

Существует ли дефицит программистов или размышления о путях развития российского образования

Толчком к написанию данного текста послужили:

1. Заметка Ольги Самофаловой "Программисты становятся дефицитом" на сайте ВЗГЛЯД.РУ
Источник: vz.ru
 2. Статья Алексея Барабанова "Open Source взглядом оптимиста" датированная 2006 годом, актуальность которой за прошедшие 6 лет только возросла.
Источник: barabanov.ru
Копия: soft4edu.org
-

Первый текст фрагментарен, так как состоит из цитат, насыщен цифрами (иногда противоречащими друг другу) и в целом ведет к выводу обозначенному в подзаголовке: "В России прогнозируется острая нехватка специалистов по информационным технологиям".

Бессмысленно полемизировать с автором и экспертами, которых она цитирует по конкретным цифрам, ясно что это экспертные оценки. Объективных данных, дающих полную картину, не существует, но все знают, что очень существенная часть кода написанного в России написана [фрилансерами](#), а самые сложные работы по IT-администрированию часто выполняются по схеме [аутстаффинга](#), обычно "в серую".

Только одна цитата:

"Российским IT-компаниям необходимо развивать систему наставничества. Компании-интеграторы должны учить квалифицированных сотрудников работе с молодыми специалистами, потому что готовых кадров на рынке просто нет" - сказал вице-президент по развитию и инновациям группы компаний ЛАНИТ Степан Масленников.

Подобные разговоры в среде практиков IT-отрасли ведутся уже лет двадцать, кое-что даже делается, вот только результаты обескураживают - положение все ухудшается.

В США (страна несомненный лидер IT-отрасли) университеты являются донорами идей. Лучший код придумывается и пишется там. Вендоры охотятся за выпускниками, ибо они гарантия успеха в конкурентной борьбе, а наш бизнес размышляет, как бы использовать эти места, где концентрируется активная молодежь (ВУЗы называются), для пополнения своих стареющих кадров, и не может придумать! Идея ввоза спецов из-за границы уже давно втихую реализуется, даже соответствующие поправки к закону о гастарбайтерах провели через Госдуму.

Открывать кафедры крупных предприятий в университетах, конечно полезно, но недостаточно и часто даже неэффективно. Чтобы ВУЗ выпускал специалистов востребованных рынком, ВУЗ должен располагать не единицами достойно оплачиваемых преподавателей, а сотнями. Тогда спектр дисциплин прослушанных студентом будет достаточно широк, а глубина усвоенных знаний и актуальность полученных навыков позволит сразу после выпуска претендовать на хорошую вакансию.

Вывод все тот же. Университету нужны деньги. Много денег. Чтобы не только тупо вбивать в головы студентов необходимые, но давно открытые истины, а вовлекать их в поиск новых, т.е. в Науку. Именно так с большой буквы.

Университет не должен мучительно изыскивать средства на оплату отопления или зарплату трех новых лаборантов, он должен искать таланты и финансировать каждую, иногда даже бредовую, идею, если это позволит студенту стать креативнее, а аспиранту открыть новую тему в науке.

Роль государства сегодня в России определяющая. Оно не может ограничиться только введением правил игры (неполноценных пока, понимаемых слишком узко), процессом необходимо управлять умно и тонко. Если Минобр ограничится только бесконечным усовершенствованием ЕГЭ, мы рискуем скоро получить на месте Университета десяток кафедр глобальных корпораций (а у них, понятно, совсем другие приоритеты, далекие от наших национальных сверхзадач).

Количество университетов необходимо резко сократить. Оставшиеся финансировать по нормам ведущих стран (или лучше) из казны и найти способ стимулировать меценатство бизнеса. При этом вмешиваться в оперативное управление ВУЗом нельзя разрешать никому: ни чиновнику, ни благотворителю. Постфактум контролируйте, воров сажайте, но не более - самим же лучше будет - специалистов высшего класса получит и гос орган, и корпорация, и стартап.

Рассмотрим второй из анонсированных текстов.

Как будто бы тема совсем не связана с первой, однако это не так. В проблематике свободно софта, как в капле воды отражаются все аспекты более общей темы российского образования.

Алексей Барабанов в своей статье дал полный и точный абрис Open Source Software (OSS) в современном мире и рассмотрел его перспективы в России. Парадоксален вывод автора:

Для многих противостояние OSS и проприетарного ПО сосредоточено в соревновании Linux vs Windows. Это очень примитивно, но почти верно. Ошибкой является стереотипное приравнивание MS Windows старому и капиталистическому, а Linux новому и коммунистическому. Отсюда и напрасное ожидание, что Linux и все продукты, произведенные OSS, окажутся дешевле проприетарных. А происходит ровно наоборот. Именно Linux получается дороже в эксплуатационном цикле для корпоративных потребителей, чем MS Windows. Да, согласен, потому то я и работаю в среде Linux, что это приносит больше прибыли при меньших усилиях. Linux и другие OSS являются следующей за проприетарными моделями формой развития ИТ отрасли. И соответственно они несут большую эксплуатацию разработчикам, дороже для потребителей и выгоднее для инвесторов. Но не надо ждать, что движение OSS будет добиваться успеха повсеместно. С появлением коммунистических и развитых демократических стран на планете Земля история многих феодальных и даже рабовладельческих государств вовсе не завершилась, путем добровольного присоединения к СССР. Но также как неизбежен прогресс в общественных отношениях, так и распространение OSS уже не остановить. Потому что он усилиями таких компаний, как IBM и Novell превращается в инструмент неоглобализма.

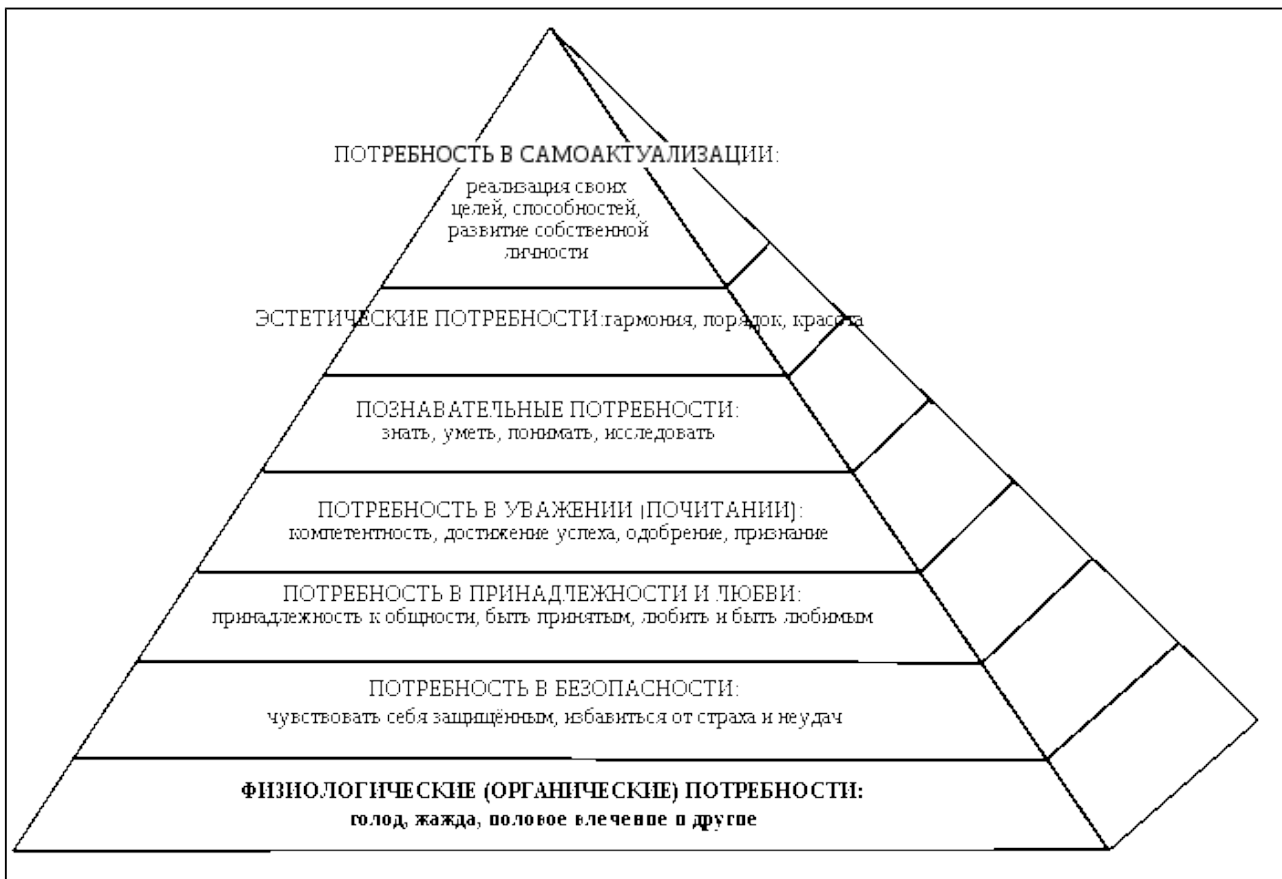
И теперь можно ответить на вопрос, угрожает ли что-нибудь OSS. Нет, не угрожает! Напротив, это сам OSS угрожает оригинальным коммерческим проектам. Чужой, импортированный OSS это быстрая и сладкая смерть национальной ИТ индустрии той

страны, которая не сможет противопоставить аналогичный свой OSS. Зачем вывозить `мозги`? Можно подарить `бесплатный` OSS и тем привлечь чужих специалистов к обслуживанию наукоемкой индустрии `доброго дяди` взамен отечественной.

Если принять эти два абзаца за постулат, окажется, что OSS не "капля воды", а цистерна сильнейшего катализатора не только для нашего образования, но для любой современной экономики.

Есть русский синоним для OSS - свободное программное обеспечение или СПО, далее будем употреблять его. (Специалисты знают, что эти термины не полностью взаимозаменяемы, есть важные нюансы, но здесь будем считать, что в первом приближении это так).

[Пирамиду Маслоу](#) упоминает Барабанов в своей статье, как один из опорных пунктов своего вывода, приведенного выше.



Барабанов утверждает, что только достигнув вершины пирамиды, человек способен на свободное творчество и на свободную жертвенность (т.е. в контексте СПО - может позволить себе бесплатно, ради удовольствия, трудиться или просто давать - не инвестировать! - деньги на разработку софта, который потом станет общественным достоянием).

Серьезные ученые, прежде всего сам [Abraham Maslow](#), не разделяют столь примитивную трактовку.

Вот пример такой критики:

Теория иерархии потребностей, несмотря на свою популярность, не подтверждается и имеет низкую валидность (Hall and Nougaim, 1968; Lawler and Suttle, 1972)[3]. Когда Холл и Нугейм (Hall и Nougaim) проводили своё исследование, Маслоу написал им письмо, в котором отметил, что важно учитывать удовлетворение потребностей в зависимости от возрастной группы испытуемых. `Счастливики` с точки зрения Маслоу удовлетворяют потребности в безопасности и физиологии в детстве, потребности в принадлежности и любви - в подростковом возрасте и т. д. Потребность же в самоактуализации удовлетворяется к годам 50-ти у `счастливи́ков`. Именно поэтому нужно учитывать возрастную структуру.

Главная проблема при проверке теории иерархии - как измерять потребности. Вторая проблема теории связана с разбиением потребностей в иерархии, их последовательность. Сам Маслоу указывал, что порядок в иерархии может меняться. Тем не менее, теория не может объяснить, почему некоторые потребности продолжают быть мотиваторами даже после того, как они были удовлетворены.

Поскольку Маслоу исследовал биографии только тех творческих личностей, которые, по его мнению, были успешными (`счастливи́ками`), то из исследуемых личностей выпал, например, Рихард Вагнер, великий композитор, лишенный практически всех личностных черт, ценимых Маслоу. Учёного интересовали необычайно активные и здоровые люди, такие как Элеонора Рузвельт, Авраам Линкольн и Альберт Эйнштейн. Это, конечно, накладывает неизбежные искажения на выводы Маслоу, так как из его исследований неясно было, как устроена `пирами́да потребностей` большинства людей. Также Маслоу не проводил эмпирических исследований

Источник: <http://ru.wikipedia.org>

Здесь мы не станем дальше углубляться в критику теории иерархии потребностей, а перейдем собственно к темам заявленным в заголовке.

В основном мы будем опираться на опыт работы в МГИМО, однако нам известно, что в других университетах фактическое положение дел либо близко к нашему, либо существенно хуже.

Программист - это ключевая профессия 21-го века, так же как рабочий был главной движущей силой в 20-м, а крестьянин в предыдущее тысячелетие. (Понятно, что наименования эти обобщающие. Крестьянский труд состоит из полеводства и животноводства. Рабочие это: и машинисты паровозов, и фрезеровщики, и сборщики космических аппаратов. Программистами по традиции называют всех, кто прикладывает руки к компьютерам: системные администраторы поддерживающие суперкомпьютеры и энкейщики, инженеры технической поддержки и собственно кодировщики, все они немножко "программисты", так же как комбайнеры и доярки - "немножко" крестьянине.)

Айтишники, компьютерщики - такие же вульгаризмы, как программист, в смысле работник IT-сферы. В этой индустрии давно произошло глубокое расслоение некогда единой профессии, теперь между web-дизайнером и web-программистом такая пропасть, что для их совместной работы необходим переводчик - руководитель web-проекта, который имеет некоторое представление о той и другой профессии, но сам может ими и не владеть в полной мере. (Американцы - наши антиподы - они не только не продвигают в руководство проектами бывших инженеров, но прямо считают опыт работы руками вредным для руководителя.)

Сами работники ИТ-сферы, различают программистов и кодировщиков. Первые обладают достаточной эрудицией, чтобы освоить предметную область, в которой они в данный момент работают, могут самостоятельно сформулировать требования к информационной системе и закодировать ее. Кодировщик может только последнее, более того он часто делает это формально, без фантазии и блеска (в чем особенно русским программистам нет равных в мире: компиляторы пишутся в Санкт-Петербурге, продукты фирмы Майкрософт - в Индии).

Кроме того программисты различаются по уровню близости к конечному пользователю (верхний уровень) или к железу на котором будет работать программа (нижний уровень).



Считается, что на нижнем уровне работают более квалифицированные узкие специалисты - интроверты, а на верхнем более широко образованные экстраверты. На рисунке явно выделены три слоя программных систем, ограниченные снизу слоем железа, а сверху слоем пользователей. Конечно это сильно упрощенная схема, на самом деле, каждый слой не монолитен, а также имеет свою, часто тоже слоистую, структуру. Толщина слоев показывает относительную трудоемкость.

Учитывая вышеизложенные замечания, уточним и конкретизируем вопросы:

1. Достаточно ли в МГИМО программистов?
2. Если нет, то каких именно не хватает?
3. Насколько критична эта проблема?
4. Как ее решать?

В штатном расписании их 2 или 3. Еще 2-3 сотрудника серьезно занимаются программированием помимо своих прямых обязанностей. Есть небольшая группа студентов 3-5 человек готовых к волонтерской работе не только морально, но и профессионально. Это все.

Возможно нам и не нужно больше? Мы не Бауманка, где каждый первый преподаватель - инженер, а каждый второй студент - кодер. Может быть нам дешевле станет вообще отказаться от собственных ИТ-специалистов, расформировать ДИТ и отдать все на аутсорсинг, как охрану, общепит или уборку помещений?

Хорошо бы еще закрыть пару непрофильных для МГИМО кафедр. Правда при этом мы рискуем потерять по формальным признакам статус Университета, а по неформальным - наши выпускники вынуждены будут добирать знания необходимые современному экономисту, юристу, дипломату, журналисту вне стен alma mater.

Современный университет не может себе позволить не только стоять на месте, но даже бежать вполсилы. Необходимо все время искать чего мы недодаем студенту и догонять, догонять... Хорошо бы, если по гамбургскому счету, еще и вперед вырваться иногда.

В отличие от железа, которое требует периодического обслуживания и/или замены, информационные системы должны развиваться постоянно. Перманентная модернизация - это нормальное состояние современного софта, поэтому без программистов обойтись не удастся. Совокупность прикладного софта более трудоемка в проектировании и поддержке, чем другие типы софта, а тиражи каждого из продуктов существенно меньше - это ножницы, которые нужно осмыслить.

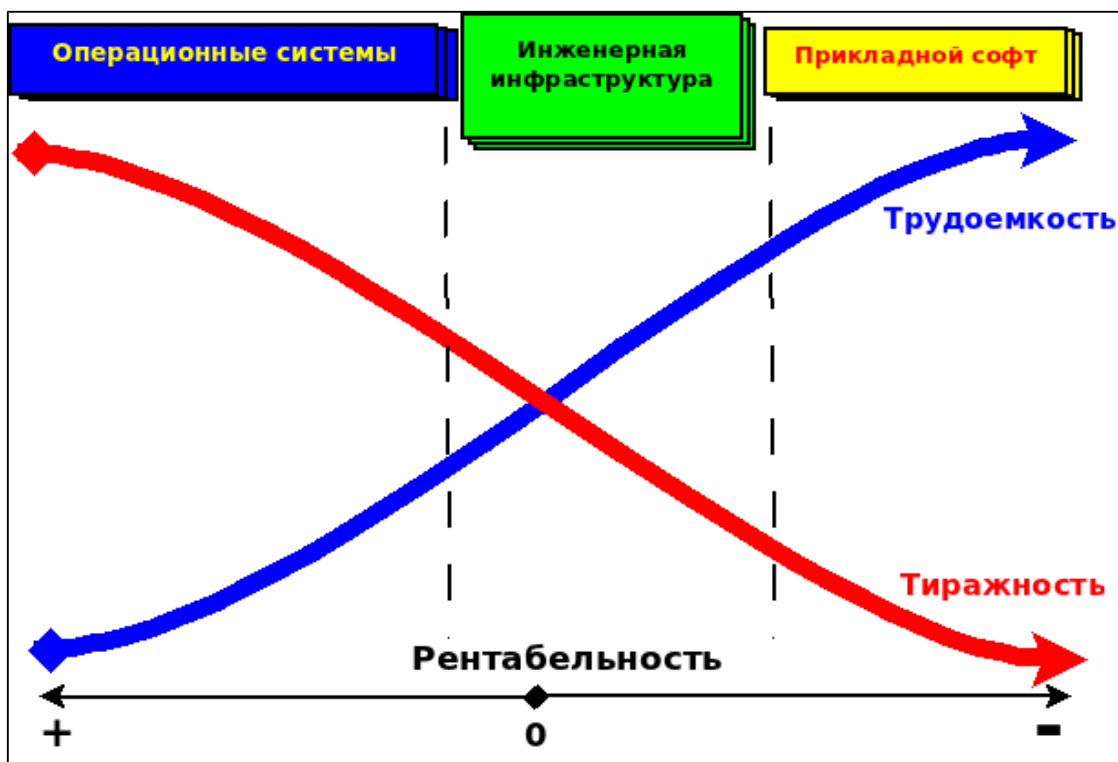


Схема намеренно гипертрофированна, это не количественный график, а качественная иллюстрация проблемы.

Теперь понятен корень проблемы: серьезный прикладной софт по сложности не уступает системному, при том что последний распространяется огромными тиражами и значит

рентабелен. Но ядро операционной системы (действительно сложная и важная штука) непосредственной пользы конечному пользователю не приносит (кому нужно базовое шасси от автомобиля? - всем нужен автомобиль, а точнее удобное средство решения транспортной проблемы), поэтому реальному человеку нужен именно прикладной софт и его нужно большое разнообразие (аналог - автомобили спецназначения: спектр от пожарной до снегоуборочной - количество каждой разновидности меньше легковых на порядки).

Можно ли найти на рынке коммерческую фирму, которая возьмет на себя наши проблемы с прикладным софтом? Можно - 1С несомненно успешный пример. Однако чаще крупные системные интеграторы легко выигрывают конкурсы на разработку и даже иногда честно отработывают оплаченный заказ, однако сразу после подписания актов, они переключаются на следующего клиента, а мы остаемся один на один со своими проблемами (либо вынуждены платить высокую ренту "за глупость", т.е. за так называемую техническую поддержку). Собственно так устроен корпоративный капитализм. Жаловаться некуда.

Каждый университет пытается решать типовые для всех прикладные задачи в одиночку, и при этом каждый, кроме множества мелких технических граблей, наступает на огромные непреодолимые экономические грабли - денег на действительно качественный программный продукт все равно не хватает и на свет будет произведен очередной уродец. Вместо того, чтобы в складчину (как когда-то бедные колхозы создавали МТС) создать общий репозиторий продуктов и совместно их развивать и поддерживать, каждый в отдельности практически впустую тратит свои скудные средства.

СПО, как метод производства кода и как метод его распространения/использования/потребления является следующей ступенью развития способов производства.

Фактически похожая парадигма существовала всегда, просто теперь она становится существенным и необходимым фактором мировой экономики. Художник, который рисовал на стене пещеры сцены охоты и тем воодушевлял племя для следующей, ничего не требовал себе оплату труда - вождь давал ему еду тоже не потому, что осмыслил пользу картины и его автора, а просто так. Бескорыстны были оба. Поэты и изобретатели порождали свои творения потому, что не могли иначе. У одного процента популяции - любой! - есть этот ген. Творческий продукт выдается на гора при любых обстоятельствах: ни голод, ни чума, ни война не могут остановить этого - вот почему не правы Барабанов и Маслоу. И здесь нет ни грана романтики - чистый Дарвин.

В другом, как нам представляется, Барабанов прав: страна не вкладывающая в СПО существенные деньги, а только потребляющая чужое СПО, рискует потерять отечественную индустрию программного обеспечения, так же, как потребляющие чужие автомобили, мясо и фильмы, отвыкают их производить на конкурентном уровне и отстают навсегда. Страна же состоит из людей, предприятий, университетов. Отставание ведущих вузов страны, к которым МГИМО несомненно относится, критично вдвойне, втройне, ибо построить новый дом, университет, космодром на несколько порядков трудней и дороже, чем поддерживать в добротном состоянии то, во что уже вложен огромный труд прошлых поколений.

МГИМО - гуманитарный университет и странно было бы нам писать свою операционную систему, это пока удастся только Беркли, значит системных программистов нам содержать не нужно. А вот свои прикладные задачи мы вполне можем решать (конечно существенно лучше

это делать в кооперации с другими). В мире существует множество СПО-проектов, ничто не мешает нам присоединиться к тем, которые нам сразу же принесут большую пользу.

Вот три из них:

RunaWFE - это свободная система управления бизнес-процессами и административными регламентами. Фактически - система позволяет построить документооборот любой сложности, быстро, своими силами или при небольшой поддержке авторов.

Moodle - система управления обучением или виртуальная обучающая среда. Она кстати у нас стоит уже лет 6 на запасном пути.

Drupal (друпал) - система управления содержимым и [фреймворк](#) для web-проектов. Наш портал может быть портирован на друпал небольшими силами в течении 1 года, причем первые разделы, например новостные, могут быть запущены за 3-5 месяцев (потребуется консультации внешних специалистов).

Выводы:

- Любому университету нужны программисты.
- Разрабатывать (участвовать) своими силами/средствами имеет смысл только СПО.
- Естественная для СПО кооперация университетов (или их программистов) во время работы над общими проектами может дать очень большой синергетический эффект.

В данной статье мы немного углубились в узкую область прикладного программирования для университетов. Очевидно, что она является только небольшой частью айсберга проблем российского образования. Однако, на наш взгляд, данная проблема относится к разряду краеугольных, так как наше время - это время высоких технологий и тот кто не включится в процесс их создания и освоения останется на обочине. Неважно технический или гуманитарный продукт ты производишь - без программистов его не сделать ни современным, ни качественным, ни востребованным.

А. Немченко
<http://nemchenko.ru>
an2k@mail.ru