

темпы роста производительности труда, например в обрабатывающей промышленности США, не превышали 30% за 10 лет. Таким образом, на основании данных табл. 17 и 18, которые приводились их авторами, чтобы подчеркнуть медленные темпы роста производительности труда программистов, может быть сделан прямо противоположный вывод о рекордных, невиданно высоких темпах роста их производительности.

Какой из двух противоположных по смыслу выводов, сделанных по одним и тем же статистическим данным, является верным?

Если ошибочность первого вывода (о медленных темпах роста производительности программистов) заключается в сопоставлении производственных характеристик людей и машин, то ошибочность второго (о рекордных темпах роста этого показателя) — в «реликтовом» способе измерения производительности труда программистов.

Возможно, в начале 50-х годов и существовала прямая зависимость между трудоемкостью программирования и длиной результирующего программного кода. Это позволяло, видимо, измерять производительность труда программистов (как, например, землекопов и лесорубов) «по валу»: в кубометрах (или тоннах) перфокарт или, наконец, чтобы исключить погрешность из-за разных сортов бумаги, непосредственно в строках программного кода.

Спустя 30 лет так много изменилось в программировании, что сопоставлять производительность труда программистов 50-х годов и 80-х оказалось нельзя: принципиально иным стал характер труда и соответствующие единицы его измерения. Если в 50-х годах решались в основном хорошо определенные задачи и соответственно главной производственной проблемой программистов было корректно перевести точные формальные спецификации в эффективный код программы, то в 80-х годах большая часть профессиональных программистов — это системные аналитики, которые, потребляя свыше 70% общих ресурсов, выделяемых на программирование, вообще не создают ни одной строки программного кода, а заняты лишь анализом приложений и постановкой задачи. Как в этих условиях можно оценивать темпы роста производительности труда программистов по скорости, с которой они создают программный код?

Из приводимых на рис. 24 данных о структуре распределения трудозатрат на программирование видно, что даже если бы завтра оказалось возможным создать, наконец, долгожданный «перпетуум мобиле» большой науки программи-