

Готовим базу знаний на основе wiki

Константин Овчинников

27 марта 2009 г.

Для тех, кому нужен практичный инструмент организации знаний.

Содержание

1 Введение	2
2 Причины успеха wiki	4
2.1 Проблемы с другими инструментами	4
2.2 Многомерность представления знаний	6
2.3 Новые функции и старые процессы	8
2.4 Развёртывание знаний в плоскую структуру	9
2.5 Расширяемость	9
2.6 Простота обучения	11
2.7 Простота доступа к приложению	11
3 Подходы к наполнению базы знаний	13
3.1 Демократия	13
3.2 Здравый смысл	13
3.3 Иерархия	13
4 Карты Freemind и интеграция с приложениями	14
5 Выводы	14

1 Введение

Лично я отношусь к тем людям, которым нужно всё и прямо сейчас. Никак не могу понять, почему это должно быть не так. Если есть потребность в организации знаний — с моей точки зрения, должен быть инструмент, который бы позволял организовывать знания нетехническому специалисту, а не вынуждал бы решать нудные методологические или технические вопросы. Да, есть и будут в дальнейшем экспертные системы, с семантическим разбором и прочим, но сейчас есть ингредиенты, которых вполне хватит для того, чтобы приготовить что-то «съедобное»:

1. Веб-приложения, обеспечивающие безопасный удалённый доступ;
2. Wiki-технология¹, которая позволяет:
 - a) Наиболее быстрым способом создавать содержимое (контент) за счёт простоты синтаксиса²;
 - b) Разрешать конфликты (история версий, запрет на одновременное редактирование, разделение прав на разделы и страницы по группам пользователей);
 - c) Просто каталогизировать ссылки за счёт отображения их в итоге в HTML;
 - d) Детализировать объект на отдельной странице — даже больше, переделывать структуру данных, вынося всю информацию на одну страницу или делать список разделов с отдельными страницами на каждый раздел;
 - e) Простой поиск по тексту³;
 - f) Технология тегов;
 - g) Технология плагинов vs. большой объёмный функционал сразу.

Я использовал до недавнего времени постоянно 3 wiki:

1. На работе — база знаний департамента;
2. На работе — локальные заметки по проектам;
3. Дома — моя личная база знаний: мои представления об окружающем мире, мои планы, мысли, которые мне не хотелось бы потерять.

Наиболее интересной является самая масштабная задача — организация базы знаний департамента, поэтому в дальнейшем мы и будем ориентироваться на что-то подобное. Эта задача предполагает максимальный объём требований и диктует

¹Подробнее о wiki

²Практика показала, что для начального обучения любого человека, владеющего компьютером, хватает 15 минут — дальше человек обучается сам, если есть подробное русское руководство.

³Опять же практика показала, что вполне можно обойтись поиском без учёта падежей и прочих языковых особенностей — это же не поиск по глобальной Сети.

только практичный подход к их реализации — время на работе оплачивается и ограничено.

В итоге во всех трёх случаях с технологической точки зрения я пришёл к одному и тому же — [dokuwiki](#)⁴ (Apache в качестве веб-сервера — собственно, это всё, что требуется). Позже на форумах нашёл несколько комментариев, видимо, таких же требовательных и занудных товарищей, как я — после очень долгого перебирания различных wiki-движков они тоже склонились к [dokuwiki](#). Система действительно исключительно грамотно написана и за кажущейся простотой скрывается мощная функциональность. На момент написания статьи доступно более 300 плагинов, что позволяет реализовать почти все имеющиеся фантазии сотрудников (я их аккуратно записывал на отдельную страницу в той же базе знаний департамента). Тем не менее, свой выбор я никому не навязываю.

Для справедливости назову ещё несколько движков, которые, как мне показалось, относятся к `mainstream`:

- Mediawiki
- MoinMoin

До этого в качестве базы знаний использовались сетевые папки, также мы пробовали VSS — в обоих случаях выбор инструментов оказывался неудачным. Использование конкретной версии wiki было успешным (на данный момент база знаний содержит около 3 Gb информации и, что важнее, этой информацией пользуются), на мой взгляд, по следующим причинам:

1. Многомерность представления знаний
2. Встраивание функций работы с базой знаний в существующие процессы
3. Развёртывание знаний в плоскую структуру
4. Расширяемость
5. Простота обучения
6. Простота доступа к приложению

Также в связи с использованием подобного инструмента появились некоторые подходы к наполнению базы знаний, о которых я также хотел бы здесь рассказать:

1. Демократичный подход к наполнению базы знаний
2. Использование здравого смысла при выборе способов оформления содержимого (контента)
3. Соблюдение строгости иерархической структуры

⁴[Веб-сайт dokuwiki](#)

В таком порядке мы и рассмотрим перечисленные пункты.

Ну и напоследок — некоторые «продвинутые» технологичные плагины. В частности, мы рассмотрим представление карт Freemind⁵ и интеграцию с другими приложениями (форумами и CMS).

2 Причины успеха wiki

Прежде всего, хотелось бы отметить, что этот успех — не хвастовство и не условность. То, что люди пользуются этим инструментом (и для чтения материалов, и для организации хранения новых материалов) без какого-либо дополнительного воздействия со стороны руководства, на мой взгляд, говорит само за себя.

Была идея распространить базу знаний и на весь холдинг — к сожалению, эта идея осталась нереализованной.

С одной стороны, есть объективная причина, заставляющая пользоваться базой знаний — информации, которая требует хранения и каталогизации даже исключительно для личного пользования слишком много — для традиционных инструментов. В своё время я разработал лично для себя структуру сетевых папок с пониженным количеством уровней вложенности и некоторым налёком на интуитивность организации данных. Кроме того, я, как правило, помню, где что лежит. Я пользовался этой структурой успешно, но это было мучительно. По именам файлов часто трудно определить содержание, поэтому приходится открывать много файлов и просматривать их, чтобы что-то найти.

Даже с хорошей памятью при поиске информации возникала одна и та же проблема. Я помнил о статье именно как об одной из статей, которые у меня хранятся в папке «articles». Когда же мне нужен был материал по какому-то предмету, я вспоминал, что у меня что-то было, но вот где... В руководствах? В документах по какому-то проекту? Или всё-таки там, где у меня «свалка» статей?

С другой стороны, есть причина, которая побуждает людей не выкладывать в базу знаний новую информацию — некоторые считают, что получают преимущества за счёт имеющихся у них уникальных знаний. Лично я никогда не считал это преимуществом консультанта — только более быстрое освоение новых знаний может быть преимуществом. Возможно, мой пример на кого-то подействовал и запустил реакцию. Тем не менее, рассчитывать только на альтруизм не стоит — это мы обсудим в разделе встраивания функций работы с базой знаний в существующие процессы.

Сначала мы вернёмся немного назад в истории и рассмотрим, почему нельзя обойтись более простыми инструментами.

2.1 Проблемы с другими инструментами

Как уже упоминалось, первые версии базы знаний департамента были в сетевых папках и в VSS. Что это из себя представляет, видно на рисунке. Специально при-

⁵Веб-сайт [Freemind](#)

вожу один из самых сложных случаев: в процессе поиска чего-либо натыкаешься на интересную статью, сохраняешь себе с исходным названием файла.

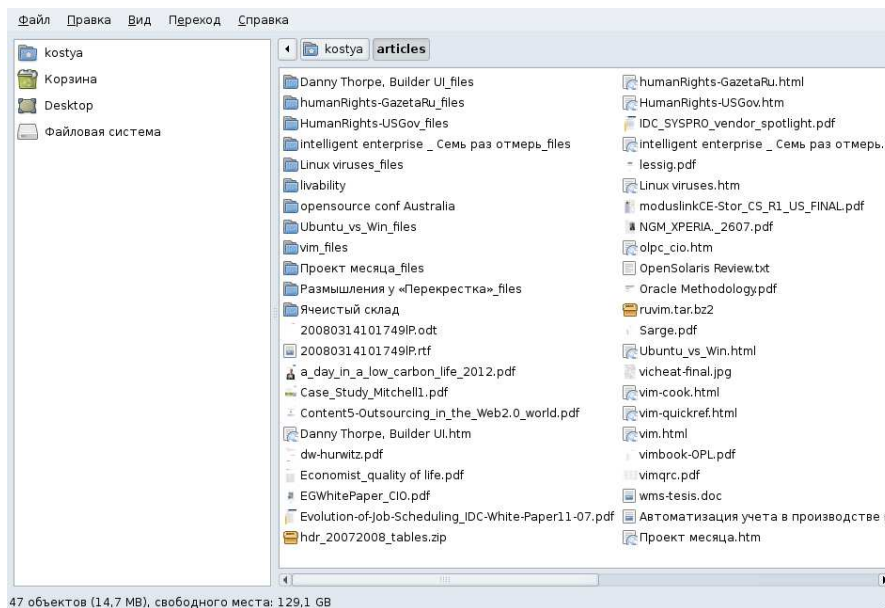


Рис. 1: Список статей в файловой системе

Подробно на этом останавливаюсь, потому что всегда найдутся «специалисты», готовые предложить замечательное решение, не требующее баз знаний и прочего. Вот задача. Что делаем? ОК, у нас много свободного времени, заходим в каждый файл, копируем заголовок статьи, переименовываем файл. Что это нам дало? Ничего. По старой журналисткой традиции название статьи, как правило, *вообще ничего* не говорит о её содержании. Скорее даже наоборот — запутывает.

Дальше, хотелось бы сохранить коллекции ссылок, доступные другим. Наверное, это смешно, но я использовал для этого текстовые файлы. Но ведь этот текстовый файл надо ещё догадаться открыть! MS Word для этого мне казался слишком громоздким, HTML-редактора под рукой не было, а сервисы хранения закладок совершенно не подходят для закрытой корпоративной информации, пусть даже это коллекция ссылок.

Ещё что поражает меня сейчас в приведённом скриншоте: какое большое количество свободного места, а там, где есть названия файлов, информации почти нет!

Обе эти проблемы позволяет решить движок wiki. По сути, это наиболее быстрый из известных мне способов создания HTML-контента. Но основная идея не в HTML. Должно быть описание файла и должен быть способ «спрятать» гиперссылку, при этом оставить читаемое название. Гиперссылки есть много где,

не только в HTML, но этот формат сразу приводит к идее сайта, доступного для других через Интернет. Поскольку люди работают в проектных офисах, эта возможность очень важна.

2.2 Многомерность представления знаний

Описанная ситуация привела к единому мнению по поводу сетевых папок и VSS — «не удобно!». Но это только одна из причин. Другая состоит в том, что все файловые системы и хранилища VSS навязывают единую иерархию. Уже приводился пример с поиском информации в предметной области — при этом сложно было вспомнить, что это была именно статья. Следовательно, должны быть другие измерения, в которых должна индексироваться информация — для простоты возьмём файлы и их описания, а также коллекции ссылок. Определить, что в каждом случае будет являться этими измерениями — пожалуй, и является наиболее сложным в процессе создания базы знаний. С первоначальной иерархией тоже стоит разобраться, хотя она вроде бы всем понятна.

Здесь, как выяснилось⁶ необходимо некоторое пояснение, что же представляют из себя измерения. Возьмём презентацию. Презентация была сделана в определённый день, определённой компании, конкретным людям. Компания, в свою очередь, относится к одной или нескольким горизонтальным областям, отраслям. Далее, презентация была подготовлена конкретными людьми. Всё перечисленное можно рассматривать как пространства (или измерения) в которых существует презентация. Не все измерения нам важны. Скорее, важны те, в которых мы мыслим, когда ищем информацию. Например, дата нам будет не очень важна, также как конкретные люди, которые смотрели презентацию. Вот компания и отрасль, для дальнейшей работы с этой компанией и этой отраслью вполне могут нас заинтересовать, поэтому их имеет смысл рассматривать как измерения для базы знаний. Аналогично презентации любой «кусочек знаний» можно рассматривать как координату в наборе пространств. Иерархия — первое из этих пространств, наиболее устоявшееся и привычное для нас.

Таким образом, в общем случае модель работы с базой знаний выглядит так: информация по-прежнему сначала помещается в некоторую первичную иерархию (заметьте, здесь достигается вещь, очень редкая для новых систем — *полная обратная совместимость*). Но вторым действием мы задаём координаты для этой же информации в других измерениях. В дальнейшем можно «вытащить» информацию по координате в любом измерении.

Приведём пример. Есть проект, на нём делаются доработки. Проект лежит в папке «коммерческие проекты» или «завершённые проекты» (кстати, почему бы информации не перемещаться в иерархии?), а доработки могут быть разными:

- Доработки под конкретного Заказчика, которые вряд ли кому-то ещё пригодятся;

⁶Спасибо отзывчивым читателям. Те, кто сталкивались с OLAP, могут совершенно спокойно этот раздел пропустить.

- Доработки, пригодные для использования на других предприятиях той же отрасли;
- Исправление ошибок, интересное почти всем;
- Доработки, пригодные для использования в некоторой горизонтальной области — например, дистрибуции.

Мы можем захотеть увидеть все доработки для конкретной отрасли, но нам совершенно не хочется перелопачивать все проектные папки.

Есть вариант копировать информацию, но это не выход. Хотя бы, помня о законе Мерфи — копия информации в разных местах рано или поздно начнут отличаться.

Теги⁷ не являются чертой wiki — насколько мне известно, они появились в блогах. Возникают две проблемы для адаптации идеи тегов к базе знаний:

1. Если страницу wiki приравнять к записи в блоге — то есть набором тегов отмечается одна страница — то что должно быть на этой странице?
2. Как сделать набор тегов одинаково удобным для всех? Ведь в блогах каждый создаёт свой набор.

Первый вопрос мы пока отложим. Некоторые наводки будут даны в разделах про развёртывание знаний в плоскую структуру и про здравый смысл при создании страниц. На второй вопрос ответить можно так:

1. Набор тегов должен включать все координаты иерархической структуры, так как она является первым измерением многомерного пространства, в котором мы организуем знания;
2. Набор тегов должен включать все координаты всех других пространств.
3. Набор тегов может быть расширен (фактически, добавляются новые измерения), если это удобно большинству.

Второй тезис подразумевает, что не стоит увлекаться детализацией пространств и добавлением новых измерений. Вся масса тегов будет в едином «облаке тегов», поэтому для сохранения удобства использования (непосредственно чтения) их должно быть ограниченное количество. Да, остаётся поиск на странице в браузере, но это уже не так удобно. Для примера: мы ограничились двумя дополнительными измерениями. Попытки добавить новые измерения не проходили тест «зачем?». Например, тег «тренинги» — зачем он? Кто в здравом уме захочет вытащить все тренинги, вне зависимости от предметной области или системы?

⁷Теги, метки, ярлыки — как предпочитаете. Ключевые слова или выражения, которыми, как правило, внизу, помечают свои записи авторы блогов. Например: «кино», «работа», «авиация», «музыка».

Меню с тегами и иерархической структурой можно сделать двумя меню. Позже я чаще стал замечать, что когда думаю, куда положить информацию, пользуюсь меню с иерархией, а когда ищу информацию — меню с тегами. Первичная иерархия обязательно должна быть, повторюсь — полная обратная совместимость крайне полезна для завоевания сторонников. Технически движок wiki позволяет, конечно, не пользоваться иерархией вообще, но это, с моей точки зрения, скорее вредно, поэтому об этой возможности я никому не рассказывал.

Бывает, что и теги не помогают — тогда помогает возможность текстового поиска. Здесь всё зависит от полноты описания файлов и ссылок.

Так от организации базы знаний мы постепенно переходим к функциям работы с ней.

2.3 Встраивание функций работы с базой знаний в существующие процессы

С одной стороны, работу с базой знаний можно представить как процессы наполнения и чтения, которые запускаются по определённым событиям. Именно так и составлялась документация к базе знаний департамента, но позже я пришёл к выводу, что более правильно считать наполнение и чтение базы знаний функциями. Эти функции необходимо встроить в существующие процессы организации. Тогда, после изменения должностных инструкций, работа с базой знаний будет регулярной и устраняется риск, описанный выше — если кто-то не хочет делиться информацией с другими.

Первый подход лучше тем, что не требует наличия схем существующих процессов в организации, что довольно типично для российских предприятий (как мне недавно сказали, и для европейских предприятий тоже). Да и должностные инструкции есть не везде.

С другой стороны, появляются и новые процессы, которые поддерживаются базой знаний. Например, работа центров компетенции по отрасли.

Первоначальное наполнение базы знаний целесообразно планировать, как и любую проектную деятельность (и так же контролировать).

Ну и «последний писк моды» — установка сотрудникам КРІ по их вкладу в базу знаний.

Изначально были выделены роли экспертов — тех, кто наполняет базу знаний. На самом деле, это было сделано для управляемости и чтобы можно было быстро подкорректировать что-то «на ходу». В целом, с моей точки зрения, роль эксперта — не в особых правах на создание контента, а в объёме созданного полезного контента, направлении других в этом. Права на запись (кроме удаления файлов) можно раздать всем — об этом чуть позже в разделе подходов к наполнению базы знаний.

При подключении новых людей, которые наполняют базу знаний, их желательно снабдить некоторыми инструкциями. Здесь речь не о технических инструкциях, а скорее о рекомендациях, как стоит работать с базой знаний, чтобы использовать преимущества именно wiki. Часть из этих рекомендаций приводится ниже.

2.4 Развёртывание знаний в плоскую структуру

После долгой работы с файловой системой сам себя я ловил на том, что воспроизвожу эту же структуру на страницах wiki — с её совершенно неудобным и избыточным для wiki количеством уровней вложенности. На страницах wiki можно разворачивать иерархическую структуру, используя не ссылки на новые разделы, а заголовки разного уровня на той же странице.

Конечно, есть определённые ограничения:

- Страница должна соответствовать определённому набору тегов. Если на странице скапливается большой объём разнородной информации, количество тегов возрастает и в целом их использование несколько теряет смысл. Хотя теги не обязательно ставить внизу страницы — можно их ставить в конкретном разделе, к которому они относятся.
- Страница должна оставаться пригодной для использования. Страницы длиной в 2-3 экрана не являются проблемой, но здесь также важно, чтобы разделы были примерно одинакового размера.

В целом, wiki позволяет знаниям искать форму. Если есть страница, на которой приведено большое количество информации, при этом больше половины занимает один раздел — возможно, его стоит перенести на отдельную страницу.

Предупреждения приведены скорее для того, чтобы рекомендация по разворачиванию структуры на одну страницу не воспринималась как абсолютная истина. При использовании wiki всегда есть, как минимум, два средства работы с очень большими страницами:

1. В браузер встроено поиск, не забывайте об этом.
2. Wiki, как правило, создают содержание в начале страницы по названию заголовков разного уровня.

И вот, пользователи базы знаний стали очень довольны, но в какой-то момент им перестаёт хватать возможностей приложения. Здесь возникает вопрос, как можно это приложение расширить.

2.5 Расширяемость

Во многих приложениях, не только в dokuwiki, реализована архитектура, позволяющая независимым разработчикам создавать плагины. Чем популярнее приложение, тем больше шансов, что у кого-то возникала потребность, схожая с вашей, и был создан соответствующий плагин. Именно поэтому я за то, чтобы брать приложения из mainstream. Есть и другой подход, когда в системе изначально закладывается значительный функционал. Например, система Microsoft Sharepoint Portal подходит не только для базы знаний, но и намного большего — организации корпоративного документооборота. То, что это коммерческое приложение, с определённой точки зрения обязывает иметь значительный базовый функционал, который не всегда востребован. С другой стороны, есть wiki с лицензией

GPL (а разработчики заинтересованы в развитии решений, прежде всего, с такой лицензией), с узко специализированным базовым функционалом и большим количеством плагинов. Например, есть плагин для тегов и создания облака тегов (наиболее часто используемые теги выделяются более крупным шрифтом), загрузки файлов архивом (и распаковки на сервере). Да, для Sharepoint также есть плагины. Но это другой рынок — коммерческих систем, здесь за всё принято брать деньги. Тот же плагин тегов для него, возможно, где-то существует, либо вам предложат его написать из расчёта некоторой стоимости человеко-часа разработчика.

Для примера, сходу в Гугле можно найти каталог из 60 плагинов для Sharepoint, тогда как для dokuwiki на сайте разработчика доступно более 300. Причём Microsoft не удосуживается поддержанием каталога плагинов на своём сайте — ведь это не их заработок.

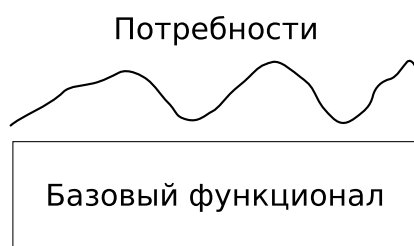


Рис. 2: Система с обширным базовым функционалом

Можно изобразить графически условные потребности конкретной организации и базовую систему, которая без плагинов реализует некоторый значительный объём функционала. Эта линия у всех организаций уникальна, расстояние между двумя линиями показывает, насколько система устраивает конкретную организацию.

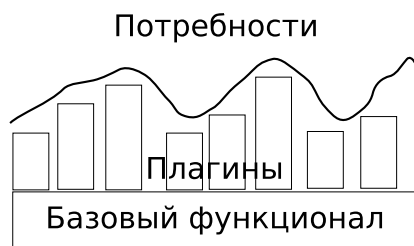


Рис. 3: Система с небольшим базовым функционалом и значительным количеством плагинов

С другой стороны, можно взять систему с не очень значительным базовым

функционалом (прямоугольник в основании намного меньше, чем на предыдущем рисунке), но с помощью плагинов система «доводится» до установленных требований. Именно такое ощущение у меня возникло при работе с dokuwiki, причём мы уже устали выдумывать новые потребности, а неосвоенных плагинов осталось ещё много.

2.6 Простота обучения

База знаний — относительно небольшая система, но её внедрению также нужно уделять внимание. Любое внедрение подразумевает обучение. С моей точки зрения, достаточное количество обучения — это:

1. Общая презентация для всех, где подробно рассказывается о возможностях системы, показывается в системе же, как с ней работать.
2. Ссылка на главной странице системы на документацию на русском.
3. Краткое вводное обучение для любого нового пользователя (с доступом на запись), которое включает основные моменты по редактированию страниц, рекомендации по представлению информации, ссылка на playground, где можно потренироваться.
4. Контроль в течение некоторого времени за тем, что пользователь создаёт и рекомендации по более эффективным приёмам работы и представления информации.

Сам синтаксис wiki делает обучение простым, но есть несколько тонких моментов. Мне хотелось полного контроля над кодом, поэтому мы отказались от визуальных редакторов. Также иногда они портили код. Как ни странно, редактирование кода не вызвало проблем, тогда как разбираться в программных ошибках WYSIWYG-редакторов у нас не было бы никакой возможности.

Отмечу, что из виденного мной dokuwiki оказалось единственным приложением, в котором код может вставляться по нажатию кнопки. Этот нехитрый приём сильно облегчает ввод wiki-кода, но больше нигде я такого не видел.

С другой стороны, нужно понимать, что использование базы знаний подразумевает некоторый интеллект. Поэтому есть разумная причина не делать всё «для совсем тупых», как меня однажды попросили. Но веб-доступ, как к обычному сайту, возможен, безусловно, для всех — ведь это почти обычный веб-сайт, к которому возможен удалённый доступ.

2.7 Простота доступа к приложению

С одной стороны, есть веб-доступ, с которым, казалось бы, всё понятно. Известны недостатки: медленное время реакции, разрыв сессии, трудно обрабатывать кнопку «назад» в браузере и т. д. С другой стороны, всё несколько не так. Продолжая сравнение с Sharepoint, в нём соответствие стандартам (W3C) появилось

относительно недавно. В целом, часто с продуктами Microsoft можно столкнуться с ситуацией, когда это не веб-приложения, хотя они могут использовать HTTP, HTML, CSS и другие стандарты. Microsoft неоднократно уличали в том, что эти стандарты оказывались неожиданно для всех расширены и поддерживали их с двух сторон (клиент и сервер) только продукты Microsoft (см. рис. 4).

Несколько сложнее выглядит схема с ориентацией и клиента, и сервера на независимые стандарты. Она выглядит сложнее, но только она может защитить инвестиции (время людей, наполняющих базу знаний — это тоже инвестиции). Кроме всего прочего, это облегчает настройку клиентов — не нужно разрешать элементы управления ActiveX, создавать защищённые зоны и т. п.

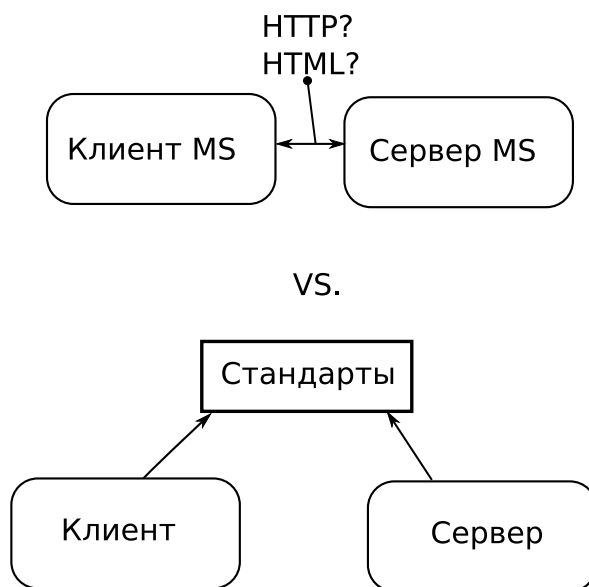


Рис. 4: Система вендора vs. ориентация на независимые стандарты

Многим, конечно, хотелось бы получить доступ к базе знаний чужой компании. Здесь может помочь openssl и шифрование хотя бы страниц, на которых может передаваться конфиденциальная информация. Если в базе знаний вся информация является конфиденциальной, можно шифровать и весь трафик.

В целом технических моментов при организации базы знаний возникает немного. Это приложение чем-то похоже на социальную сеть, поэтому внимание необходимо уделить социальным и даже эмоциональным аспектам работы людей в базе знаний. К этим аспектам мы и перейдём, хотя они не менее важны, чем всё описанное выше.

3 Подходы к наполнению базы знаний

Далее я перечислил некоторые подходы, которые могут значительно повысить эффективность работы такого зависящего от конкретных людей приложения, как база знаний. Каюсь, не всегда им сам следовал — часто озарение приходило позже.

3.1 Демократия

Уже описывалась модель с выделением ролей «эксперты» и «читатели». Эта модель — не такая уж плохая. Она позволяет выделить инициативных людей, у которых есть предложения по улучшению базы знаний, но нет прав на запись. Таким людям я сразу давал права на изменение всех разделов. После шока, однако, немногие вносили реальный вклад.

Вообще, я бы сделал модель «все могут править всё» изначально. Это хороший мотивирующий фактор, почему бы его не использовать? Во-первых, это некоторые полномочия. Во-вторых, возможность сделать что-то реальное и полезное, что все увидят, к тому же. Были предложения сделать систему рейтинговых оценок материалов, которые каждый выкладывает, а там недалеко и до материального стимулирования выдающихся авторов. Возможно, это стоит попробовать, но мне кажется, что это создаст нездоровую конкуренцию и отменит один из принципов базы знаний — коллективную работу над контентом, что способствует постоянному улучшению качества информации.

3.2 Здравый смысл

Wiki часто не предполагает единого способа реализовать то, что вы хотите сделать. Иногда можно сделать одно и то же в виде одной страницы с разделами, можно в виде списка со ссылками на страницы, можно в виде таблицы.

По привычке всё делать на работе, предварительно обдумав, мы устраивали какие-то дискуссии. Потом я просто стал говорить так: «Делай, как считаешь нужным. Если большинству будет удобно, оставим. Если нет — я смогу вернуть старую версию». Иногда затраты на споры выше, чем возможный эффект от этих споров. Это вопрос не только здравого смысла, но и управления амбициями, готовности уступить. Потому что база знаний, повторюсь — приложение *коллективной* работы.

3.3 Иерархия

Использование такого инструмента, как wiki, часто способствует формированию «кучи» из страниц, без разделов и подразделов — просто формируется ссылка на страницу, которая нужна, потом эта страница создаётся. Именно так я и поступаю в своих локальных базах знаний, но при больших объёмах данных такая база может стать неуправляемой. Кроме того, в *dokuwiki* при помощи разделов (папок

или областей имён) осуществляется разделение доступа на страницы, поэтому отходить от иерархической структуры совсем — с моей точки зрения, опасно.

По сути, база знаний нашего департамента существовала именно в виде структуры — вне зависимости от инструментов. Это — скелет, к которому крепится всё остальное. Пройдя путь от стесняющих творчество инструментов, через многомерные пространства и полный доступ всех для коллективной работы, разумно вернуться вновь к иерархии и поддерживать дисциплину, потому что и иерархия — тоже удобно.

4 Карты Freemind и интеграция с приложениями

Есть ещё несколько моментов, повышающих удобство представления информации в базе знаний, если используется dokuwiki. Инструмент Freemind позволяет создавать ассоциативные карты — они используются для решения трудно формализуемых задач. В dokuwiki ряд плагинов позволяет представить такие карты в виде flash-объектов, которые можно масштабировать, сворачивать и разворачивать узлы и ветви. Такие инструменты позволяют подтолкнуть и других к новым идеям. И ещё это первое полезное приложение технологии flash из всех, что я видел.

Разработаны плагины для интеграции с форумами, CMS и даже аутентификации через LDAP. Это позволяет создавать именно интегрированные решения, а не набор разрозненных, пусть и полезных приложений. Мне кажется, это то, к чему стоит стремиться.

5 Выводы

Не всегда требования успевают за появлением новых технических возможностей, но основа, по крайней мере, базы знаний департамента, о которой я писал, была заложена, когда мы обсуждали нарисованную на одном листе А4 иерархическую схему — я отчётливо запомнил этот момент. Пожалуй, главное достижение технологий в этом смысле, что можно быть от них независимыми, свободно реализуя новые творческие идеи. А для начала одного листа А4 вполне достаточно. Главное, чтобы было, что на нём записать.