Тема 2
«Инсталляционное тестирование»

Варианты работы инсталлятора, стандартные проверки

Вспомним определение

Инсталляционное тестирование (installation testing) – тестирование, направленное на выявление дефектов программного обеспечения, влияющих на протекание стадии инсталляции (установки) и деинсталляции (удаления) приложения.

Напутствие

Многие (особенно – начинающие) разработчики напрочь забывают о том, что их программное средство будет работать не только на том компьютере, на котором оно разрабатывалось.

И начинается: не та версия того или иного ПО, отсутствуют какие-то библиотеки, где-то жёстко прописаны пути и т.п.

Пути

Ещё на стадии разработки следует принять в качестве стандарта использование только относительных путей с префиксами, позволяющими прекратить их в абсолютные. Т.е. вместо

```
$path1="c:/www/images/";
$path2="/images/";
$path3="../images/";
```

следует писать

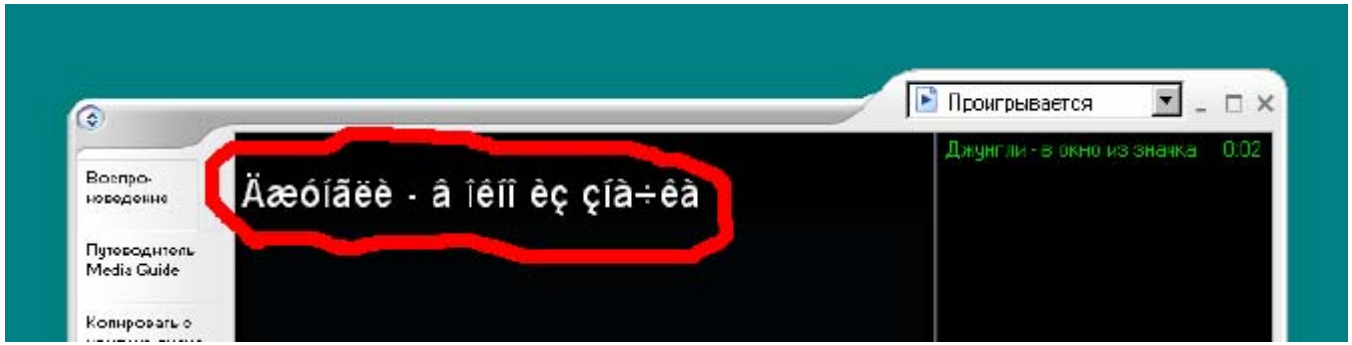
```
$path=$path_prefix."images/";
```

где $path_prefix задаётся в процессе инсталляции (можно попробовать определить его автоматически на основе данных из массива $_SERVER).

Пути

Второе магическое правило: только нижний регистр, только латиница (буквы, цифры, знак подчёркивания). **Без исключений.**

Это сразу отметает проблему "под виндой всё работает, а под никсами – нет". Знак подчёркивания вместо тире – хороший способ не путать на подсознательном уровне пути и арифметические выражения.



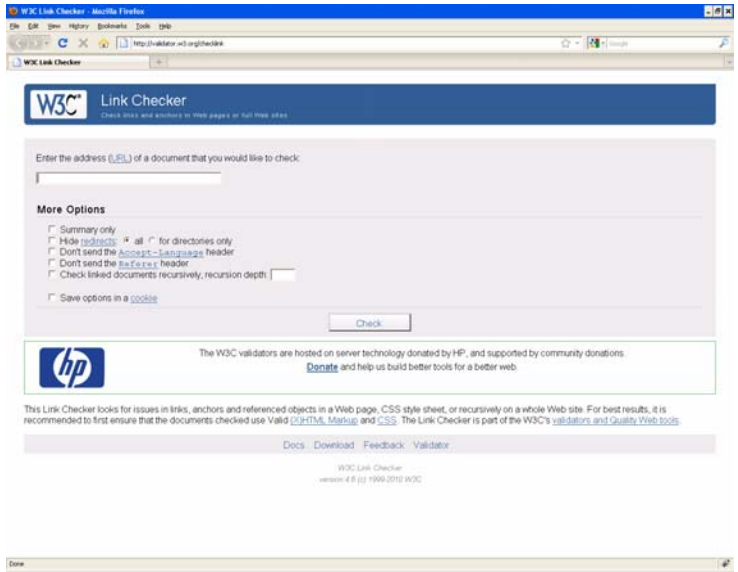
Пути – технология решения

Проверить корректность путей помогает анализ на наличие битых ссылок и общее функциональное тестирование продукта.

Проверить ссылки – это святое. Используем софт:

- Semonitor;
- онлайновый валидатор (особенно удобно, когда сайт уже выгружен на хостинг) Link Checker);
- Xenu's Link Sleuth

Подобных утилит и сервисов много.



Инсталлятор – ДОЛЖЕН БЫТЬ!

Второе, о чём следует сказать, – необходимость наличия хоть какого-нибудь инсталлятора.

Даже если ваше приложение простое, как кирпич, можно написать скрипт, который проверит, всё ли хорошо.

Инсталлятор – что проверяем

Наличие и размер (можно и md5-хэш) всех входящих в инсталляционный комплект файлов. Часто можно встретить ситуацию, когда что-то не докачалось, повредилось и т.п. Эту информацию можно хранить в явном виде в массиве.

Наличие свободного места на диске. Оценку проводим по двум параметрам:

а) Хватит ли места на завершение инсталляции (это – обязательно, если в процессе инсталляции мы создаём новые файлы).

б) Достаточно ли свободного места для устойчивой работы программного средства хотя бы в ближайшее время. Так, например, глупо ждать светлого будущего от фото-галереи, которой осталось доступно 10 Мб дискового пространства.

Инсталлятор – что проверяем

Ограничения:

- доступного объёма памяти;
- времени выполнения;
- и т.п.

Версии:

- сервера приложений;
- расширений и библиотек;
- СУБД;
- веб-сервера;
- операционной системы.

Инсталлятор – что проверяем

Возможность установки исходящих соединений (если они необходимы приложению). Достаточно просто открыть сокет функцией fsockpoen().

Значение критических для безопасности и производительности настроек сервера приложений, СУБД и т.п.

Инсталлятор – что проверяем

Всё или почти всё вышеперечисленное можно адаптировать для самодиагностики приложения на случай "всё работало и вдруг перестало".

Если проект предполагает установку и настройку разработчиком, здорово облегчить себе жизнь можно с применением модульного тестирования.

Варианты работы инсталлятора

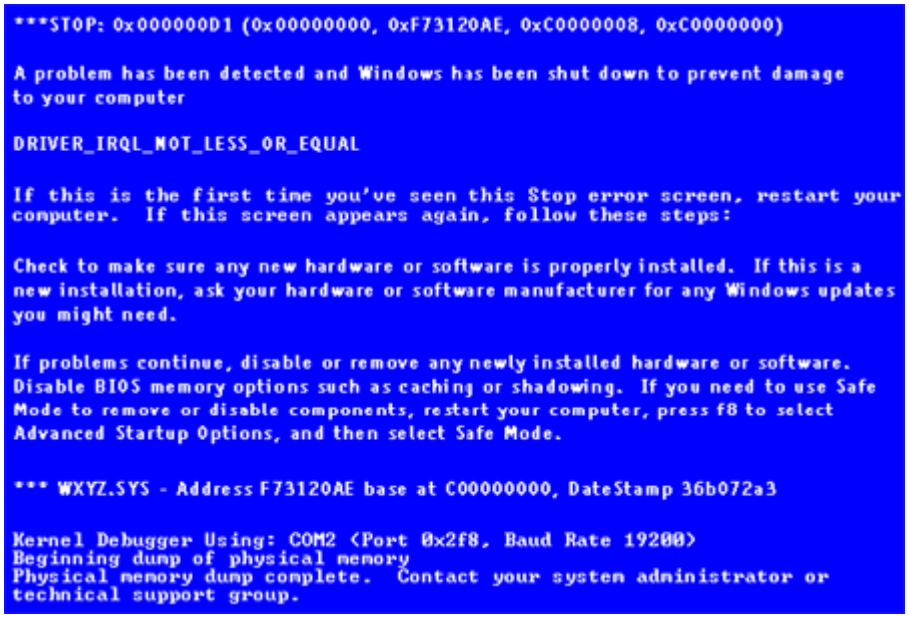
Новая среда исполнения, в которой приложение ранее не было инсталлировано – сюда относятся все только что рассмотренные идеи и проверки.

Обновление существующей версии ("апгрейд") или изменение текущей версии на более старую ("даунгрейд") – особое внимание следует уделить вопросу сохранности данных.

Следует также настойчиво предложить пользователю сделать резервную копию базы данных, файлов данных и файлов приложения (можно даже автоматизировать этот процесс).

Варианты работы инсталлятора

Повторная установка приложения с целью устранения возникших проблем ("переинсталляция") – приложение должно в такой ситуации информировать пользователя об альтернативных вариантах решения проблем, предлагать сделать резервную копию и НЕ удалять данные пользователя ни из базы данных, ни с диска.



Варианты работы инсталлятора

Повторный запуск инсталляции после фатальной ошибки, приведшей к невозможности продолжения инсталляции.

Возникновение такой ситуации – плохой симптом. Но, допустим, это случилось по объективным причинам (например, в результате сбоя аппаратного обеспечения).

Тогда приложение должно быть в состоянии или продолжить инсталляцию с того места, где она была прервана, или начать её заново, "аннулировав" все предыдущие действия (удалив из базы данных и с диска свои "старые" файлы и данные).

Варианты работы инсталлятора

Удаление приложения.

Приложение должно ПОЛНОСТЬЮ удалить себя из системы.

Существует золотое правило: "система до установки приложения и после его удаления должна выглядеть совершенно одинаково". Это значит, что нельзя допустить "забывания" приложением каких-то своих файлов или данных в БД.

Варианты работы инсталлятора

Установка нового приложения из семейства приложений (например, мы выпускам форум и блог).

Эта ситуация опасна тем, что одно приложение может повредить данные другого (перезаписать своими в процессе инсталляции или, посчитав своими, удалить в процессе деинсталляции).

Варианты работы инсталлятора

Ещё раз, одним списком:

- новая среда исполнения, в которой приложение ранее не было инсталлировано;
- обновление существующей версии ("апгрейд");
- изменение текущей версии на более старую ("даунгрейд");
- повторная установка приложения с целью устранения возникших проблем ("переинсталляция");
- повторный запуск инсталляции после фатальной ошибки, приведшей к невозможности продолжения инсталляции;
- удаление приложения;
- установка нового приложения из семейства приложений (например, мы выпускам форум и блог).

Обновление и модификация ПО: внедрение в код, проблемы "хаков", "фиксов", обновлений БД

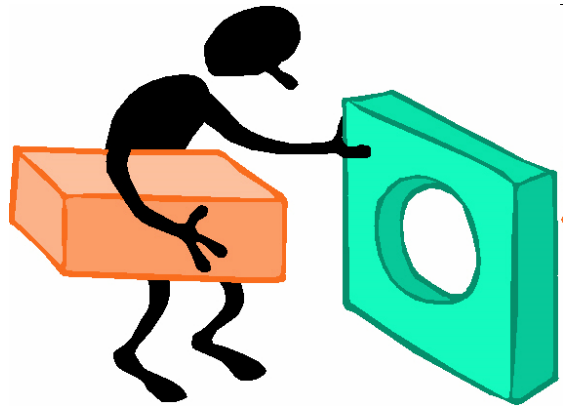
Обновление и модификация ПО

Важной особенностью веб-ориентированных приложений является необходимость их постоянного обновления в целях устранения известных уязвимостей и расширения набора функций.

Обновление и модификация ПО

С этим связаны следующие проблемы:

- установленные дополнения могут оказаться несовместимы с новой версией;

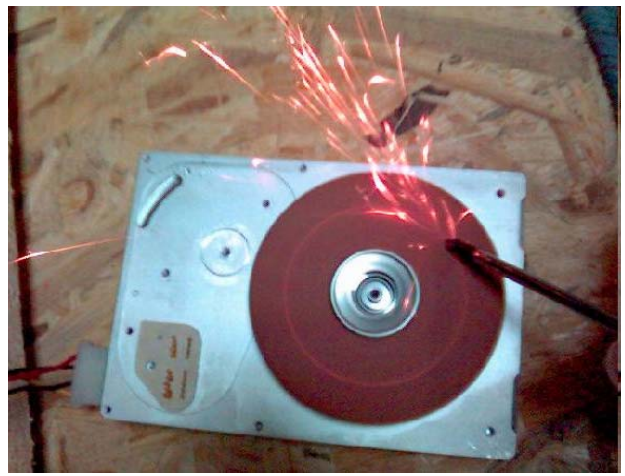


- установленные дополнения могут подвергать угрозе безопасность и даже работоспособность ПО;

Обновление и модификация ПО

- ПО может не поддерживать установку обновлений как таковых, а допускать лишь модификацию кода;

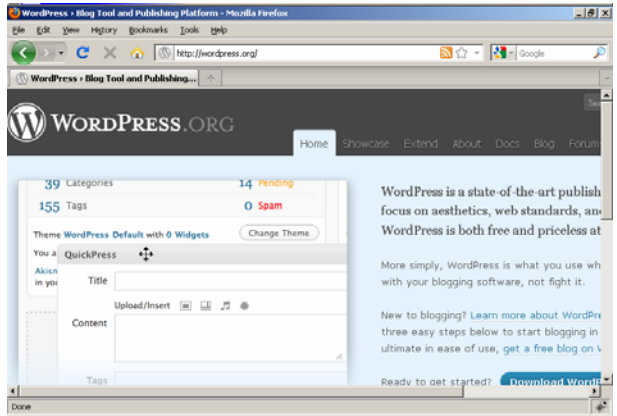
- любые добавления/удаления дополнений или обновления текущей версии могут затрагивать БД, что само по себе может стать источником проблем.



Два примера

Для наглядности рассмотрим два примера очень распространённых и популярных веб-ориентированных приложения:

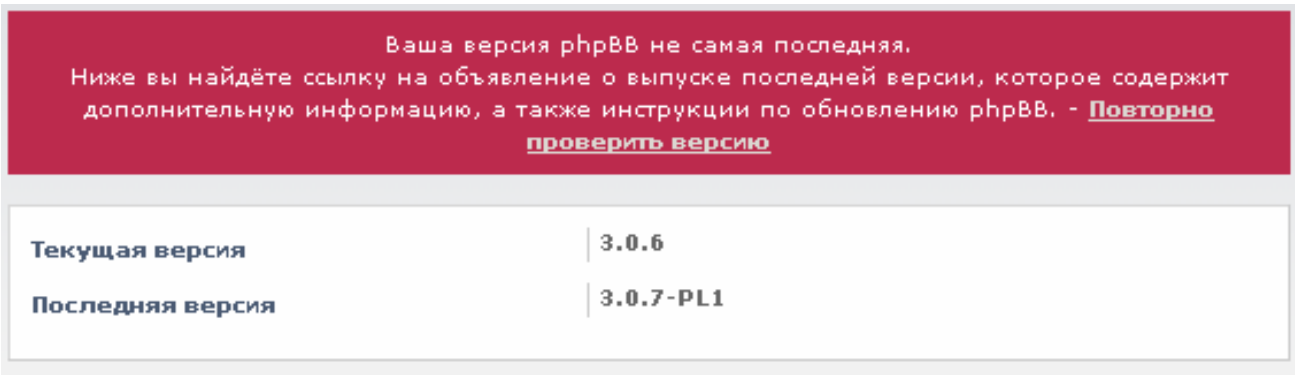
- движок форума phpbb;
- движок блога wordpress.



Два примера: phpbb

phpbb поддерживает автоматическую проверку актуальности установленной версии и возможность полуавтоматического обновления.

Но! phpbb НЕ поддерживает модули (плагины), т.е. все дополнения к данному форуму устанавливаются в виде т.н. “хаков” (hacks), и после установки обновления, как правило, требуют повторной установки, если модифицированные администратором файлы были заменены инсталлятором в процессе обновления.



Два примера: phpbb

Какие у этой ситуации плюсы и минусы?

Минусы очевидны:

- про полуавтоматическое обновление приходится забыть, т.к. намного разумнее производить его вручную, отслеживая, какие файлы приходится заменять;
- после каждого обновления приходится заново устанавливать все “хаки” и проверять их работоспособность.

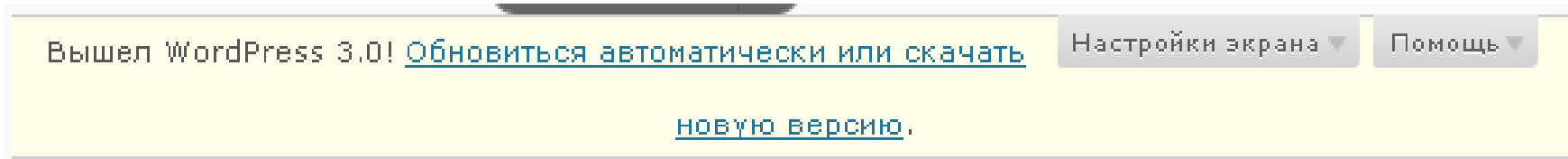
А **плюсы**?

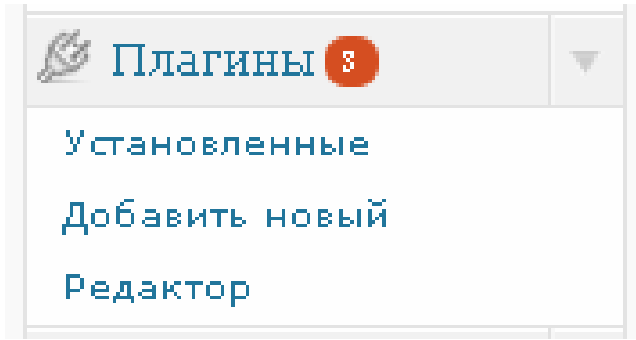
- администратор действует осознанно – исключена ситуация «кликнул по кнопке, опомнился, но уже было поздно»;
- начинающие администраторы, чья квалификация крайне далека от идеала, не смогут «в один клик мышью» установить некое дополнение, логику и последствия работы которого они не понимают.

Два примера: wordpress

wordpress поддерживает автоматическую проверку актуальности установленной версии и возможность полностью автоматического обновления.

При этом wordpress поддерживает дополнения в виде плагинов (plugins), по которым также умеет вести контроль актуальности версий и (если это поддерживает сам плагин) осуществлять автоматическое обновление.





Два примера: wordpress

А у этой ситуации какие плюсы и минусы?

Плюсы:

- удобно;
- нет необходимости совершать лишние действия.

Минусы:

- если что-то перестало работать, диагностика усложняется в десятки раз;
- начинающий администратор может угробить блог одним кликом мышки.

Два примера: вывод

Если вы разрабатываете программное средство:

- обеспечьте автоматический контроль появления новых версий;
- предоставьте пользователю на выбор:
  - автоматически обновить версию;
  - скачать только изменённые файлы и скрипты для обновления БД;
  - скачать новую версию целиком.

Если вы – пользователь (администратор) программного средства:

- думайте, думайте и ещё раз думайте – автоматизация при всей своей красоте иногда создаёт много проблем;
- перед обновлением ОБЯЗАТЕЛЬНО сделайте полную резервную копию приложения и БД (а ещё лучше – всего аккаунта на сервере).
- после обновления несколько дней особенно внимательно следите за работой приложения – могут быть сюрпризы.

Тест для закрепления материала

1. Какие рекомендации вы можете дать разработчикам веб-ориентированных приложений по формированию путей к файлам?
2. Какие параметры среды должен проверять инсталлятор?
3. Какие варианты работы инсталлятора вы можете назвать?
4. Какие проблемы могут возникнуть в процессе или в результате инсталляции обновлений или дополнений к программному обеспечению?
5. Перечислите плюсы и минусы архитектуры приложения, не поддерживающей дополнения.
6. Перечислите плюсы и минусы архитектуры приложения, поддерживающей дополнения.
7. В чём особенности инсталляционного тестирования веб-ориентированных приложений в сравнении с инсталляционным тестированием классических «настольных» приложений?
8. В чём заключаются отличия инсталляционного тестирования веб-ориентированных приложений, предназначенных для конкретного заказчика и предназначенных для широкого круга пользователей.
9. Какие проблемы в работе инсталлятора могут возникнуть в случае, если он построен на пошаговых формах?
10. Какие функции инсталлятора можно использовать в последующей работе приложения для самодиагностики?