

## КРАТКИЙ АНАЛИЗ ИНСТРУМЕНТАРИЯ И СРЕДСТВ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ

*Кыргызский Государственный Технический Университет им. И. Раззакова*

**БАБАК В.Ф., КАСЫМАЛИЕВА А.Т., ЗВЕРЕВ П.А.**

[izvestiya@ktu.aknet.kg](mailto:izvestiya@ktu.aknet.kg)

*В статье излагается точка зрения авторов на использование известных и наиболее популярных программных комплексов (связок), применяемых при разработке проектов современных больших (корпоративных) информационных систем. Приводятся сильные и слабые стороны этих связок. Статья написана в дискуссионном стиле, отражает опыт работы авторов с этими связками и не навязывает никаких предпочтений на использование той или иной связки.*

*The article presents a view of authors to use well-known and most popular software packages (bundles), used in the development projects of modern large (corporate) information systems. Given the strengths and weaknesses of these bundles. The article is written in the style of the discussion, reflects the authors' experience with these bundles, and does not impose any preference to the use of bundles.*

В статье рассматриваются две возможные и, пожалуй, самые популярные и эффективные связки, которые используются обычно при разработке Internet- и Intranet-информационных систем (ИС): «**Apache + PHP + MySQL**» и «**IIS + ASP.NET + SQL Server**». Обе связки являются современным решением, однако каждая из них обладает своими особенностями, достоинствами и недостатками.

### **Связка «IIS + ASP.NET + SQL Server».**

Связка «IIS + ASP.NET + SQL Server» и каждый из этих продуктов по отдельности являются продуктами корпорации Microsoft. Эта связка принципиально базируется на операционной системе все от той же корпорации Microsoft – на Windows. Вследствие этого она является зависимой от некоторых API-функций операционной системы, что не является позитивным фактором. Большинство программистов считают, что по скорости, данная связка уступает «Apache + PHP + MySQL», что является мифом (тесты скорости доказывают обратное), однако в данном случае скорость быстроедействия системы, базирующейся на «IIS + ASP.NET + SQL Server», является разумной ценой тех средств, которые предоставляются для произведения деятельности системы. Microsoft предлагает полноценное решение многих задач программирования – платформу dotNET Framework. Эта платформа уже содержит множество реализованных программных модулей и решений, а также богатую библиотеку компонентов разного назначения. Помимо всего прочего, для визуального проектирования систем предлагается среда разработки Microsoft Visual Studio, которая позволяет построить систему любой сложности с нуля.

ASP.NET реализован через платформу dotNET Framework. С появлением платформы dotNET Framework перечень языков реализации web-приложений расширился и, на данный момент это уже полноценный объектно-ориентированный Visual Basic, языки C# и J#.

Среда Microsoft Visual Studio позволяет расширять перечень языков разработки проектов – известны, например, расширения для PHP (проект «Phalanger») и Delphi/Pascal («Delphi Prism»),

Пролог. NET, и, скорее всего, этот список будет расширяться. Соответственно, технология ASP.NET не привязана к какому-либо языку программирования конкретно (что действительно для CLR-языков). Она работает в зависимости от выбранной разработчиком настройки, позволяя смешивать языки программирования.

Технология ASP.NET предлагает множество уже созданных компонентов разработки, таких как стандартные и расширенные элементы управления, фоновые компоненты, компоненты доступа к множеству СУБД. Помимо этого, можно использовать и компоненты сторонних производителей, которые будут зарегистрированы в операционной системе веб-сервера. Все это позволяет максимально автоматизировать разработку любого приложения и свести фактор человеческих ошибок к минимуму. Посредством Microsoft Visual Studio можно также проводить тестирование системы, оценку эффективности работы кода и многое другое.

СУБД Microsoft SQL Server является отличным решением при выборе СУБД. Она предоставляет убедительный графический пользовательский интерфейс (в полной версии). Базы данных, создаваемые этой реляционной СУБД, являются транзакционными, поддерживают блокировки, пользовательские роли, пользователей, репликацию и хранимые процедуры и функции, определяемые пользователем, а также многое другое. При всем этом SQL Server отлично управляется с данными и показывает отличную скорость и масштабируемость. Она также позволяет создавать пользовательские диаграммы базы данных и осуществлять работу в графическом виде. Помимо этого, Microsoft SQL Server содержит перечень очень полезных утилит, которые мало что может заменить.

Среда Microsoft Visual Studio также предлагает отличные службы предоставления отчетов, основываясь на данных, взятых из СУБД. Процесс создания приложения, предоставляющего отчет, при этом занимает минимум времени, но охватывает максимум возможностей реализации и решений в области программирования и дизайна. Программировать в некоторых случаях вообще не приходится, так как данная среда предоставляет практически полную автоматизацию создания приложений для отчетов. Считается, что только около 30% кода разработчикам приходится писать вручную.

Суммируя сказанное, можно отметить следующие положительные стороны связки «IIS + ASP.NET + SQL Server»:

1. ASP.NET не зависит от языка программирования.
2. Компиляция проектов осуществляется в промежуточный язык CIL, поддерживаемый международным стандартом CLR.
3. ASP.NET предоставляет богатый набор инструментов создания веб-приложений.
4. ASP.NET базируется на платформе dotNET Framework, которая обеспечена множеством базовых функций, что значительно облегчает кодирование и развертывание проектов.
5. Среда Visual Studio является готовым решением для проектирования как баз данных, так и приложений.
6. Visual Studio изначально создавалась как средство RAD, что позволяет более быстро создавать достаточно сложные системы.
7. ASP.NET имеет полностью объектную структуру, что позволяет подойти к проектированию сайта с точки зрения классической концепции ООП.
8. Обеспечивает хорошую переносимость (сериализацию) готовых продуктов, снижает нагрузку при разработке инсталляционных программ и на работу с реестрами.
9. Microsoft SQL Server является транзакционной СУБД, способной обеспечить реализацию структуры БД, ее обслуживание и бизнес-логику; имеет перечень

полезных утилит и способна осуществлять распределенные транзакции, а технологии ADO.NET еще более расширяют возможности масштабирования и обеспечивают безопасное соединение прикладных систем с Web-сервером.

Однако за все хорошее надо платить. Учитывая то, что связка является зависимой от платформы, придется ориентироваться на операционную систему Microsoft Windows, что само по себе недешево. Тем более – это единственный способ получить веб-сервер Microsoft Internet Information Services (IIS), который не поставляется отдельно, а является службой данной операционной системы. Вследствие этого, если потребуется поменять веб-сервер, придется менять его вместе с операционной системой, что является крайне неудобным, более того, это может повлечь еще и потребность в переработке и оптимизации кода с учетом требований новой операционной системы.

Учитывая, что обслуживание и кодирование приложений, реализованных на ASP.NET, полноценно возможно только в среде, потребуются затраты на приобретение ряда инструментальных средств и комплексов, обеспечивающих поддержку новой среды. К тому же обладателю набора компонентов среды Visual Studio потребуется произвести набор квалифицированного персонала, хорошо знакомого с новой средой разработки, либо переучить уже набранный персонал. СУБД Microsoft SQL Server является коммерческой системой, но в данном случае затраты полноценно ее окупают, так как, на наш взгляд, ее конкурентом может являться только СУБД Oracle.

Платформа dotNET Framework является закрытой, то есть изменения на данный момент в неё могут вносить только разработчики Microsoft, при этом у конечного пользователя нет возможности быть уверенным в том, что весь код в них выполнен оптимально. В отличие от компонентов компании Borland, исходный код компонентов dotNET Framework нельзя даже просматривать.

Полноценная функциональность некоторых веб-компонентов платформы dotNET Framework может поддерживаться только в браузере Microsoft Internet Explorer. Особенно опасно использование компонентов, работа которых осуществляется как на клиенте, так и на сервере (распределенные «двойные» компоненты). И совершенно не гарантировано, что видимый отчет в Internet Explorer будет полновесно представлен в других популярных браузерах, таких как Mozilla FireFox, Google Chrome или Opera. У некоторых компонентов просто «едет» оформление, некоторые вообще не работают. Привязка к браузеру может в данном случае быть фатальной.

Можно осветить следующие негативные стороны связки «IIS + ASP.NET + SQL Server»:

1. Профессиональные компоненты разработки системы будут стоить недешево.
2. Высокая цена специалистов, производящих обслуживание, и их рабочих мест.
3. ASP.NET прочно зависит от платформы dotNET Framework, что зажимает ее в «тиски» уже готовых компонентов.
4. Большинство сторонних компонентов Visual Studio – коммерческие, в том числе и службы отчетов, за которые тоже придется платить.
5. Неизвестно, какие ошибки могли допустить программисты Microsoft при создании компонентов dotNET Framework, что чревато возможностью взлома.
6. При смене web-сервера потребуется менять операционную систему.
7. Успешность реализации разработки может зависеть от браузера.

**Связка «Apache + PHP + MySQL»**

К связке «Apache + PHP + MySQL», хотя каждый ее компонент был создан независимо от остальных, было предъявлено всего два требования: переносимость и бесплатность. Каждый из компонентов связки в свое время привлек множество внимания и стал популярным именно по этим двум причинам. Вся связка (и каждый ее компонент по отдельности) может быть установлена на Linux, Unix либо Windows без каких-либо потерь функциональности. Учитывая, что к сети может быть подключено множество компьютеров, каждый из которых может использовать разную операционную систему, данный пакет быстро набрал свою популярность как среди хостинговых компаний, так и среди разработчиков. Учитывая то, что свежий полнофункциональный дистрибутив каждого из компонентов можно «скачать» в Internet в любой момент времени, описание функциональности всей связки можно свести к одной фразе – «дешево и сердито».

Веб-сервер Apache является универсальным и превосходно настраиваемым решением. Его базовый комплект не поддерживает на данный момент ASP.NET (хотя есть сторонние модули поддержки), но поддерживает большинство (если не все) остальные web-технологии. Его настройка так же тонка, как настройка любого музыкального инструмента, – наиширочайший выбор опций позволяет регулировать пользовательский доступ к ресурсам, вести обработку специфических типов ресурсов и MIME-типов. Apache часто становится фрагментом мобильных пакетов разработки веб-приложений: на данный момент его используют «Д.н.в.р» («Джентльменский набор веб-разработчика» или «ДеНВеР») и «WAMP» («Windows + Apache + MySQL + PHP» – аналог «LAMP»). Web-сервер Apache является предпочитаемым web-сервером у групп разработчиков, работающих с PHP. Разработан даже специальный модуль, который позволяет подключить PHP к Apache как часть его сервиса. Apache и PHP – творение сообщества различных категорий специалистов, а не одной корпорации, монополизирующей одновариантность связки.

Язык серверных сценариев PHP («PHP: Hypertext Preprocessor» – препроцессор гипертекста) в свое время эволюционировал из PERL. Свою популярность PHP приобрел сначала среди новичков, которые хотели быстро создать не статический сайт. Затем появилось сообщество, которое и по сей день развивает PHP. В своей структуре он имеет C-подобный синтаксис с доступной альтернативой, слабую типизацию переменных и встраиваемость кода прямо в HTML-разметку (что, справедливости ради, следует отметить – не лучшее решение). Начиная с 4-й версии, PHP поддерживает программирование на объектах, но несколько своеобразное – перегрузка представлена в необычном виде, событийной моделью могут быть только примитивные триггеры. Так же, как и в языках для ASP.NET, присутствует так называемый «GC» («Garbage Collector» – сборщик мусора), который автоматически чистит память без необходимости явного высвобождения использованных переменных и ресурсов. В основном программисты на PHP пользуются его функционально-ориентированной стороной.

Конечно, учитывая это, можно сказать, что PHP несколько обеднен, но свои задачи он выполняет не «навороченностью», а простотой и гибкостью. Любой сценарий на PHP является линейным, но главное преимущество – модульность, которой нет в других языках сценариев. Сценарий PHP способен включать функционал из других файлов сценариев «налету», что делает его еще более эффективным. Принципиально код PHP можно хранить даже в таблицах реляционной базы данных, а потом поэтапно загружать и выполнять (не самое лучшее решение, но сам факт обнадеживает).

К бесплатному интерпретатору PHP можно подключать различные расширения, модернизирующие функциональный состав языка. Большинство из них бесплатные и при этом предоставляют весьма мощный функционал. К примеру, можно подключить библиотеку для создания FLASH-роликов с анимацией, PDF-документов или рисунков прямо «на лету». Есть и

коммерческие проекты для PHP. Разработку расширений ведут как сообщества, так и компании. Причина хорошей интеграции PHP и сторонних динамических библиотек – исходный код интерпретатора, который доступен всем желающим.

Есть еще одно достоинство языка PHP – его простота и легкость в изучении. Создавать сценарии можно при помощи самого обыкновенного текстового редактора, такого, как «Блокнот Windows», а примерами работ пестрит весь Интернет. К тому же у PHP есть хорошая справочная система, «скачать» которую можно там же, где сам дистрибутив. Хотя это вряд ли можно учитывать, так как при разработке крупных систем необходим квалифицированный персонал, обладающий широким спектром знаний, а не новички.

Еще одно удобство PHP – способность соединяться с большинством современных СУБД (драйвер для SQL Server 2005 и выше можно найти в сети MSDN), но «родным» для него всегда являлась именно СУБД MySQL. «Родная» связка позволяет добиваться поразительных результатов при низких затратах оперативной памяти сервера.

СУБД MySQL является реляционной (не в чистом виде) СУБД с открытым кодом, принадлежащей компании MySQL AB, выкупленной Sun Microsystems, ныне входящей в состав корпорации Oracle. Начиная с 5-й версии, MySQL поддерживает возможности, которые ранее представлялись лишь в коммерческих СУБД: транзакции, блокировки, хранимые процедуры и функции, курсоры, триггеры, ссылочная целостность и другие. При всем этом данная СУБД бесплатна, за исключением тех случаев, когда она входит в состав коммерческого продукта. Наряду с ведущими СУБД, MySQL показывает отличные результаты по скорости и использованию памяти. На данный момент она может работать на базе более 20 операционных систем и является стандартом «де-факто» у хостинг-провайдеров стран СНГ. Драйверы подключения к MySQL есть у многих сред (так же и в Visual Studio), хотя MySQL может работать и через ODBC.

Отличительными особенностями MySQL можно считать ее диалект SQL (впрочем, как и у любой другой СУБД), возможность менять типы таблиц, что позволяет более эффективно ими пользоваться и хранить в них данные. Так же имеется возможность определения перечисляемых типов данных и множеств, как атомарных значений, что является неприемлемым для реляционных СУБД в чистом виде. Поэтому такое высказывание, как: «MySQL – чисто реляционная СУБД», – без учета особенностей, является ошибочным.

Таким образом, для связки «Apache + PHP + MySQL» можно отметить следующие положительные стороны:

1. Все компоненты связки являются бесплатными, за исключением коммерческого использования.
2. Все компоненты связки являются кросс-платформенными и не имеют жесткой привязки к операционной системе.
3. В единой связке компоненты показывают высокую производительность по низкой цене.
4. Для осуществления разработки и обслуживания не нужно никаких специализированных программ.
5. Присутствует отличная справочная система.
6. Система, базирующаяся на основе PHP, может быть более динамичной и модульной, чем на основе ASP.NET.
7. СУБД MySQL может быть неплохой альтернативой MS SQL Server, при этом обеспечивая те же функции.

Конечно, нельзя сказать то, что связка «Apache + PHP + MySQL» является идеальной. Тем более на это есть ряд причин, первая из которых – отсутствие достойных графических оболочек для разработки. Это, впрочем, касается только Apache и PHP. Для СУБД MySQL можно найти бесплатные графические приложения, позволяющие как создавать модели базы данных, так и управлять самими данными. Однако нужно отметить, что то, что бесплатно, вряд ли будет идеально (если сравнивать с SQL Server, то отмеченные средства MySQL ему уступают). Конечно, корпорация Oracle предлагает и более удобные, но платные средства графической поддержки.

Web-сервер Apache в большинстве случаев конфигурируется именно с помощью простых текстовых редакторов. Без хорошего знания настроек вряд ли конфигурация окажется эффективной. У PHP в этом плане тоже есть «скелет в шкафу». На данный момент проектирование на PHP можно осуществлять во многих средах разработки веб-сайтов, в том числе и в среде Adobe DreamWeaver. Но автоматизация в данном случае очень сильно «хромает». Имеется и платная версия среды разработки от компании Zend (разработчиков PHP) – Zend Studio. Многие веб-программисты все же предпочитают бесплатные модернизированные текстовые редакторы.

Еще один негативный (как и позитивный фактор) этой связки – условия соглашений использования. Все компоненты связки можно использовать по модели «AS IS», то есть программные продукты принимаются «как есть», независимо от их удобств и неудобств. Разработчики компонентов не обязуются исправлять ошибки программного обеспечения, если допустили таковые, а так же не гарантируют полноценной преемственности предыдущих версий продуктов их новыми аналогами. Если в случае с Apache и MySQL эти проблемы мало актуальны, то для PHP это проблема. Часто новые версии PHP выходят с кардинальными изменениями, что не дает 100% гарантии, что полноценные рабочие сценарии, подходившие к прежней версии, подойдут к новой. К счастью, учитывая популярность PHP, разработчики при выпуске новых версий продуктов почти всегда оповещают пользователей об изменениях в продуктах, а также снабжают эти продукты пояснительными записками.

Важно сказать и об автоматизированных компонентах разработки. MySQL может быть использована через ODBC, что позволяет использовать любое CASE-средство разработки ER-моделей, поддерживающее реверс- и форвард-инжиниринг. К тому же есть и собственное неплохое средство MySQL WorkBench. Остальные компоненты остаются «в накладе». То есть практически каждый сайт, использующий связку «MySQL + Apache + PHP», требует огромной ручной работы. Если в ASP.NET, можно добавить кнопку на сайт, просто «бросив» ее на рабочее пространство, то на PHP ее придется вписывать соответственно формату DTD («Document Type Definition» – определение типа документа). То есть программист, взявшийся за разработку любой системы на базе этой связки, встает на путь «трудоголика». Однако есть и позитивные стороны. Если кто-то делает что-то на PHP, то сам несет за это ответственность – вы не будете иметь дело с программными ошибками других людей, потому что определяете свою работу сами. Системные неполадки в таком случае – вина разработчика конечной системы, а не разработчика ее базовых компонентов. Если страница на PHP некорректно отображается каким-либо из браузеров, то это лишь явное свидетельство того, что у разработчика либо плохое знание web-стандартов, либо недостаточно опыта в разработке.

Но в этой связке имеют место следующие негативы:

1. Отсутствие полноценных автоматизированных средств разработки.
2. Большое количество ручной работы, влекущее временные и денежные затраты на разработку и дизайн системы.
3. Постоянное осуществление модернизации системы для ее соответствия новым версиям ее базовых компонентов.

4. Постоянное обучение и тестирование разработчиков ИС на профессиональную пригодность.

## Сравнительный анализ обеих связок

Фактор	IIS + ASP.NET + SQL Server	Apache + PHP + MySQL
Автоматизация разработки и сопровождения	<u>есть</u>	не везде
Аппаратное обеспечение сервера	выше среднего	<u>среднее</u>
Временные затраты на производство	<u>низкие</u>	высокие
Гибкость, модульность и системная расширяемость	обычная	<u>повышенная</u>
Документирование базовых компонентов связки	хорошее	хорошее
Итоговая производительность	высокая	высокая
Квалификация сотрудников	<u>обычная</u>	высокая
Независимость от платформы, кроссплатформенность	в разработке	<u>есть</u>
Независимость от языка программирования	<u>есть</u>	нет
Поддержка актуальности и частота модернизации	<u>обычная</u>	частая
Поддержка разными браузерами	важен подход разработчика	
Сопровождение базовых компонентов связки	<u>высокое</u>	низкое
Цена базовых компонентов связки	высокая	<u>очень низкая</u>
Цена инструментария разработки	средняя	<u>очень низкая</u>
Цена разработки**	высокая	высокая

\* подчеркиванием отмечены лучшие результаты по фактору, если есть весомые отличия.

\*\*зависит от масштаба разрабатываемого продукта

### Заключение

Конечно, выбор технологии и инструментария разработки корпоративных ИС – дело предпочтений и опыта разработчиков. Однозначно утверждать, какая связка предпочтительнее в той или иной ситуации, весьма и весьма рискованно. Так же сложно утверждать, что ASP.NET меньше подходит для переноса системы в среду Internet/Intranet, чем PHP. Но в условиях КГТУ можно предположить следующее:

1. Захочет ли университет или его структурное подразделение (факультет, институт) приобрести лицензионную связку Windows Server + SQL Server + Visual Studio и её базовые компоненты? Во сколько это обойдется? Сомнительно, что вуз будет заинтересован в трате на это 3000 – 4000\$ (а то и более) и в доплате за ежегодное обновление компонентов.

2. Если же лицензионное программное обеспечение не по силам вузу, тогда один путь разработки корпоративной ИС – использование связки Linux + Apache + MySQL + PHP, либо использование версий с ограниченным функционалом. Трудоемко, но дешево.

3. Те, кто пренебрегает PHP и берет ASP.NET только из-за удобства работы, забывают о смысле слова «программист». На ASP.NET можно строить системы, но это не программирование в чистом виде, так как имеет слишком высокий уровень. Эффективность систем – вопрос к производителю.

4. За примером ходить далеко не следует – это известная в кыргызских вузах система «AVN», функционирующая на данный момент и в КГТУ им. И. Раззакова. Служба отчетов, построенная на основе одного из расширений ASP.NET, при загрузке под «Орега» выводит далеко не то, что под

«MS Internet Explorer», хотя «Opera Software» достаточно прочно придерживается web-стандартов. Однако, с точки зрения авторов статьи, это проблема кодирования, а не исполнения.

### **Литература**

1. <http://MisFitGeek.com> - Windows Hosting Experts;
2. <http://www.tbray.org/talks/php.de.pdf> («Issues in web frameworks») - Tim Bradley, Sun Microsystems;
3. <http://onjava.com> («The PHP Scalability Myth») - Jack Herrington;
4. <http://stackoverflow.com/questions/153947/whats-faster-php-vs-asp-vs-jsp-vs-cgi-etc/154657>
5. Jason M. Hanley, B. Math. Web Development: A Comparison of Three Major Platforms, Syllogistic Software Inc.
6. <http://www.wrensoft.com> («PHP vs ASP vs ASP.NET vs JavaScript vs CGI») – WrenSoft
7. [http://www.ehow.com/about\\_5637315\\_net-vs\\_php-speed.html](http://www.ehow.com/about_5637315_net-vs_php-speed.html) («Net Vs. PHP Speed») - Michael Dance