

## Козволюционная модель разработки программного обеспечения для обучения в контексте исследования механизмов самоорганизации системы образования

Владимир Вихрев

Институт проблем информатики РАН, Россия

[VVikhrev@ipiran.ru](mailto:VVikhrev@ipiran.ru)

### Введение

Система образования является сложной открытой социальной системой. Информатизация трансформирует ее в свехсложную социотехническую систему. Важным фактором существования таких систем является наличие развитых механизмов саморегулирования. История информатизации образования в нашей стране показывает, что достигнутые в ее ходе результаты в значительной степени связаны с активностью масс, самоорганизацией образовательного сообщества.

Ключевым звеном информатизации является развитие отрасли разработки программного обеспечения (ПО) для обучения. Именно компьютерная программа превращает универсальное средство ИКТ в инструмент обучения.

Между тем, отрасль разработки ПО для обучения почти не исследована с научной точки зрения, она в значительной степени представляет «вещь в себе». Показательно, что для обозначения ее продукции используется такой термин как электронный (цифровой) образовательный ресурс (ЭОР). Его нечеткость, размытость и явно «технократический» характер при одновременной популярности характеризуют бессилие назвать и систематизировать многообразие продуктов отрасли – хороший индикатор сложности отрасли разработки ПО для обучения как предмета исследования и объекта управления. Нечеткость представления об объекте управления ведет к неадекватным управленческим решениям. Это повышает важность активизации механизмов самоуправления.

### Программист vs учитель

Первое, что выделяет отрасль разработки программного обеспечения для обучения, есть особые отношения между программистами (разработчиками ПО) и учителями (потребителями ПО). Уже в начале информатизации образования многие школьные учителя активно подключились к разработке программ по своим предметам. Это породило комплекс проблем. «Программные средства учебного назначения могут создаваться следующими способами:

1. Программное средство создается программистом – низкое качество с точки зрения педагогики.
2. Программное средство создается педагогом – низкое качество с точки зрения программиста.
3. Программное средство создается при тесном контакте педагога с программистом (сюда же относится случай педагога-программиста) – качество высокое, но недостаточно большое число таких коллективов.
4. Программисты создают некоторый пакет прикладных программ, с помощью которого педагог-непрограммист в состоянии строить требуемые программные» [1, с. 6].

Другими словами, решение оппозиции «программист vs учитель» предполагалось осуществить в рамках парадигмы «автоформализации знаний» или «персональных вычислений» как этапа развития информационных технологий. «Основная задача персональных вычислений – формализации

профессиональных знаний – выполняется, как правило, полностью самостоятельно непрограммирующим профессионалом или при минимальной технической поддержке программиста, который в этом случае имеет возможность включиться в процесс формализации знаний только на инструментальном уровне, оставляя наиболее трудную для его понимания содержательную часть задачи специалисту в данной предметной области» [2, с. 99].

## **Разработчики: взгляд со стороны и изнутри**

Существует рынок программного обеспечения для обучения, наполняемый софтверными компаниями. К ним присоединяются издательства с электронными приложениями для своих учебников. Одновременно, в Интернете имеются в бесплатном доступе хранилища электронных образовательных ресурсов (по сути – программное обеспечение для обучения), где созданные профессионалами на государственные деньги коллекции соседствуют с коллекциями ресурсов социальных сообществ учителей, материалами школьных и учительских персональных сайтов. Получается, что деятельность команд-разработчиков гармонично соединяется с автоформализующими свои знания учителями?

Приглядимся внимательнее к программистам-разработчикам. Какие типы команд принимали участие в процессе информатизации?

1988 (по данным тематической выставки на ВДНХ [3]):

- специалисты отделов академических институтов, НПО, ВЦ;
- коллективы кафедр вузов (педагоги, инженеры, студенты);
- коллективы школ (учителя, школьники);
- независимые авторы.

2008 (по данным Единой коллекции образовательных ресурсов):

- специалисты медиаиздательств, книжных издательств;
- программисты коммерческих фирм;
- коллективы кафедр вузов;
- специалисты НИИ;
- независимые авторы.

За двадцать лет непрерывного усложнения техники и системного программного обеспечения, учителя и школьники выпали из числа разработчиков конкурентоспособного ПО.

## **Общая характеристика отрасли**

Если продолжить анализ состава разработчиков, исследуя, например, менее формализованные коллективы кафедр вузов, обнаружится, что за каждым коллективом стоит, как правило, один или несколько энтузиастов. От независимых авторов они отличаются лишь наличием команд и более сложными социальными отношениями. Коллективы 1,2,4 типологии авторов Единой коллекции в большинстве своем опираются на таких же энтузиастов.

Ситуация в отрасли разработки ПО для обучения может быть в целом охарактеризована так:

- Систематическое недофинансирование является не проблемой, а контекстом существования отрасли.
- Рынок ПО для обучения – не классический рынок законченных товаров, а, скорее, рынок высокотехнологичного оборудования. Самоокупаемость здесь невозможна без государственной поддержки.

В суровых условиях выживают лишь энтузиасты. Их энергия – главный ресурс развития отрасли. Чтобы энтузиазм не выгорел дотла, его следует поддержать.

Важно понять, какие проблемы надо решить в отрасли разработки программного обеспечения для обучения.

## **Модель жизненного цикла**

В качестве точки отсчета возьмем концепцию жизненного цикла программы.

Он включает следующие этапы:

- постановка задачи, формулировка требований;
- исследование проблемной области, разработка проекта;
- программирование, отладка;
- тестирование, доработка по результатам тестирования;
- сопровождение, доработка в процессе эксплуатации;
- формирование требований на переработку.

Главное внимание в литературе по разработке ПО [4, 5], сосредоточено на этапах 1-4. Но существует обширная литература и об этапе сопровождения [6]. Он признается настолько важным, что предлагается выделять специалиста, ответственного исключительно за сопровождение. Поскольку схема жизненного цикла в целом ориентирована на программу как товарный продукт, в ней нет процедур систематического взаимодействия пользователей и разработчиков.

## **Формулировка проблем**

Обучение является одним из сложнейших видов деятельности. Педагогическая наука пока не исследовала его до такой степени, чтобы относительно просто различать содержательную и дидактическую сторону этого процесса. Всякий раз соединение учебного материала со способом его подачи является искусством.

Разнообразие условий обучения и качеств участников этого процесса делает неразрешимой задачу однозначного методического описания. Как следствие – спектр проблем разработчика и пользователя-учителя.

Проблемы разработчика:

- точность постановки задачи.
- разработка методического обеспечения.
- привязка к разнообразным условиям применения.
- сопровождение, силы и возможности для решения этой задачи.

Главная проблема разработчика ПО – он не обладает ресурсами для полноценной реализации схемы жизненного цикла.

Проблемы учителя:

- овладение возможностями ПО при недостаточной помощи разработчика;
- адаптация возможностей ПО к конкретной ситуации при недостаточной квалификации или сложности ситуации;
- разработка педагогической технологии на базе имеющейся программы.

Главная проблема учителя – он не обладает ресурсами для самостоятельного углубления в программу.

В условиях хронического недофинансирования проблемы решаются за счет главного человеческого ресурса – времени жизни.

Наконец, при ограниченности финансовой поддержки неплохо бы компенсировать затраты сил энтузиаста хотя бы удовлетворением, что он получит, внедрив свой труд.

## **Снятие противоречия**

Как объяснить тот факт, что количество ЭОР, выложенных учителями в Интернете, возможно, превосходит количество ресурсов, созданных разработчиками? Несомненно, есть учителя, профессионально освоившие

ремесло разработчика. Они создают конкурентоспособные программы. Но их – единицы. Они не меняют положения вещей: учитель, включаясь в процесс информатизации, входит туда со своей особой ролью. Не подменять разработчика. Дополнять.

Задача учителя – спроектировать педагогический процесс, включив в него ИКТ. Учитель привлекает некоторое ПО как основу процесса. Лишь теоретически существует ПО, точно вписывающееся в «ландшафт» урока. Необходима привязка к местности, адаптация. Она может быть следующих типов:

- параметрическая, настройкой параметров;
- модульная, отбором используемых блоков;
- конструктивная, разработкой недостающих блоков.

Множество созданных учителями ЭОР появились из-за вынужденной необходимости конструктивной адаптации ПО.

Другая часть учительских ЭОР является электронными версиями методики проведения урока. Они представляют особую ценность как артефакты накопленного социального опыта. Каждую программу в информационно-образовательном пространстве можно представить окруженной «облаком» таких учительских ресурсов. Именно это «облако» с программной системой в центре является истинным результатом коллективного труда разработчиков и учителей.

### **Коллективный разум**

Описанная ситуация идеально ложится на концепцию коллективного интеллекта [7, 8]. Разработчик ПО и учителя, применяющие в своей практике его разработку, образуют социальную группу с четким разделением социальных ролей и хорошо понятными ролевыми функциями. Каждый из участников действует автономно, но у них есть интерес объединиться для интеллектуального исследования общей ситуации. Коммуникация позволит им выявлять проблемы, находить пути их решения, т. е. реализовать основную функцию интеллекта. Возникает коллективный разум.

### **Система поддержки коэволюционной модели**

Все стороны, участвующие в коммуникации, заинтересованы в развитии на основе этой коммуникации. Возникают предпосылки для совместной эволюции – коэволюции.

Технической базой совместной работы выступает сайт разработчика, спроектированный на базе технологий Web 2.0. Здесь размещается ПО и методические наработки учителей, здесь идет обсуждение. Для разработчика – это реализация функций этапа сопровождения и этапа проектирования развития системы. Для учителя «копилка» методического опыта. Администратором и модератором выступает энтузиаст разработчик.

### **Вывод**

Коэволюционная модель разработки ПО для обучения реализует принципы коллективного интеллекта и базируется на сервисах Web 2.0. Ее применение повышает уровень взаимодействия участвующих в информатизации образования сторон, разработчиков ПО и учителей-пользователей ЭОР. Повышается эффективность принимаемых при разработке ПО решений, растет эффективность применения ЭОР учителями, увеличивается суммарный эффект информатизации. Реализация модели опирается на силы заинтересованных в общении участников и в общем случае не требует привлечения значительных дополнительных ресурсов.

Это позволяет заключить, что коэволюционная модель разработки ПО для обучения может считаться эффективным механизмом самоорганизации для системы образования.

При широком применении коэволюционной модели повышается прозрачность системы образования в таком важном сегменте ее информатизации как разработка и использование ЭОР. Интенсивность и глубина коммуникаций вокруг ресурса разработчика становятся индикатором востребованности его разработки. Это может служить дополнительным критерием при подготовке программ финансирования отрасли.

## Ссылки

1. Федосеев А. А., Христочевский С. А. Об инструментальных средствах для разработки ППС // Разработка и применение программных средств ПЭВМ в учебном процессе: Материалы IV Всесоюзного семинара. – М.: ИПИАН, 1988. 384 с.
2. Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации. – М.: Наука, 1985. 240 с.
3. Компьютеризация образования (средства, методы, передовой опыт): Тематическая выставка. – М.: ВДНХ, 1988. 109 с.
4. Боэм Б. У. Инженерное проектирование программного обеспечения / Пер. с англ. – М.: Радио и связь, 1985. 512 с.
5. Брауде Э. Дж. Технология разработки программного обеспечения / Пер. с англ. – СПб.: Питер, 2004. 655 с.
6. Гласс Р., Нуазо Р. Сопровождение программного обеспечения / Пер. с англ. – М.: Мир, 1983. 156 с.
7. Tovey M. (ed.) Collective Intelligence: Creating a Prosperous World at Peace. – Oakton: Earth Intelligence Network, 2008. 612 p.
8. Шуровьески Дж. Мудрость толпы. Почему вместе мы умнее, чем поодиночке, и как коллективный разум влияет на бизнес, экономику, общество и государство / Пер. с англ. – М: Вильямс, 2007. 304 с.