

## **И.С. Маринин** **ТРЕБОВАНИЯ СОВРЕМЕННОГО БИЗНЕСА К РАЗРАБОТЧИКАМ** **ИНФОРМАЦИОННЫХ СИСТЕМ**

---

*zverbars@mail.ru*

*Российский государственный профессионально-педагогический университет*

*г. Екатеринбург*

Современные информационные системы окружают нас повсюду: в промышленности, медицине, связи, быту и т.д. Информатика как бизнес и как наука продолжает динамично развиваться, она нуждается в квалифицированных кадрах. Требования к специалистам продолжают расти быстрее, чем уровень подготовки выпускников большинства учебных заведений. И даже несмотря на снисходительность многих работодателей к опыту работы, проблема трудоустройства среди молодых специалистов остается актуальной. Насколько наше образование отстает от требований современного бизнеса в области ИТ трудно судить. Автор статьи лично столкнулся с уже общепринятой практикой среди работодателей в отрасли разработки ПО, суть которой заключается в следующем.

Предположим вы – недавний выпускник ВУЗа, дипломированный специалист в области информационных технологий. Вы претендуете на вакансию начинающего разработчика (junior developer) или помощника программиста. Казалось бы вполне подходящая работа для целеустремленных, но пока еще неопытных молодых специалистов. Но не все так легко и просто как кажется, ваш диплом зачастую служит лишь пропуском на собеседование с работодателем, это даже не половина пути к получению заветной должности. После успешного собеседования вам дают так называемое «тестовое задание» (небольшая информационная система или отдельная программа) - достаточно популярный ход. В течение определенного срока выполненное тестовое задание сдается и оценивается работодателем.

Составим основные критерии оценки тестового задания в порядке увеличения их важности для работодателя:

1. Работоспособность.
2. Документированность.
3. Качество реализации.

Логично, что работодателя интересует в первую очередь КАК реализовано задание, ведь именно это является показателем качества. Именно качество, которое не могут обеспечить большинство молодых специалистов, и служит преградой в трудоустройстве и дальнейшем профессиональном становлении.

Какие же знания и умения необходимы современному разработчику, чтобы продемонстрировать высокое качество своей работы? Разумеется, это базовые знания информатики, знание тех или иных языков и систем программирования, умение выстроить логику и алгоритм какого – либо процесса. Этому и многому другому молодые специалисты обучались в ВУЗе. Возникает вопрос: если выпускник владеет основными знаниями и умениями и способен дальше развиваться сам, то что же было упущено во время обучения и почему он не соответствует требованиям современного бизнеса?

Разработка ПО – это целая индустрия, такая же, как тяжелое машиностроение, объединяющая множество специалистов различных областей. Если присмотреться к сложности современных информационных систем, становится очевидно, что их создают слаженные команды.

Одно из странных свойств, присущих индустрии программного обеспечения, связано с тем, что какой-либо термин может иметь множество противоречивых толкований. Наиболее многострадальным оказалось понятие архитектура. У этого термина много трактовок, но, в любом случае, архитектура – это разбиение системы на компоненты [1].

При чем же здесь архитектура? В одиночку реализовать «монолитную» систему сложно по объему трудозатрат, а коллективная реализация монолитного проекта напоминает

басню Крылова «Лебедь, Щука и Рак». Только разбив систему на части становится возможным грамотное разделение труда в коллективе. Подобно конвейеру Форда, каждый специалист отвечает за свою «деталь» всего реализуемого «механизма».

Проблема молодых специалистов зачастую заключается именно в том, что они не знакомы с таким подходом к разработке информационных систем. За весь период обучения лишь единицы сталкиваются с изучением принципов построения архитектуры ПО и командной разработкой.

Какие преимущества дает знание принципов построения архитектуры ПО?

1. Как уже говорилось, это позволяет облегчить реализацию сложной системы путем разделения труда. Каждый из разработчиков реализует какую-то свою часть, причем реализует ее по заданным стандартам и спецификациям, ведь в таком деле чрезмерная самостоятельность станет преградой к интеграции компонентов между собой.

2. Становится проще следить за качеством реализуемого продукта посредством тестирования каждого из компонентов. В компаниях, занятых разработкой ПО, как правило, есть целое подразделение, занимающееся тестированием. Если провести аналогию с конвейером на автозаводе, то можно с достаточной долей уверенности говорить о качестве автомобиля, зная, что каждая деталь в нем проверена, и все верно собрано воедино.

3. Увеличивается жизненный цикл и гибкость системы. Требования к созданному продукту могут постоянно меняться, и правильное построение архитектуры ПО позволяет значительно снизить затраты на дальнейшую модификацию системы. Представьте себе автомобиль, колеса которого были бы «намертво» приварены к ступицам. С наступлением зимы поменять шины с летних на зимние стало бы практически невозможно.

4. Проектирование архитектуры подразумевает использование шаблонов (design patterns) – проверенных общепринятых решений, что положительно влияет на качество и время создания системы [2].

Особенность классического университетского образования такова, что изучение основ в области программирования и проектирования информационных систем сводится к созданию того, что уже было кем-либо создано и опробовано. Несомненно, необходимо изучать базовые алгоритмы и решения – без этого трудно будет освоить нечто более сложное. Но одними только базовыми знаниями не обойтись при разработке сложных систем. Яркий пример: вы можете быть экспертом в области электроники, но вы никогда не сможете представить себе каким образом соединены миллионы транзисторов в процессоре вашего компьютера.

Хорошее знание стандартных подходов и шаблонов проектирования позволит разработчику свести к минимуму «изобретение велосипеда», позволит сконцентрироваться на том, что действительно важно. Не случайно к вакансиям «разработчик ПО», «программист» и т.п. все чаще можно встретить требование «знание шаблонов проектирования».

Материалы для изучения по данной теме имеются в достаточном количестве. Проектированию архитектуры посвящено множество литературы, перечень существующих шаблонов и методологий проектирования можно признать внушающим. Мартин Фаулер и Джимми Нильсон заслуженно считаются «классиками» концепции типовых решений (design patterns). Существует множество руководств по проектированию архитектуры от известных компаний, таких как Microsoft. Многие среды разработки включают в себя инструментарий проектирования архитектуры ПО, юнит-тестирования и документации проекта.

Помимо выше обозначенных знаний и умений, разработчик должен обеспечивать качество продукта путем постоянного взаимодействия как с заказчиками либо их представителями, так и с коллегами. Он должен понимать свою роль в проекте и нацеливать себя не только на выполнение проекта как такового, но и на достижение высокого качества своей работы.

Автор статьи убежден в целесообразности включения методологий построения архитектуры ПО в соответствующие дисциплины специальностей, связанных с разработкой информационных систем. Также необходимо уделить должное внимание воспитанию разработчика как целеустремленной и ответственной личности. Это позволит выпускать специалистов, способных обеспечить высокое качество своей работы, т.е. успешных специалистов.

#### *Список литературы*

1. Фаулер, Мартин. Архитектура корпоративных программных приложений.: Пер. с англ. — М.:
2. Издательский дом "Вильямс", 2006.
3. Нильсон, Джимми. Применение Domain Driven Development и шаблонов проектирования.: Пер. с англ. — М.: Издательский дом "Вильямс", 2008

**О.В. Марченко**

#### **ОТ ТЕХНОЛОГИЙ WEB 2.0 К SMART EDUCATION**

---

*olmar@yandex.ru*

*Пермский филиал МЭСИ*

*г. Пермь*

Одним из стратегических направлений деятельности Пермского филиала МЭСИ является использование компетентностного подхода в образовательной деятельности. Цель - формирование профессиональной компетентности, т.е. способности решать профессиональные задачи. В условиях информатизации всего общества, и образования в частности, одной из составляющих профессиональной компетентности является способность использовать информационные технологии для решения профессиональных задач. Одним из показателей уровня сформированности профессиональной компетентности является способность применять в профессиональной деятельности инструменты Web 2.0.

Web 2.0 – это «технологическая платформа, совокупность различных сервисов, на базе которых можно создать социальную сеть или какой-либо другой сервис» (Джемс Калаканис). Типичными навыками пользователя сети Web 2.0 являются умение оформить и развивать свое пространство в сети, включая регистрацию аккаунта, наполнение его контентом и т.д., а также возможности использования Интернета для коммуникации (почта, чаты, видеоконференции, календари), визуализации и трансформации (в том числе перевода) информации.

Например, при обучении иностранным языкам преподаватели Пермского филиала МЭСИ, применяя технологии Web 2.0, используют такие инструменты как Wiki (Англоязычная энциклопедия Wikipedia), Web-блоги (Twitter, Blogspot), социальные сервисы (Facebook, vkontakte.ru), сервисы мультимедийных ресурсов (Flickr, YouTube, TeacherTube, Slideshare). В настоящее время инструменты Web 2.0 полностью интегрированы в учебный процесс Пермского филиала МЭСИ. Такая интеграция позволяет организовывать дистанционные учебные курсы, предоставляет широкие возможности для индивидуального консультирования, организации сетевой научно-исследовательской работы, для создания учебного контента в сотрудничестве, для классификации и хранения учебных материалов и ресурсов, для осуществления межкультурной коммуникации с помощью современных видов связи.

В настоящее время объемы знаний удваиваются каждые два часа, большая часть образовательного контента отстает от создаваемых и используемых технологий на два-три поколения. МЭСИ одним из первых ставит вопрос об объединении образовательных учреждений в рамках Smart Education («умного» образования) для осуществления совместной образовательной деятельности в сети Интернет на базе общих стандартов, соглашений и технологий. Smart Education предполагает переход от книжного к активному контенту: знания размещаются в репозиториях в форме знаниевых объектов. Новые