

Ремесленник или учёный?

Эдсгер Вибе Дейкстра

Несколько обтекаемый заголовок моей речи относится, как вы, конечно же, догадались, к профессии программиста. Однако, по всей вероятности, вы не догадываетесь, что я хочу использовать слова «ремесленник» и «учёный» в очень узком значении: они выбраны, чтобы охарактеризовать результаты двух крайних типов образования, и эта речь будет посвящена (краткому) обсуждению их ро-

ли в образовании программистов, в обучении программированию. Для передачи знания и навыков оба этих метода использовались на протяжении многих столетий.

Будущий ремесленник приходит к мастеру на семь лет ученичества, он работает как подмастерье под его руководством и надзором, постепенно постигая навыки ремесла, — до тех пор, пока сам не станет мастером. Мастера обычно образуют цехи (гильдии), и члены гильдии держат секреты ремесла как хорошо охраняемую тайну: не записывать свои рецепты — одно из правил их профессиональной этики. Обратите внимание, наконец, что многие старые ремёсла были утеряны, поскольку передавались изустно от поколения к поколению.

Будущий учёный изучает свою науку как студент у преподавателя, который (в отли-

чие от мастера, передающего знание ученику на личном примере) формулирует знания и описывать навыки настолько чётко, насколько это возможно. Именно такая методика преобладает в университетах. Бурное развитие университетов произошло именно тогда, когда повсеместно прижилось книгопечатание, и с тех пор каждый университет воспринимал свою библиотеку как самую большую ценность: библиотека была воплощением образа жизни. Учёные придают огромное значение свободному обмену знаниями и идеями, отсюда следует общая открытость науки как одно из правил профессиональной этики.

На сегодняшний день применяются оба метода: физики, например, являются главным образом учёными, однако врачи больше похожи на членов гильдии. Математики — где-то между ними: математические резуль-

таты изданы и преподаются открыто, но студентов очень редко обучают тому, как делать математические открытия, а публикация не только результатов, но и ведущих к ним эвристик расценивается многими как «ненаучная» и потому считается дурным тоном: зачастую редакционная цензура старается не допустить публикацию эвристик.

Я сделал для вас набросок двух крайних точек на шкале образовательных методов, но это только вступление: моя настоящая тема — «куда на этой шкале мы должны поместить обучение программированию?». Как я понял на своём грустном опыте, тема эта весьма рискованна для открытой дискуссии. Я часто обсуждал её с людьми, имеющими непосредственно отношение к профессиональному программированию, и предмет этот всегда вызывал у них сильные эмоциональные реак-

ции. Давайте попробуем понять эти реакции, и тогда мы сможем попытаться справиться с ними.

Дабы сделать неявное знание явным и описать навыки так, чтобы их можно было передать, необходимо сформулировать некие концепции. Однако мы должны понимать, что переход ремесла в науку и публикация секретного знания всегда вызывает у членов гильдии ощущение угрозы. Для многих виртуозных программистов начала шестидесятих годов научное развитие в прошлом десятилетии было в высшей степени неприятно. Эти люди напоминают средневекового живописца, который создавал шедевры, поскольку опыт позволял ему хорошо соблюдать пропорции, но неожиданно оказался обойдѐнным молодѐжью, учениками Альбрехта Дюрера и др., которым преподали основы математики,

позволяющие гарантированно превзойти интуитивные навыки старого живописца. С ностальгией он оглядывается на добрые старые времена, когда его опыт и ощущения делали его непревзойдённым мастером.

Мы должны понять, что в программистской сфере это сражение ещё не закончено. Будучи в одной из европейских стран, неважно в какой, я изучил там предложение об организации преподавания вычислительных наук в университетах. Большинство его авторов — профессора вычислительной науки (computer science) в этой стране — должны быть охарактеризованы как «ремесленники». В результате их предложение имело явную антиинтеллектуальную окраску: в нём делался упор на то, что студентам преподавать, как решать задачи из «реального мира», и дальше следовал вывод, что учебный план

должен касаться абстрактных тем настолько мало, насколько возможно. Вы, несомненно, узнаете в этом тезисе точку зрения ремесленника в чистом виде.

На другом краю мы имеем чистого учёного: если мы дадим ему власть принимать решения, результат тоже будет плачевным. Он будет видеть свою дисциплину — будь это теория автоматов, рекурсивная теория функций, теория формальных грамматик, логика или теория очередей — с исключительной ясностью, которую мы имеем право ожидать от современного учёного, но одну вещь ему принять почти невозможно: что его красивый формальный аппарат не исчерпывает содержания обучения. Со времён Тьюринга мы имеем полную теорию того, как управлять битами. Разве она не делает лишними все вычисления? К чему интересоваться проблема-

ми «реального мира»? Теория доказала, что все эти проблемы могут быть решены, так зачем беспокоиться о фактическом решении? Вы, несомненно, узнаете в этих тезисах точку зрения настоящего учёного в чистом виде.

Следовательно, обе крайности плохи, и мы должны как-то смешать их. Но теперь мы должны быть внимательными, поскольку «смешивание» — не одномерный процесс. Оно не сводится к «столько-то процентов ремесленника и столько-то процентов учёного», а должно быть описано как «вот это от ремесленника, а это от учёного».

Увы, я вынужден описать вам всю гибельность картины смешивания технологий Ремесленника с подходом Учёного. Ремесленник не имеет никакого сознательного, формального владения предметом, он только «знает», как использовать инструментальные

средства. Если объединить это с подходом учёных к созданию явного знания, ремесленник опишет то, что он знает (то есть свои инструментальные средства), вместо описания, как надо их использовать! Если он — живописец, он сообщит ученикам всё, что знает о кисточках, а также всё, что ему известно о колебаниях цен на холст. Если он — профессор вычислительной науки, он сообщит своим студентам всё, что знает о существующих языках программирования, машинах, операционных системах, пакетах прикладных программ, и все уловки, обнаруженные им при непосредственном программировании. Он не просто перечислит указанные в руководстве пробивки 17-го столбца первой перфокарты для установки приоритета вашего задания в очереди, но обязательно упомянет недокументированную пробивку в 17-м

столбце, которая обеспечит вашему заданию высший приоритет, а оплату начислит как за низший. Опять-таки, вы несомненно можете узнать знакомые черты в этом описании.

О гибельности такого смешивания нужно специально предупреждать, и не только для того, чтобы продемонстрировать точку зрения, согласно которой выбор пробивки в 17-м столбце перфокарты имеет такое же отношение к программированию, как выбор между карандашом и шариковой ручкой — к математике. Об этом нужно предупреждать, потому что такое смешивание не просто гибельно, оно до сих пор весьма уважаемо! С одной стороны, вы находитесь в сфере проблем реального мира, и никто не может обвинить вас в чрезмерном навязывании студентам абстракций. С другой стороны, вы настолько конкретны, насколько это возможно, а всё, что

вы сообщаете, является бесспорной истиной. Когда кто-то имеет наглость указывать вам, что знания, которые вы передаёте, являются в лучшем случае малополезными и при этом совершенно произвольными, а кое в чём даже запутывающими, вы пожимаете плечами и говорите: «Но это — лучшее из имеющегося, не так ли?» Как будто это оправдывает подобный стиль обучения...

Всё же я боюсь, что подобная форма обучения вычислительным наукам преобладает. Ведь как ещё мы можем объяснить часто высказываемое мнение, что «период полужизни» специалиста по вычислительным системам приблизительно равен пяти годам? О чём ещё говорит это высказывание, как не о том, что ему преподали хлам?

Немного зная человеческую природу, я понимаю, что после вышеприведённой тирады

против неправильного смешивания все вы теперь ожидаете, что я скажу о своей симпатии к «обратному» смешиванию. Вы не ошиблись: преподаватели программирования должны пробовать смешать технологию учёного с рецептами ремесленника.

Основа технологии учёного — сделать явными как можно больше аспектов своего предмета. При этом обучение программированию включает обучение фактам — о системах, машинах, языках программирования и т.д. — и всё это очень легко сделать явным. Но неприятность состоит в том, что эти факты составляют всего лишь 10 процентов. То, что должно преподаваться в оставшихся 90 процентах, — это «как решать задачи», «как избежать непреодолимых трудностей», — короче, это обучение умению мыслить, не больше и не меньше. Явное обучение мышлению

— нетривиальная задача, но кто сказал, что обучение программированию проще? В нашей терминологии: чем более явно преподавать мышление, тем больше доля Учёного в будущем программисте.

Это, конечно, приводит нас к вопросу о том, как обучать мышлению. Чтобы сделать этот вопрос более реалистичным, мы несколько уточним его: знание, как преподавать мышление, не будет подразумевать, что каждый студент способен научиться этому. Это не должно нас смущать: в этом отношении «мышление» не отличается от любой другой темы, которую мы преподаём. Можно ли преподавать мышление? Реклама на последней странице обложки моего издания книги По́йа «Как решать задачу» звучит весьма обнадеживающе: «По́йа, как умелый преподаватель, показывает нам, как снять

шоры, мешающие думать, и ведёт нас к ясному и продуктивному образу мыслей».

Звучит отлично, однако это всего лишь реклама... Между тем другая большая работа Пойа по этой же тематике — «Математика и правдоподобные рассуждения» — была прохладно встречена математическим сообществом и оказала крайне незначительное влияние на обучение математике в университетах. Однако после некоторого размышления понимаешь, что прохладный приём книги Пойа математическим сообществом не означает неправоты автора. Напротив! Такое отношение может интерпретироваться и как реакция математической гильдии, которая, как и все другие гильдии чувствует угрозу раскрытия тайн ремесла. Впервые изданная тридцать лет назад, книга Пойа о процессе математических открытий была сочте-

на ересь, и таковой она остаётся в глазах многих математиков и сегодня. Прочитую книгу «Менеджмент и Макиавелли» Энтони Джея: «В корпоративных религиях, как и в остальных, еретик должен быть изгнан не из-за вероятности того, что он ошибается, а из-за возможности, что он прав». Другими словами, неприятие работ По́я говорит больше об интеллектуальной инерции математического истеблишмента, чем о качестве книг, так что на сей раз я предлагаю вам поверить рекламе!

Я рассматриваю книгу По́я «Как решать задачу» в качестве первого шага в нужном направлении. Он рассматривает эвристику как своего рода контрольный список стандартных вопросов, которые могут быть полезны, чтобы не упустить простого, хотя и неожиданного решения, если оно существует. Когда я прочитал книгу в первый раз, я был

несколько разочарован. Моё разочарование было прямым следствием того, что я уже был глубоко вовлечён в программирование: я чувствовал, что мои проблемы как программиста лежали в основном вне области, рассмотренной Пойа. Сначала я не хотел говорить так категорично, потому что те, кто заявляет об исключительности собственной сферы исследований, как правило, выглядят крайне смешными. Но после внимательного анализа я сделал вывод, что интеллектуальный вызов, представленный задачей программирования, на самом деле столь же беспрецедентен, как и высокоскоростной компьютер. Это вызвало в моём сознании сдвиг внимания от вопроса «как открыть неожиданное» к вопросу «как избежать непреодолимых трудностей», а затем к «как уменьшить объём запросов к количественно ограниченным ресурсам нашего

мышления».

Вы должны поверить мне на слово, что именно опыт сделал меня сторонником этого нового аспекта мышления. Вопрос «как избежать непреодолимых трудностей» действительно может быть предметом преподавания. Возможно, это заявление шокирует вас, поскольку вы знаете, что я обычно не впадаю в немотивированный оптимизм.

Однако следует помнить, что существуют некоторые пределы того уровня, на котором методы мышления можно преподавать явно, «в научной манере». Как сказано у По́йа, «первое правило открытия — иметь мозги и удачу. Второе правило открытия — напрячься и ждать, пока не придёт озарение. Следует также помнить, что некоторые стремления безнадёжны. Безошибочные правила, ведущие к решению всех возможных матема-

тических проблем, были бы ещё более желательны, чем философский камень, безуспешно разыскиваемый алхимиками. Такие правила работали бы как волшебство; однако волшебства не бывает. Находить неизменные правила, применимые ко всем видам проблем — старая мечта философов; но она никогда не станет больше чем мечтой». И это — в книге, где обучение размышлениям напоминает обучение ремеслу, где, как в добрые старые времена цехов и гильдий, Мастер творит чудеса и может создать Школу своим примером.

К тем из вас, кто преподаёт в университетах и колледжах, у меня есть только одна просьба: пожалуйста, не стыдитесь, что ваше обучение размышлениям «ненаучно»! Помните, что вся непонятая глубина человеческой мысли — всего лишь игра ума в процессе человеческого общения. Мы — несмотря на то,

что думают по этому поводу психологи, педагоги и т.п. — очень слабо понимаем, как происходит передача знаний, понимания и навыков. Весьма вероятно, что передача знаний в виде фактов — всегда всего лишь имитация, и что всё явное обучение в научной традиции не больше, чем сообщение студенту вербальных подпорок, которые просто-напросто помогают его памяти. Если это так, то всё «научное обучение» — то есть явные правила и ничего более — должно быть навсегда отброшено как бесплодное и непродуктивное занятие.

Я хотел бы закончить маленькой историей, приоткрывшей мне абсолютную тайну человеческого общения. Однажды я пошёл к фортепиано с намерением сыграть сонату Моцарта, но у клавиатуры внезапно передумал и вместо этого заиграл Шуберта. После

первых тактов моя удивлённая мать прервала меня словами «я думала, что ты собирался сыграть Моцарта!» В тот момент она читала книгу и всего лишь краешком глаза видела, что я иду к фортепиано. Однако выяснялось, что всякий раз, когда я шёл к инструменту, она уже знала, что именно я собирался играть! Как? Единственным объяснением является то, что она знала меня в течение семнадцати лет. С тех пор я думаю, что тщетно пробовать понять то, что происходит в классной комнате между тем, кто преподаёт, и тем, кто узнает новое. Следовательно, не иметь никакой модели этого процесса безопаснее, чем, имея такую модель, забывать о её погрешностях. Спасибо за внимание.