

Почему американская информатика кажется неизлечимой

Эдсгер Вибе Дейкстра

Обсуждая, как чувствует себя академическая информатика в США, мы должны уделять больше внимания компьютерной индустрии, чем в других странах, и должны мы это делать по двум причинам. Во-первых, в сравнении с другими странами, здесь нет барьера, отделяющего американский студенческий городок от окружающего общества, во-

вторых, большая часть, если не вся, мировой компьютерной индустрии сосредоточена в США. Следовательно, мнения и предрассудки, господствующие в американской компьютерной индустрии, для академической информатики здесь неоспоримы, невзирая на то, являются ли они движущими вперёд или парализующими.

Ради достижения стабильности предприятия идеал для менеджера — организация, максимально независимая от индивидуальных способностей служащих. Преобладание этого идеала — это хорошо документированный международный феномен. Этот идеал возник в индустрии высоких технологий, в которой он чрезвычайно неуместен и отталкивает от сотрудничества промышленников с учёными, особенно с блестящими и оригинальными учёными. Американская ситуация

усугубляется общим неверием в её образовательную систему и глубоко укоренившимся недоверием к интеллектуалам.

Кстати. Насколько типично американским является это недоверие, хорошо иллюстрируется фактом, что американское слово «яйцеголовый» не переводится на голландский язык. Я поискал его в моём «Websters New Collegiate Dictionary» (1973), выдержка из которого относится к непереводаемому слову: яйцеголовый сущ.: интеллектual, далёкий от жизни учёный <с точки зрения практичных людей, которые свысока относятся к планам и мечтам «яйцеголовых» W.L.Miller>.

По голландским меркам, цитата из Миллера является однозначной дискредитацией «практичных людей». (Конец Кстати).

В соответствии с этим, повышенное внимание со стороны индустрии оказывает силь-

ное давление на университеты — не поощрять увлечения вроде академического образования, а углубляться в профессиональную подготовку того или иного рода.

Но в случае информатики индустриальные предрассудки влияют не только на преподавание, исследования страдают от них ничуть не меньше. Например, я припоминаю одного так называемого «кандидата в преподаватели», который докладывал о своей диссертации на звание доктора философии — так было принято, — темой которой являлась система «распараллеливания» весьма скромного класса Fortran-программ. Докладчик не претендовал на какое-либо новое видение предмета или на то, что вы могли бы научиться чему-то интересному, изучая его тезисы. Единственное обоснование его работы заключалось в её конечном продукте, а именно —

программном обеспечении для распараллеливания Fortran-программ. И это программное обеспечение было весьма важно, поскольку (i) имеются тысячи Fortran-программ и (ii) система является полностью автоматической — абсолютное требование для её принятия промышленностью. (Я не выдумал всё это!)

Я думал, что основной критерий, по которому должны оцениваться наши академические исследования, — это насколько они улучшают учебные материалы, но этот бедняга принял в качестве критерия качества «применимость в промышленности», и в результате его творение работало в таких узких рамках условий, что его основным достоинством стала бездумная лёгкость применения.

Есть и другие примеры того, насколько исследования страдают от давления промышленности.

Быть лучшим программистом — означает быть способным разрабатывать более эффективные и достоверные программы и знать, как делать это эффективно. Это относится к экономному использованию ячеек памяти или машинных циклов, а также к избеганию сложностей, которые увеличивают количество рассуждений, необходимых для удержания строгого интеллектуального контроля над разработкой. То, что, по моему мнению, для этого нужно, — это совершенствование математических навыков, при этом я употребляю слово «математика» в смысле «искусство и наука эффективных рассуждений». В действительности задачи разработки высококачественных программ и построения высококачественных доказательств весьма сходны, настолько сходны, что я уже не могу их различать: я не вижу осмысленных разли-

чий между методологией программирования и математической методологией в общих чертах. Короче говоря, повсеместное вторжение компьютеров сделало способность к применению математического метода важнее, чем когда-либо.

По жестокой шутке истории, впрочем, американское общество выбрало именно двадцатое столетие для того, чтобы становиться всё более и более нематематическим (кстати, явление, рассмотренное Моррисом Клайном и вызвавшее у него глубочайшее сожаление). Мы достигли парадоксального состояния, когда из всех так называемых «развитых наций» именно США сильнее всех зависят от программируемых компьютеров и хуже всех интеллектуально оснащены в данном направлении. Предположение о том, что проблема программирования может быть вылечена ма-

тематическими средствами, мгновенно отвергается как совершенно нереалистичное.

В результате Разработка Программ ограждена от возможности стать поддисциплиной Информатики. Имеет место значительная озабоченность корректностью, но она почти полностью направлена на верификацию программ *a posteriori*, потому что опять же это легче укладывается в мечту о полной автоматизации. Но, разумеется, многие рассматривают верификацию *a posteriori* как установку телеги впереди лошади, потому что процедура «сначала программирование, потом проверка» поднимает насущный вопрос, откуда берётся программа, подвергающаяся проверке. Если же она выведена, то верификация сводится к простой проверке вывода. А между тем методология программирования, переимено-

ванная в «программную инженерию», стала настоящим раем для гуру и знахарей.

Будучи лишённой того, что обычно рассматривается как основное направление компьютерной науки, американская информатика превратилась в большого неудачника. И мы не вправе винить в этом университеты, поскольку, когда промышленность, наиболее нуждающаяся в их научной помощи, неспособна понять, что это высокотехнологичный бизнес, лучшие университеты оказываются бессильны. Университетам следует быть более просвещёнными, чем их окружение, и они способны на это, но не очень (точнее, стараются этого не показывать). В нынешней политической ситуации непохоже, чтобы что-то улучшилось вскоре; в ближайшем будущем нам придётся жить среди предрассудков, что программирование — это «настолько просто,

что им могут заниматься даже члены республиканской партии».

Остин, 26 августа 1995

Prof. dr. Edsger W. Dijkstra

Department of Computer Sciences

The University of Texas at Austin

Austin, TX 78712-1188

USA