

Как нам сохранить и развить научноёмкое программирование

У всех на устах феноменальный успех Индии в аутсорсинге IT- услуг, объём которых составляет 15-20 млрд долларов в год. Не вдаваясь в подробности, перечислим основные причины их успеха:

- государственная поддержка (освобождение от налогов, развитие телекоммуникаций и образования);
- почти миллион индусов, получивших образование и опыт работы в развитых странах, которые вернулись и теперь работают в Индии, а оставшиеся на Западе всячески поддерживают своих сородичей;
- относительно хорошее знание английского языка (хотя далеко не 100% - ое, как многие думают);
- дешевизна рабочей силы.

Но не всё так страшно для нас в сравнении с Индией. Во-первых, в миллиардной стране число грамотных людей меньше числа грамотных в России. Во-вторых, среди 150,000 программистов, которые выпускаются индийскими университетами в год, не более 3-5 тысяч, по их собственным оценкам, получают действительно хорошее образование, ещё 10-15 тысяч получают более и менее нормальное образование, а остальные – просто никакие. Индийские власти это понимают (это их собственная градация на 3 сорта – 3 tiers), всячески борются за подъём уровня образования, но пока ситуация именно такая. Наконец, из 15-20 млрд долларов экспорта IT- услуг только 2-3 млрд связаны именно с программированием. Если 300 женщин отвечают на звонки в call-центре американской компании, расположенном в Индии, - это тоже IT- аутсорсинг.

Таким образом, бороться с Индией и другими странами за рынки сбыта можно, хотя и сложно. В чём состоят наши потенциальные преимущества? Прежде всего, в высоком уровне образования в нашей стране, креативности (созидательности) наших специалистов, высоком научном уровне наших университетов и академических институтов. Это признается и западными специалистами, например, на Gartner Outsourcing Summit 2006 вице-президент Cartner Ян Мариот, выступая с трибуны перед 1300 участниками, сказал: «Если у Вас есть трудная задача – несите её в Россию».

Наша задача – не растерять это преимущество, а, наоборот, всячески его развивать.

В этом докладе я попытаюсь рассказать о нашем опыте в этой области.

Начнём с 14 февраля 1991 года, когда было образовано малое государственное предприятие “Терком”. Чем мы владели в этот момент?

- 20-летний опыт создания трансляторов (Алгол 68, Ада, языки искусственного интеллекта) для различных платформ;
- 7 лет работы в промышленности (технология программирования, ПО телефонных станций);
- собственная ЭВМ “Самсон”, ориентированная на языки высокого уровня, доведённая до серийного производства.

Мы получили хороший заказ от Управления правительственной связи КГБ на разработку функционального программного обеспечения, технологии программирования и комплекса вычислительных средств для АТС “Фобос - К”. Всё было хорошо и интересно.

Но уже в 1992 году начался обвал в промышленности, сбой в финансировании.

Пришлось срочно искать зарубежных заказчиков. Такого опыта у нас не было, но уже тогда я понимал, что размениваться на мелкие простые задачи очень опасно. Голодных программистов в России было много, они хватались за любую работу за смешные деньги. Важно было сохранить коллектив, чтобы было возможно использовать наши заделы, наш опыт.

Нашим первым зарубежным заказчиком стала итальянская компания ITALTEL, которая поручила нам разработать плату и ПО устройства Set top - приставки к телевизору позволяющей выходить в Internet. Когда я года через 2 впервые оказался в Милане, они мне показали АТМ-коммутатор, в котором основным элементом была наша плата. Оказалось, что они боялись сообщить истинное назначение платы, чтобы информация не утекла к конкурентам. Нашим специалистам потребовалось много времени, чтобы объяснить итальянцам глупость такого подхода. Получившийся коммутатор был страшно избыточным и дорогим. В конце концов, они заказали нам устройство целиком, ПО для него, а также ПО управления сетью в целом. Под эти работы мы получили рабочие станции фирмы HP, специализированное ПО фирмы Mentor Graphics, различное диагностическое оборудование. Наши инженеры прошли обучение в Италии, один математик даже умудрился поучиться в школе НАТО.

За пару лет мы создали один из первых в мире АТМ-коммутаторов. Он серийно выпускался для сети PEAN (Pan Europe ATM Network). Насколько я знаю, до сих пор в России ни один коллектив не создал свой АТМ-коммутатор, все пользуются покупными, часто даже не умея пользоваться всеми предоставляемыми возможностями.

Для нас этот пример очень важен – с одной стороны, итальянцы нас научили программировать кристаллы на VHDL, а с другой стороны, инженеры, создавшие “Сомсон”, сравнительно легко придумали, как реализовать аппаратуру АТМ-коммутатора, а математики, имевшие опыт программирования АТС, справились с ПО коммутатора.

Трудности программирования встроенных систем реального времени заставили нас серьезно заняться технологией создания надёжного ПО, мы изучили рекомендации Международного Консультативного Комитета по телеграфии и телефонии по графическому проектированию и реализовали свои программные средства. С тех пор мы ими постоянно пользуемся и совершенствуем. Технология программирования стала основным направлением наших научных исследований.

Всего через несколько месяцев после получения итальянского проекта нам удалось получить первый американский контракт. Американцы по-своему решили проблему нехватки кадров – придумали язык Rules (правила), на котором могут писать люди после 2-3х месячных курсов. Опытный системный программист проектирует структуру приложения, производит декомпозицию до такого уровня, чтобы каждый элемент можно было описать на языке Rules на 1-2х страницах.

Как ни странно, этот подход оказался очень популярным, многие финансовые приложения были реализованы по этой технологии, получившей название HPS (High Productivity System). Однако американская реализация этой интересной бизнес-идеи была неудачной. Постепенно мы переписали большую часть HPS. Опять нам пригодился опыт трансляции, более того, во многих случаях пришлось решать и новые проблемы.

В 1997 году те же американцы поставили перед нами новую задачу – реинжиниринг. В мире накоплено много тысяч программ, написанных на устаревших языках типа Кобол, ПЛ1, Адабас Натурал и т.п., причём эти программы до сих пор работают и заменить их нечем.

Сопровождение стоит баснословных денег, поэтому нужны инструментальные средства, которые помогли бы провести реструктуризацию, удаление отмерших участков кода, выделить те участки кода, которые отвечают за определённые функции и т.д.

В идеале, хотелось бы перевести эти приложения на языки Джава и С#.

Реинжиниринг оказался очень трудной задачей, по этой теме было защищено несколько десятков дипломных работ, 3 кандидатские диссертации. Многие результаты мы сумели использовать и в других работах.

Постепенно мы приобрели определенную известность, научились правильно работать с западными заказчиками, восстановили связи с российскими военными, разработали и сопровождаем серийное производство городской АТС «Квант-Е-Сокол». Мы разработали АТС для ВМФ России (совместно со Связьморпроект»), коммутатор для выездных мероприятий Президента России (совместно с НПК «Красная заря»), новые компьютеры для старых АТС МТ20 и С32. Для западных заказчиков мы разрабатывали методы борьбы с некорректным использованием Интернет, занимались ПО для электронной торговли, работали по технологиям Agile-programming и Model Driven Architecture, выполняли интересные НИР для Intel, за свои деньги выполнили несколько наукоемких инвестиционных проектов.

Совсем недавно мы успешно завершили проект эхо-компенсатора по заказу крупной финской телекоммуникационной фирмы. Для этой работы мы привлекли профессора А.Барабанова из лаборатории теоретической кибернетики нашего мат-меха, который разработал новые интересные алгоритмы цифровой обработки сигналов, мы разработали специальную плату, ориентированную на задачи такого типа, и реализовали соответствующее ПО. Этот пример, в каком-то смысле, мой любимый, так как отражает суть нашего наукоемкого подхода. К нам лучше обращаться не с готовой спецификацией (компаний, работающих по готовым спецификациям, по всему миру великое множество), а с бизнес-идеей, изложенной на одной странице. Мы беремся уточнить постановку задачи, разработать эффективные алгоритмы, профессионально реализовать ПО, а если нужно, то и необходимую аппаратуру, т.е. пройти весь путь от математики до промышленного воплощения. Для нас вполне естественно приглашать ученых с других факультетов нашего Университета для совместных работ, мы успешно сотрудничали с физиками, химиками и даже филологами.

Я мог бы привести ещё полтора - два десятка примеров наших работ, но и приведённых достаточно, чтобы сделать некоторые выводы.

1. В условиях практически полного отсутствия государственного финансирования мы можем развивать программную инженерию за счёт заказчиков, особенно зарубежных. Надо только не браться за дурацкие работы (всех денег не заработаешь).
2. Зарабатывать деньги в университетской структуре, например на кафедре, очень сложно. Нужно создать аффилированное предприятие со всеми промышленными атрибутами: охрана, интернет, системные администраторы, маркетинг, финансовые службы и т.д. Без этого нет шансов привлечь хорошего заказчика. Никто не хочет рисковать.
3. Зарабатывание денег тесно связано с обучением студентов. Нужные лекции кафедра прочитать сумеет, но обеспечить каждого студента хорошей темой курсовой, а затем и дипломной работы и хорошим персональным научным руководителем очень трудно. Предприятие, привлекая студентов к работе, решает эту задачу.

4. Преподаватели информационных технологий, не работающие на производстве, - это не преподаватели.
5. Одно, даже большое предприятие, не сможет решить все проблемы кафедры, если студентов много (а потребность в IT-специалистах очень высока). На Западе эту проблему решают созданием Технопарка в непосредственной близости от Университета. Если в Технопарке работают 30-40 фирм, можно проводить конференции, заниматься более фундаментальными исследованиями, организовывать массовую подготовку студентов.
6. Разумеется, наш опыт нельзя считать универсальным, фундаментальные науки так не поддержишь, этим должно заниматься государство. Даже в нашей ситуации не всё так гладко. Например, мне надоело заниматься исключительно теми направлениями, по которым есть заказы. Хотелось бы самому выбирать темы. В принципе, и это возможно, если зарабатывать много денег, но здесь есть другие опасности.
7. Надо аккуратно, но твёрдо объяснить студентам, что опасно гнаться за высокой зарплатой на первых курсах университета. Без хорошей фундаментальной подготовки можно на всю жизнь застрять на маленькой зарплате. Мы стараемся брать на работу студентов, только начиная с 4 курса, до этого они участвуют в исследовательских проектах, организованных сотрудниками предприятия, а не преподавателями. Лучшим студентам мы платим дополнительную стипендию, но она существенно меньше зарплаты.
8. Отдельного разговора заслуживают наука и аспирантура. Не секрет, что в последние годы аспирантура служит только защитой от армии. Лишь единицы аспирантов действительно хотят заниматься научными исследованиями. Тут нужна жёсткость. Если аспирант не участвует в работе семинара кафедры, не сдаёт экзамены по кандидатскому минимуму, не пишет статей, он безжалостно изгоняется “за потерю связи с научным руководителем”. Даже в такой прикладной науке, как программная инженерия, необходимы исследования, которые не дают быстрой отдачи, например, разработка новых способов проектирования ПО, исследования по психологии работы в коллективе, разработка метрик сложности ПО и т.д. Я договорился с собственниками нашего предприятия и могу тратить 10% прибыли по своему усмотрению, еще кучу денег мы тратим на регулярные инвестиционные проекты (хорошо обоснованные, с жестко отслеживаемым бюджетом), так вот, только мои 10% я могу тратить на науку, цели инвестиционных проектов совсем иные. По-видимому, на исследования должны тратиться деньги различных грантов, но эти механизмы у нас в стране еще очень слабо развиты, особенно в части гласности и прозрачности распределения грантов.