

Ласкаво просимо назад у software engineering

Alisa Lafoks



Хороша новина: ми знову стаємо software-інженерами. А не кодерами.

Software engineering — це про розв'язання задач. Як у математиці: є проблема — її треба розв'язати. Чим ти її розв'яжеш, якою мовою, яким інструментом — за великим рахунком байдуже. Важливий результат: задачу розв'язано чи ні.

Ось це й було цікаво. Сісти й думати. Знайти рішення, якого до тебе не було. Придумати щось нове, змусити це працювати. Інтелектуальна і творча робота — за цим ми в неї і прийшли. А не за тим, щоб півдня воювати з конфігом, тиждень — з фреймворком, місяці — з інфраструктурою.

Зміст

Як інженери перетворилися на кодерів	2
Брукс назвав це ще в 1986-му	2
AI забирає брудну роботу	2
З поверненням	2

Як інженери перетворилися на кодерів

А потім, з часом, software engineering непомітно перетворився на кодинг. Не тому що інженери подурнішали — а тому що фреймворків та інфраструктури розвелось до неподобства. Кожну бібліотеку треба знати, кожен сервіс — підключити, кожен пайплайн — налаштувати.

І інженери загрузли. Загрузли в кодингу та в налаштуванні всього підряд: піднімати середовище, морочитися з деплоєм, лагодити збірку, оновлювати залежності, гуглити, чому версія А не стикується з версією В. Загрузли в цьому — а не в розв'язанні інтелектуальних задач, з чим, власне, і пов'язана наша професія. Ні сил, ні уваги на творчість у них не лишилося.

Ось і вийшло, що інженер перетворився на кодера. На людину, яка цілими днями пише код навколо задачі, майже не доходячи до самої задачі. Розв'язання задач стало рідкісним, майже святковим станом — у короткій паузі між двома порціями кодингу.

Брукс назвав це ще в 1986-му

У цього є ім'я. Фредерік Брукс поділив складність софту на два види: сутнісну і привнесену.^[1]

Сутнісна — це трудність самої задачі. Її не прибрати: вона і є інженерія, ті самі інтелектуальні задачі. Привнесена — це все, щоросло навколо: мови, фреймворки, інфраструктура, обв'язка, інструменти. Кодинг заради того, щоб усе це стикувати між собою.

Брукс попереджав: срібної кулі немає, привнесену складність можна лише підрізати. Тридцять років її не підрізали — нарощували. Кожен новий фреймворк обіцяв спростити життя і додавав ще один шар коду, який тепер теж треба писати й тримати в голові. Так кодинг і з'їв інженерію.

AI забирає брудну роботу

Уперше той самий кодинг є кому віддати.

AI бере на себе рівно те, що писати нудно: код, бойлерплейт, перекладання даних, склейку, «як тут у цьому фреймворку зробити ось це». Нудну роботу з написання коду — саме кодинг. Брудну роботу, в якій ми загрузли останні десятиліття. Те, що Брукс називав accidental, машина робить швидше і без скигління.

Багатьох це лякає: «AI замінить інженерів». Так ось, лякаються не в той бік. AI замінює не інженера. Він замінює кодера — ту частину нашого дня, яка інженерією ніколи й не була.

З поверненням

А що лишається нам, програмістам? Лишається сутнісне. Вирішити, що ми взагалі будемо і

навіщо. Зрозуміти задачу. Обрати форму. Ухвалити рішення, за які потім відповідати. Те саме важке й цікаве, ті самі інтелектуальні задачі, заради яких ми сюди прийшли.

І ось тепер ми, software-інженери, нарешті можемо повернутися від кодингу до суті software engineering — до розв'язання задач. Від кодера назад до інженера.

Ось це і є справжній software engineering. Не боротьба з інструментами — розв'язання задач. Як у математиці. Як було від самого початку.

Тож ласкаво просимо назад. Кодер у нас тепер машина — а ми знову інженери.

І, чесно кажучи, жити знову стало цікавіше.

[1] Фредерік Брукс, «No Silver Bullet — Essence and Accident in Software Engineering» (1986). Сутнісна (essential) складність — це складність самої задачі; привнесена (accidental) — складність інструментів, якими ми її розв'язуємо. https://uk.wikipedia.org/wiki/Срібної_кулі_немає