукс предупреждал: серебряной пули нет, привнесённую сложность можно только подрезать. Тридцать лет её не подрезали — наращивали. Каждый новый фреймворк обещал упростить жизнь и добавлял ещё один слой кода, который теперь тоже надо писать и держать в голове. Так кодинг и съел инженерию.

AI забирает грязную работу

Впервые тот самый кодинг есть кому отдать.

AI берёт на себя ровно то, что писать скучно: код, бойлерплейт, перекладывание данных, склейку, «как тут в этом фреймворке сделать вот это». Нудную работу по написанию кода — именно кодинг. Грязную работу, в которой мы погрязли последние десятилетия. То, что Брукс называл accidental, машина делает быстрее и без нытья.

Многих это пугает: «AI заменит инженеров». Так вот, пугаются не в ту сторону. AI заменяет не инженера. Он заменяет кодера — ту часть нашего дня, которая инженерией никогда и не была.

С возвращением

А что остаётся нам, программистам? Остаётся сущностное. Решить, что мы вообще строим и зачем. Понять задачу. Выбрать форму. Принять решения, за которые потом отвечать. То самое трудное и интересное, те самые интеллектуальные задачи, ради которых мы сюда пришли.

И вот теперь мы, software-инженеры, наконец-то можем вернуться от кодинга к сути software engineering — к решению задач. От кодера обратно к инженеру.

Вот это и есть настоящий software engineering. Не борьба с инструментами — решение задач. Как в математике. Как было с самого начала.

Так что добро пожаловать обратно. Кодер у нас теперь машина — а мы снова инженеры.

И, честно говоря, жить снова стало интереснее.

[1] Фредерик Брукс, «No Silver Bullet — Essence and Accident in Software Engineering» (1986). Сущностная (essential) сложность — это сложность самой задачи; привнесённая (accidental) — сложность инструментов, которыми мы её решаем. https://ru.wikipedia.org/wiki/Серебряной_пули_нет