

ОСНОВНЫЕ СОВРЕМЕННЫЕ МОДЕЛИ РАЗРАБОТКИ ПО НА ОСНОВЕ СПО

ГРИГОРЬЕВА А. И., КОЛОДИН М. Ю.

УДК 004.91

Григорьева А. И., Колодин М.Ю. Основные современные модели разработки ПО на основе СПО.

Аннотация. Разработка программного обеспечения на основе свободного и свободно-распространяемого программного обеспечения (ПО) — весьма востребованная в современных условиях тема. С одной стороны, набор инструментального и организационного ПО весьма широк и доступен на всех программно-аппаратных платформах, с другой — имеются значительные юридические и методологические сложности и неопределенности в практическом применении такого ПО в России. В статье рассматриваются как свободные и открытые, так и коммерческие экономически эффективные модели разработки.

Ключевые слова: разработка программного обеспечения, технологии, сопровождение, свободное программное обеспечение, эффективность.

Grigorieva A. I., Kolodin M. Y. Main Contemporary Models of Software Design Based on FLOSS.

Abstract. Software design based on Free/Libre and Open Source Software (FLOSS) is very important contemporary theme. On the first side, the amount of instrumental and organizational software is rather wide and is available on all software-hardware platforms, on the other side there are significant juridical and methodological difficulties and uncertainties in practical application of such software in Russia. Here we study both FLOSS and commercial economically efficient design models.

Keywords: software design, technologies, maintenance, free software, efficiency.

1. Введение. На всех этапах развития информатики разработка ПО была и остается главной темой. Средства разработки программ в наше время часто являются большими программными комплексами, весьма дорогими, приобретение их экономически целесообразно только при наличии потока коммерческих разработок определенного фиксированного типа и свойственно прежде всего крупным фирмам-разработчикам. В то же время имеется большой опыт такой разработки на основе свободного (СПО) и свободно-распространяемого программного обеспечения, и это весьма востребованное в современных условиях направление [1].

Набор инструментального и организационного СПО весьма широк, охватывает все стороны разработки ПО и его сопровождения, и доступен на всех программно-аппаратных платформах. Однако имеются значительные юридические и методологические сложности и неопределенности в практическом применении такого ПО в России.

Немаловажным вопросом остается и экономическая эффективность применения СПО в организациях, прежде всего в коммерческих. Имеющийся мировой опыт показывает, что и на СПО можно зарабатывать, что не противоречит свободным лицензиям.

2. Понятие о моделях разработки и распространения ПО. Заметим, что исторически именно СПО в его «наивном» виде появилось первым. Действительно, на начальных этапах разработки и распространения программ общего назначения почти все результаты деятельности программиста тут же становились доступными в первую очередь его соседу, от него — далее, и так для всех желающих. Здесь, конечно, не идет речь о закрытых военных проектах, а также о драйверах уникальных устройств. Понятно, что до всеобщей стандартизации и унификации множество устройств были оригинальными, часто единичными, а программы по их обслуживанию — закрытыми. Но по мере развития аппарата авторского права, повышения сложности и ценности разработок, явного выделения их из общей массы работ и, в частности, отделения от разработки аппаратного обеспечения, программы стали приравниваться к литературным произведениям, за их копирование, распространение, использование стали взиматься существенные суммы денег по все более усложняющимся лицензионным схемам, и сейчас едва ли найдется человек, который смог бы без обращения к специальной справочной и юридической литературе и без консультаций со специалистами ответить на «простые» вопросы о правилах и даже возможности использования всего ПО, установленного на его собственном компьютере. Это неизбежный результат развития современной цивилизации в коммерческом направлении.

Однако есть и более удобные модели, которые применяются во всем мире уже много лет. Это использование свободного и свободнораспространяемого ПО.

3. Свободное ПО — точное определение. Прежде всего, необходимо дать определение СПО.

Существует много модификаций «свободного» ПО, множество типов лицензий, много способов и особенностей их применения. Здесь рассмотрены только основные.

В соответствии с классическим определением [2], применительно к ПО есть четыре области свободы:

- 1) использования,
- 2) изучения и модификации,
- 3) распространения исходного продукта,
- 4) распространения модифицированного (производного) продукта.

Для реализации второй и четвертой областей свобод нужен доступ к исходным текстам программы. В общем случае важны первая и третья области свободы, но для профессиональных целей (а именно они нас интересуют в данном случае) нужны и вторая и четвертая области свободы.

Свободно-распространяемое (бесплатное) ПО (СРПО) не дает доступа к исходным текстам программ. Соответственно, мы не можем быть уверенными, что программа не содержит закладок и другого злонамеренного кода, не можем повторно использовать наработки коллег. Однако для некоторых областей работы достаточно и этого типа ПО, особенно если оно широко распространено, и имеется уверенность, подкрепленная массовым опытом, что данное ПО достаточно надежно, корректно, соответствует спецификациям, документировано и, таким образом, может быть использовано для снижения денежных затрат.

Отметим, что и свободное ПО может быть коммерческим. Оно может также иметь «двойные лицензии», указывающими, например, что данное ПО бесплатно и даже полностью открыто для некоммерческого, научного, образовательного, личного использования, но платно для других применений; это один из основных способов заработать на СПО, которые рассмотрены далее.

4. Средства разработки ПО и сопровождающие инструменты: требования, примеры, выбор. Для разработки ПО необходимы несколько групп программ [2]:

- 1) утилиты ОС;
- 2) компиляторы и интерпретаторы языков программирования,
- 3) средства управления разработкой (управление сборкой, версиями кода и документации);
- 4) средства поддержки деятельности разработчиков;
- 5) средства отладки и сопровождения.

Каждая из них имеет своих представителей как в коммерческом «мире», так и в «свободном».

Утилиты ОС нужны для общих организационных нужд (организация выполнения программ), для справочных целей (помощь по использованию команд), для настройки ОС и пакетов установленного ПО, и т. п. Здесь полезно максимально использовать преимущество UNIX-систем: их технологичность. Изначально UNIX-системы, к каковым относятся и Linux, и FreeBSD, разрабатывались как набор многочисленных небольших утилит, каждая из которых имеет небольшую функциональность, зато выполняет свои функции хорошо; из таких отдельных утилит и организуются технологические цепочки для каж-

дого пользователя, проекта или организации. Соответствующие наборы утилит есть практически на всех платформах; в том числе под коммерческой MS Windows можно порекомендовать пакеты cygwin, mingw, unxtools, а также отдельные аналоги утилит UNIX.

Свободные системы разработки ПО широко распространены на всех платформах. Прежде всего это пакеты cygwin, mingw, gcc, GTK, Qt, среды Eclipse, NetBeans, языки Python, Perl, Tcl, Java и др. Важно, что для каждого средства есть свободный ли хотя бы бесплатный вариант, причем с практически всеобъемлющими библиотеками готового отлаженного, легко получаемого, документированного специализированного ПО. Аналогично обстоит дело и с СУБД, как SQL-базируемыми, так и иными. Но всегда нужно иметь в виду, что они распространяются под существенно разными лицензиями; в частности, массово используемая СУБД MySQL является коммерческой при использовании в коммерческой ОС для коммерческих целей.

Средства поддержки разработки ПО также присутствуют, это следующие системы: сборки программных продуктов, управления версиями (cvs, svn, hg, bazaar, git, и др.), управления ошибками (типа bugzilla). Многие из них изначально разработаны именно как свободные продукты и могут быть применены в любой области своего использования.

Средства сопровождения и организации деятельности разработчиков важны сейчас как никогда раньше вследствие многократно возросшей сложности создаваемых продуктов, территориальной и временной распределенности разработчиков, многоверсионности кода. Причем важны не только персональные организующие средства, но все более — коллективные инструментарии и организационные дисциплины масштабов компании (института, рабочей группы и т. п.).

Особо остро встает задача защиты программ и разработчиков от внешних и внутренних угроз. Здесь важна правильная организация самой ОС, а также дисциплина в организации труда.

Становится необходимой переносимость средств разработки и получаемых программ и систем между следующими объектами:

- программно-аппаратными платформами;
- различными культурно-языковыми окружениями (удобство «локализации»);
- между проектами, разработчиками (повторное использование кода) и т. п.

Это важно, поскольку повышает технологичность работы и позволяет в общем случае существенно сэкономить человеческие, временные, материальные затраты.

5. Юридические вопросы применимости лицензий на СПО в России. Часто считают, что «лицензии», разработанные в США и других странах с иной, нежели в России, правовой системой, неприменимы в нашей стране. Это неверно, поскольку важно не точное соответствие между таковыми «лицензиями» (как бы они ни назывались) и постоянно меняющимся российским законодательством, а отсутствие явных противоречий между лицензиями и законами.

При разработке на основе СПО нужно учитывать, наследуема ли лицензия на то ПО, с помощью которого производится новое ПО. В случае наследуемой лицензии (например, GPL2) производитель обязан и далее выпускать продукт как свободный (т. е. с открытыми текстами или с возможностью их легкого получения), с приложенной к производному продукту той же самой лицензией. Это мешает некоторым разработчикам, поскольку они, например, используют собственные закрытые наработки и не желают раскрывать их код. В таком случае лучше использовать ненаследуемые лицензии, позволяющие менять тип производного ПО, в частности, закрывать его части, а также продавать; такова, например, лицензия BSD.

При приобретении и установке ПО, в том числе СПО, даже в том случае, если программа «просто» скачивается из Интернета, необходимо произвести официальный учет программы, постановку на баланс (управленческий, а для платного ПО — и на бухгалтерский), и постоянно отслеживать все изменения в установках ПО на каждом компьютере в организации. Это трудоемкий процесс, но, во-первых, он необходим, а во-вторых, есть немало программных средств для его автоматизации. Полезно иметь общедоступный список ПО с указанными лицензиями по каждому компьютеру и по каждой программе; и лучше именно с этого начинать организацию оснащения компьютеров необходимым ПО, а не пытаться задним числом вспомнить, что где установлено. Наш опыт показывает, что регулярный учет требует малых затрат сил и времени, а автоматизация на основе внешних и самописных программ с выводом сводной информации на интранет-сайт легко реализуется и надежно работает.

6. Экономически эффективные модели, или как заработать на СПО. Следует отметить, что разработка и распространение ПО как в виде СПО, так и под иными лицензиями, не противоречит принципу получения материальных или иных благ [3].

Прежде всего, даже на основе изначально свободного ПО можно получать коммерческие продукты; достаточно использовать ненаследуемые лицензии типа BSD.

Можно также использовать двойное лицензирование, разделяющее использование производных программ в личных, офисных, образовательных, научных и тому подобных целях, с одной стороны, и в коммерческих целях, с другой; за последнее можно взимать плату.

Основным вариантом получения прибыли за распространяемые бесплатно программы является взимание платы за консультирование и обучение (очные, заочные, сетевые курсы и семинары). Известно, что и для ОС семейства Linux есть курсы, стоящие десятки тысяч рублей за одного обучаемого; по окончании курсов учащиеся, как правило, проходят официальные экзамены фирмы-разработчика (или сертифицированной ею компании) и получают (в случае успешного прохождения) соответствующие дипломы, действительные постоянно или в течение определенного времени.

Классическим способом привлечения денежных средств к разработке и получения выгоды является создание двух (и более) параллельных версий одного и того же продукта: младшая версия с минимальными возможностями распространяется бесплатно, а более старшие требуют все больших денежных вложений. То же относится и к документации: с самим продуктом может быть поставляема только самая необходимая справочная информация по установке, запуску и использованию ПО, достаточная для специалистов; если же продукт все же непонятен, если возникли сложности, то можно купить более полную версию документации, книги по данному продукту или технологии. Иногда отдельно продается основная документация по продукту, полная или специальная; но это не самый уважаемый сообществом вариант.

Могут быть и прямые формы оплаты: консультирование по телефону, по электронной почте, на форумах и т. п. Каждый такой способ должен быть правильно просчитан с точки зрения вложений и предполагаемой выгоды (включая затраты на рабочее время консультирующих сотрудников, подготовку материалов, учет звонков и писем и прочие процедуры взаимодействия с клиентами).

Есть обычный способ получения денег, знакомый научному и образовательному сообществу: гранты. Если ваша тема имеет общественный интерес, то можно подать заявку на грант и, возможно, обеспечить себя и организацию средствами для ведения вашего и других проектов.

Фирмы, производящие аппаратное обеспечение, заинтересованы в максимальной прибыли от продажи их продукции. Поскольку немало пользователей работает на СПО, то эти фирмы сами разрабатывают как минимум бесплатное ПО для того, чтобы и такие пользователи оказались в числе их покупателей. Иногда эти фирмы публикуют документацию по программному интерфейсу для разработчиков, что позволяет независимым разработчикам по всему миру создавать нужные программы для поддержки нового оборудования; таким образом, фирмы и сами экономят на разработке ПО, и обеспечивают расширение продаж.

Наконец, блага от реализации собственных разработок могут быть не только материальными, по крайней мере, не обязательно связанными с немедленным получением материальной выгоды. Хорошо сделанная программа работает на имя автора, что сейчас повышает его статус в профессиональной среде, а в дальнейшем может стать поводом для получения новых выгодных контрактов или должностей.

6. Заключение. Чтобы разработка ПО и программных продуктов на основе ПО, а также их широкое использование были перспективными, для каждого случая нужно учитывать много факторов и делать выбор между СПО и коммерческим ПО. Однако в большинстве случаев преимущество должно быть за СПО.

Литература

1. Григорьева А.И., Колодин М.Ю. Свободное программное обеспечение в науке и образовании — проблемы, решения и перспективы использования // Тр. СПИИРАН. 2008. Вып. 6. С. 78–92.
2. Свободное программное обеспечение в государственном секторе. М.: INFOFOSS.RU, 2007. 112 с.
3. Как стать электронной компанией в России. II. Руководство. Программа сотрудничества ЕС и России. Электронные навыки для Российских малых и средних предприятий. II. Проект EuropeAid/122200/C/SER/RU. Контракт № 2006/109-210. СПб, 2007. 178 с.

Григорьева Алла Ивановна — канд. техн. наук, доцент, руководитель исследовательской группы информационных технологий в образовании (ИГИТО) Учреждения Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: информационные технологии в образовании, обоснование и создание базовых автоматизированных информационных систем для различного типа учреждений. Число научных публикаций — 118. gai@computer.edu.ru, www.computer.edu.ru/person/gai/; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, Россия; р.т. +7(812)328-0382.

Grigorieva Alla Ivanovna — Ph.D. in Technics, associate professor, head of Research Group for Information Technologies in Education (RGITE) of St.Petersburg Institute for Informatics

and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: information technologies in education, reasoning and creation of base automated information systems for different types of institutions. The number of publications — 118. gai@computer.edu.ru, www.computer.edu.ru/person/gai/; SPIIRAS, 39, 14th Line V.O., St.Petersburg, 199178, Russia; office phone +7(812)328-0382.

Колодин Михаил Юрьевич — научный сотрудник исследовательской группы информационных технологий в образовании (ИГИТО) Учреждения Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: метасистемы, свободное программное обеспечение, образовательные технологии. Число научных публикаций — 74. myke@iias.spb.su, www.myke.spb.ru; СПИИРАН, 14-я линия В.О., д. 39, Санкт-Петербург, 199178, Россия; р.т. +7(812)328-0382. Научный руководитель — канд. физ.-мат. наук, доцент, ведущий научный сотрудник А.Л. Тулупьев.

Kolodin Mikhail Yurievich — researcher, Research Group for Information Technologies in Education (RGITE) of St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: metasystems, free and open source software, educational technologies. The number of publications — 74. myke@iias.spb.su, www.myke.spb.ru; SPIIRAS, 39, 14 Line, St.Petersburg, 199178, Russia; office phone +7(812)328-0382. Scientific supervisor — Ph.D., associate professor, leading researcher A.L. Tulupiev.

Рекомендовано СПИИРАН. Директор — чл.-корр. РАН Р.М. Юсупов.
Статья поступила в редакцию 14.12.2009.

РЕФЕРАТ

Григорьева А. И., Колодин М. Ю. **Основные современные модели разработки ПО на основе СПО.**

Разработка программного обеспечения на основе свободного и свободно-распространяемого программного обеспечения (ПО) — весьма востребованная в современных условиях тема. С одной стороны, набор инструментального и организационного ПО весьма широк и доступен на всех программно-аппаратных платформах, с другой — имеются значительные юридические и методологические сложности и неопределенности в практическом применении такого ПО в России.

В этих условиях необходимо рассмотреть имеющийся спектр решений по каждому из основных направлений разработки ПО и ее поддержки, сравнить основные программные и организационные решения, определить, какими юридическими нормами нужно руководствоваться в нынешних условиях.

На уровне операционных систем (ОС) рекомендуется использовать ОС семейства Linux, для серверных приложений — также FreeBSD; именно эти системы лучше всего организованы, защищены и технологичны. Свободные и свободно-распространяемые ОС типа Sun Solaris имеют ограниченное применение и хуже поддерживаются в нашей стране, хотя и содержат ряд полезных свойств. ОС типа MS Windows весьма дороги, имеют много скрытых возможностей и закрытый код, более непредсказуемы в технологическом плане; их применение возможно для закрытых, крупных транснациональных корпораций, имеющих свои регламентированные наборы ПО, управленческих процедур и т. п.

На уровне инструментального ПО стоит порекомендовать семейства открытых компиляторов типа GNU, MinGW, а также свободные и свободно-распространяемые версии интерпретируемых языков типа Python (прежде всего), Perl, Tcl (в меньшей степени), системы управления версиями и т. д.

На уровне средств организации деятельности сотрудников и их коллективов и поддержки разработки ПО в первую очередь стоит отметить открытые офисные пакеты, а также бесплатные сетевые пакеты программ типа Google (имеющие открытый API).

Юридически лицензии типа GNU, BSD, Creative Commons и аналогичные им не противоречат современному российскому законодательству; однако для их корректного применения нужно быть очень аккуратными при приобретении, регистрации и сопровождении ПО. Нужно выработать и поддерживать в масштабах организации целый набор четко прописанных в должностных инструкциях действий.

В целом использование закрытого/коммерческого и открыто-го/свободного ПО будет развиваться параллельно, что правильно, поскольку дает пользователям возможность выбора в большом диапазоне доступных средств. В общем случае мы рекомендуем применять преимущественно полностью свободное ПО.

SUMMARY

Grigorieva A. I., Kolodin M. Y. **Main Contemporary Models of Software Design Based on FLOSS.**

Software design based on Free and Open-Source Software (FLOSS) is highly required contemporary theme. On the first side, the amount of instrumental and organizational software is rather wide and is available on all software-hardware platforms, on the other side there are significant juridical and methodological difficulties and uncertainties in practical application of such software in Russia.

In these conditions it is necessary to study all the existing variety of solutions on all the main directions of software design and its support, to compare principal software and organizational means, to define which juridical norms are to be used nowadays.

On the level of operating systems (OS) we recommend to use Linux family, for server applications also FreeBSD is good; exactly these systems are better organized, protected and technologically supported. Free and not-paid-for systems like Sun Solaris have restricted applicability in our country, though they have many useful features. MS Windows systems are usually rather expensive, have many hidden functionality and closed code, are more unpredictable technologically; their application is possible for closed, large transnational corporations that have their own requirements for software, management procedures etc.

At the level of instrumental software we recommend open compilers like GNU, MinGW, as well as FLOSS versions of interpreted languages like Python (mainly), Perl, Tcl (less), versions control systems etc.

At the level of organizational means for collaborates and their collectives and for software design support we should note open office packages, as well as free network office and groupware organizing programs like Google (that have their open API).

Legally licenses like GNY, BSD Creative Commons and like do not contradict to the existing Russian laws; but in order to apply them correctly one should be very attentive in acquiring, registration and maintenance of such software. The organization should elaborate a set of exactly specified instructions for its associates and maintain the process of accounting for all the software used on everyday basis for all working places.

Generally application of closed and commercial, from one side, and of free software, from the other side, will be done in parallel; and it is normal, because it gives better selection possibilities for the users. Nevertheless, for most common cases we recommend to use FLOSS.