



# Надёжное ПО в производственной логистике

White Paper

5 ноября 2009 г.

**Copyright © 2009, XR Systems GmbH**

Этот документ является информационным. Печать и хранение разрешены в полной, неизменной форме. Дальнейшие публикации, перепечатка и тиражирование - в любой форме, полностью или частично - возможны только с предварительного письменного согласия фирмы XR Systems GmbH.

В данном документе наряду с собственными данными, пояснениями, оценками содержатся также описания программных продуктов и их концепций, основывающиеся на соответствующих публикациях производителей этих продуктов.

Использование нами в этом документе фирменных марок, товарных знаков и т.п. не говорит о том, что упомянутые знаки и марки предоставлены для свободного использования кем бы то ни было в дальнейшем. Все торговые марки и названия продуктов являются собственностью их владельцев .

В данном документе также приведены ссылки на страницы в Интернете. Мы категорически заверяем, что никакой ответственности за содержание этих страниц не несем.

Мы дистанцируемся от содержимого упомянутых в этом документе Интернет – страниц. Это заявление относится ко всем приведенным ссылкам.



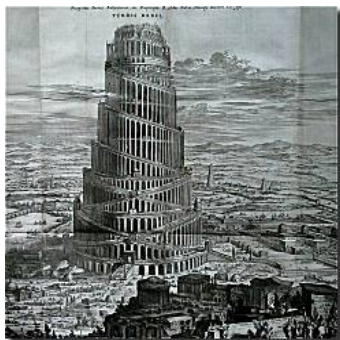
XR Systems GmbH  
Münchener Str. 14  
D-85540 Haar  
fon +49 (89) 427231-0  
fax +49 (89) 427231-70  
net [www.xrsystems.de](http://www.xrsystems.de)

---

---

# Надежное ПО в производственной логистике

## Предисловие



Каждый IT-проект, имеющий успех в течение продолжительного времени, основывается на тщательно выверенном сочетании аппаратного и программного обеспечения, а также человеческих ресурсов, используемых для развития и технического обслуживания проекта. Технические возможности современного аппаратного обеспечения вводят в искушение "церемониймейстеров" сферы программного обеспечения, побуждая их проектировать соборы непомерной тяжести в вакуумном пространстве. Увлеченные моделями и теориями, они утрачивают здравый взгляд на решение проблем и преклоняются перед возможностью поставить галочку «сделано» напротив какого-нибудь [модного слова](#).

Бесконечная гибкость и податливость программного обеспечения поддерживает иллюзию того, что все осуществимо, без оглядки на надежность и удобство обслуживания. Сегодня прежде всего недооценка ноу-хау, необходимого в течение долгого времени, приводит многие проекты еще до финансового банкротства к банкротству интеллектуальному. Как и непрозрачная продукция финансового рынка, вызвавшая нынешний финансовый кризис, - [Уоррен Баффет](#) при этом говорит об оружии массового поражения - сомнительные стратегии в области программного обеспечения отнимают деньги и время в неисчислимом количестве проектов.

Уже более 20 лет DBtron/L является одной из [надежнейших](#) систем для ведения складского учета и учета движения материалов ведущего мирового производителя. Регулярная безостановочная продолжительность эксплуатации системы превышает 1000 дней, и лишь для проведения плановых профилактических работ эксплуатация системы была приостановлена.

Предлагаемая [White Paper](#) излагает основы стратегических основ высокого производственного качества и рассчитана для пользователей объемных автоматизированных вычислительных систем.

Она описывает технические компоненты системы и основные стратегические принципы, которые применяются для Dbtron/L в течение продолжительного времени. В PDF-версии выделенные голубым цветом слова содержат ссылки на дополнительную информацию. В разделе Download нашей [Интернет-страницы](#) вы найдете PDF на разных языках.

---

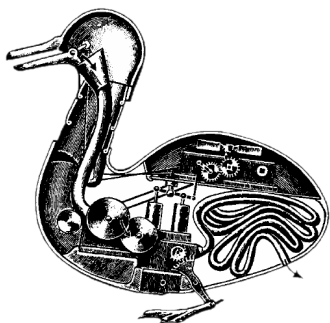
---

Представленные здесь описания мы старались сделать предельно простыми, предоставляющими Вам легкий, интуитивный подход к теме. Будем благодарны за любые предложения по улучшению текста.

### Концентрация на существенном

Высокое эксплуатационное качество DBtron/L является следствием одного-единственного решающего основополагающего принципа, который, как Мантра, красной нитью проходит через концепцию нашей системы:

*Преодоление сложности  
концентрацией на существенном*



"Лень является движущей силой прогресса", - гласит известная поговорка. Она и является - просим понять нас правильно - основой надежности нашей системы. Несомненно необходима еще и тщательность реализации для создания стабильной безошибочной системы. Но что принесут все старания в борьбе против ветряной мельницы под названием «Сложность»? Это просто наивно – бороться с огромным многообразием вещей. Особенно, когда в этом нет необходимости и когда речь идет о высоко-сложных вычислительных системах.

Разумно поступает тот, кто о сложности знает и управляет ею при помощи соответствующего менеджмента. Это не означает слепо следовать правилу *Лучше-меньше-да-лучше*. Необходимо принимать во внимание постоянную спутницу сложности – гибкость.

Гибкость IT-проекта зависит от места его последующего внедрения. Так, приложения, настроенные под конкретного клиента, будут заметно ограничены по сравнению со стандартными, рассчитанными на широкий рынок решениями.

DBtron/L создавался для долгоживущего мира производственной логистики и уже более двух десятилетий находится в эксплуатации у большого количества клиентов. Это позволяет сравнивать другие соображения и решения с системой для одноразового изменения почтовых индексов после воссоединения Германии.

Свойства системы, подтвержденные таким солидным сроком, включают в себя, помимо прочего:

- функциональные возможности системы
- продуманность
- надежность

- 
- 
- гибкость
  - стоимость приобретения (Make or Buy)
  - стоимость эксплуатации, включая сопровождение

Далее мы расскажем, какие ключевые компоненты входят в состав DBtron/L и почему, на наш взгляд, они представляют наилучший выбор.

---

---

---

---

# Технические компоненты системы

---

## База данных

С самого начала мы интегрировали в DBtron/L только одну, технологически передовую базу данных - Oracle.

По состоянию техники на сегодня в основу всякого объемного программного обеспечения принято брать систему управления базами данных (далее - СУБД<sup>1</sup>). При этом разработчики программного обеспечения должны в первую очередь решить – будут ли они в своей архитектуре поддерживать одну или несколько общепотребительных операционных систем.

В течение 40 лет своего развития благодаря растущим требованиям к ним СУБД стали действительно мощными и объемными. Несмотря на все попытки стандартизации, различий в системах в разы больше, чем сходств. Технический прогресс уже давно отменил всякую стандартизацию, точнее, свел ее к маленькому ядру функций СУБД.

В середине 80-х сложилась такая счастливая для нас ситуация, когда мы – маленькая энергичная команда программистов – благодаря одному мировому концерну деревообрабатывающей промышленности получили шанс заложить краеугольный камень нашей DBtron/L; при этом мы могли применить наши обширные знания ведущей – как тогда, так и сегодня – СУБД Oracle.

### СУБД середины 80-х годов

Рынок СУБД выглядел тогда совсем иначе, чем сегодня:

Несмотря на достаточное количество предлагаемых СУБД и большую конкуренцию между ними, уровень цен оставался высоким. К тому же СУБД требовали высококвалифицированный ИТ-персонал. Очень часто принималось стратегическое решение в пользу базы данных, действительной для всего предприятия.

Казалось разумным сделать программное обеспечение (ПО) совместимым с уже имеющимся у клиента парком СУБД. Это было, пожалуй, логично и жизненно важно для производителей ПО – в пике многочисленным дорогим системам оставаться как

---

1) Термины «база данных», «система базы данных», «сервер базы данных», СУБД являются в сфере ИТ взаимозаменяемыми и употребляются без существенных отличий.

---

---

можно нейтральнее - чтобы повысить реальные шансы на продажу.

### Иллюзия независимости

*Поддержка более, чем одной СУБД подразумевает последующие затраты, намного превышающие запланированные расходы.*

В своем стремлении к простому и понятному программному коду разработчик отказывается от полного объема функций СУБД за исключением узкого стандартного ядра; как следствие – он должен большую часть функциональности современной СУБД самостоятельно надстраивать. Результатом являются огромные затраты разработчика и/или привлечение разработок третьих лиц; и то, и другое отражается на общей стоимости программного комплекса (Total Cost of Ownership).

Существенными для DBtron/L свойствами СУБД являются, помимо стандартных, также:

- Разделение доступа в многопользовательском режиме
- Запуск и восстановление после системных сбоев
- Агрегирование данных для отчетов
- Архивирование данных и их изменений в целях последующего пошагового восстановления
- Пошаговое протоколирование изменений системы
- Характеристики системы (Скорость, производительность)

С каждым отказом от использования функций СУБД по новой возрастают зависимости, от которых, собственно, хотели избавиться; зависимости от ресурсов<sup>2</sup>, более дорогих и менее устойчивых, чем СУБД – лидер мирового рынка.

### Базы данных сегодня

Спустя два десятилетия с момента появления первой версии проекта Dbtron/L рынок СУБД полностью изменился:

Значимые<sup>3</sup> СУБД, оставшиеся от огромного числа конкурировавших, можно пересчитать по пальцам одной руки. Бесспорным лидером рынка является Oracle с удельным весом в 46%, что значительно больше, чем доля 3 следующих конкурентов, вместе взятых.

---

2) Включая разработчиков программного обеспечения

3) подходящие для разработки приложений на перспективу

---

---

При этом стоимость лицензий системы в зависимости от их типа <sup>4</sup> настолько снизилась, что стала практически незаметной в общей стоимости.

### СУБД Oracle - неотъемлемая часть DBtron/L

Эта концентрация на базе данных Oracle позволяет нам в полной мере использовать функциональные возможности столь мощной СУБД и постоянно совершенствовать наши знания в администрировании БД. Благодаря использованию инструментов, которые разработаны либо оптимизированы для Oracle, мы достигли максимальной продуктивности. Таким образом, клиент получает систему, которая долговременно и по приемлемой цене удовлетворяет его потребности.

В системах управления производством СУБД является несущей компонентой, ведь ей доверены важные данные предприятия. Поэтому принятие решения о выборе пригодной для долгосрочных сценариев СУБД – процесс достаточно ответственный. Инновационные циклы на протяжении двух десятилетий принесли разработчикам и пользователям DBtron/L много новых компьютерных и операционных систем, пользовательских интерфейсов, методов и языков программирования; и только СУБД Oracle, как остров спокойствия, пережила все изменения и сохранила данные и инвестиции пользователей.

Также и в будущем Oracle будет доступен в новых и в ходовых системах (см. Linux) и будет обеспечивать защиту инвестиций клиентов.

---

## Аппаратное обеспечение – лучше виртуально

Виртуальное аппаратное обеспечение помогает сберечь самый ценный ресурс предприятия: сотрудников. При этом снижаются расходы и повышается качество системы. Также и в аппаратном и системном обеспечении произошли разительные перемены. 20 лет назад СУБД ORACLE была доступна более, чем на 100 различных платформах (Hardware, ОС). За это время многие прежние неунифицированные системы сменились стандартизированным ландшафтом аппаратного и системного обеспечения<sup>5</sup>.

### Классическое аппаратное обеспечение

---

4) например, встроенная лицензия

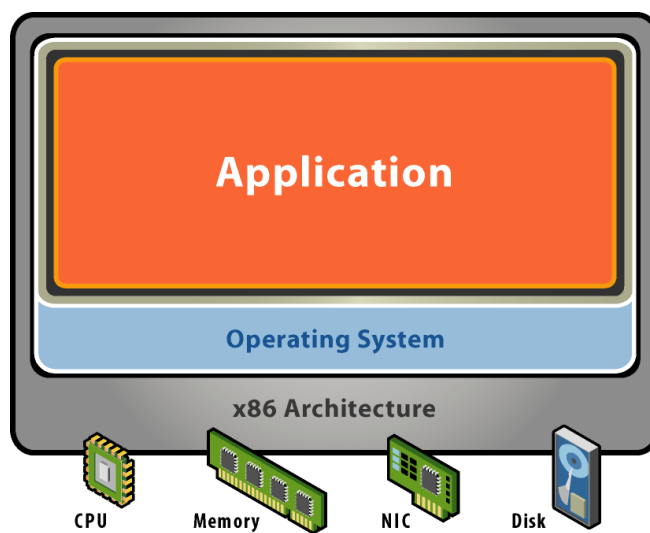
5) Лидером продаж являются системы на основе процессоров Intel и AMD (x86, x64).

---

---

Тем не менее, до сих пор очень накладно создать программный продукт, одинаково хорошо работающий под различными операционными системами. Поэтому многие программы ориентированы на определенную ОС или, как минимум, оптимизированы под нее.

Как результат – расплывшиеся серверы, чаще всего по одному на каждое приложение. Использование различных ОС в разы увеличивает потребность в квалифицированном обслуживающем персонале. Так при передаче клиенту IT-проекта разработчиком встает нелегкий вопрос о выборе подходящей или допустимой ОС.

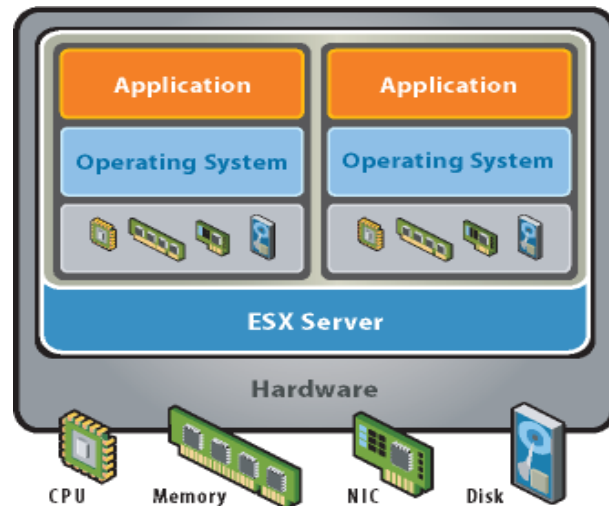


Тесная взаимосвязь между используемыми программами, операционной системой и имеющимся аппаратным обеспечением инсталляцию и конфигурирование очень затратными. И при каждом сбое сервера эти работы повторялись, что приводило к дальнейшему возрастанию затрат.

---

---

## Виртуальный компьютер



Продолжительный процесс IT-стандартизации венчается сегодня виртуализацией аппаратного обеспечения. [Виртуальность](#) - это такое свойство, когда несуществующая вещь проявляется в свойствах и воздействиях как существующая. Виртуальные компьютеры - или, иначе, [виртуальные машины](#) (VM) - состоят не из Hardware, а моделируются программно. *Виртуальные машины – это контролируемые, хорошо изолированные островки для запуска предварительно установленных и сконфигурированных приложений; они запакованы вместе с ОС (Гость) в запускаемый формат и готовы к запуску на одном из серверов (Хозяин), предварительно подготовленном при помощи специального виртуализирующего программного обеспечения.*

В этот контекст хорошо вписывается англ. термин [Virtual Appliance](#), который сравнивает комфортное использование с домашними электроприборами (англ. Home Appliance) : plug and play.

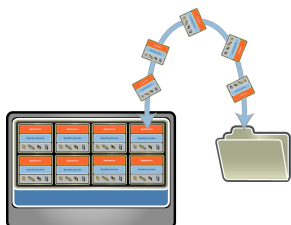
### Гибкость

***Виртуализация аппаратного обеспечения не зависит от знаний персонала об операционных системах.***

Внутри VM может находиться совсем другая ОС, чем «снаружи», а, точнее, чем внизу<sup>6</sup>. Виртуализирующее ПО моделирует внутри VM настолько стандартизированную среду (виртуальный процессор, виртуальные диски, виртуальные сетевые карты и

---

6) Как правило, программное и аппаратное обеспечение описывается снизу (физические носители) вверх (пользовательский интерфейс)



т.д.), что VM может быть легко скопирована или перенесена с одного сервера на другой.

VM запакована в небольшое<sup>7</sup> количество больших файлов. Благодаря резко возросшей мощности мобильных носителей информации и сетевой передачи данных стало возможным эффективно транспортировать уже сотни Гбайт.

Путем копирования файлов создается точная копия VM. В свете принципиальной независимости от физического сервера это дает целый ряд преимуществ для подготовки, сопровождения и безопасной эксплуатации программного обеспечения.

### Идеальное качество поставки

*Возможность создать полностью идентичную копию VM заменяет трудоемкие, дорогостоящие и подверженные ошибкам работы по инсталляции и конфигурированию очень простым технологическим процессом.*

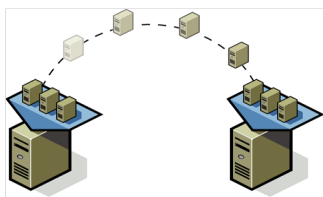
Как пользователь, Вы получаете в Ваше распоряжение 100%-точный образ VM, которая в течение долгого времени непрерывно совершенствовалась в нашем отделе разработки. Все улучшения, приобретенные в ходе эксплуатации или дальнейшей разработки, вносятся в референцируемую VM и полностью переносятся на каждый Ваш образ.

Сохранение стоимости умственной работы будет максимизировано и автоматизирована: будут предотвращаться конфигурационные ошибки. Никакое сопроводительное описание изменений и их последующее внесение не может обеспечить сравнимое качество.

### Быстрый перенос при сбое сервера

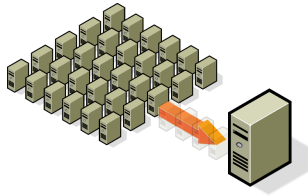
*Аппаратное обеспечение может и будет давать сбои; старение и износ, без сомнения, являются слабостью всех материальных вещей. Виртуализация аппаратного обеспечения поможет избежать сбоев в производстве или хотя бы сократить их.*

Современная технология виртуализации является значительной частью нашей стратегии. Как показано выше, виртуализирующее Software нивелирует различия между системами «Гость» и «Хозяин», а также между двумя различными «Хозяин»-системами



Мы производили уже перенос VM между двумя серверами, между которыми лежат различия многих лет разработок. Даже соответствующим образом оснащенный ноутбук при определенных обстоятельствах может стать временным пристанищем одной или нескольких VM. В зависимости от

7) В данном случае - в един-единственный



---

---

важности приложения и величины инвестируемого бюджета можно произвести перенос VM даже в работающем режиме.

### Консолидация

*Расходы на обслуживание сервера на протяжении всего срока его эксплуатации (стоимость владения) обычно превосходит его покупную стоимость. Виртуализация уменьшает аппаратный парк вычислительного центра.*

Как правило, мощности одного сервера хватает не на одну, а на несколько VM. Программные приложения, которые до сих пор были размещены на различных серверах только лишь из-за различных требований к «среде обитания» (ОС, версии и т.д.) и зачастую не загружали их полностью, теперь могут работать<sup>8</sup> на одном общем оборудовании.

### VMware - виртуализация для ПО предприятия

В случае, если виртуализация играет такую же решающую роль, как у нас в Dbtron/L, выбранное системное ПО должно удовлетворять строгим критериям.

[VMware](#) является пионером, который более 10 лет назад впервые технологию виртуализации из мира больших ЭВМ перевел на более дешевые стандартные системы. Являясь лидером на рынке технологий, VMware представляет широкую палитру продуктов и большой выбор областей их применения.

Все инсталляции Dbtron/L с 1999 базируются на VMware, с 2005 – на доступной для вычислительного центра версии ESX.

---

## Операционные системы Unix и Linux

*Системы Unix/Linux по-прежнему лидируют во всем, что касается надежности, кроме того, Linux предлагает лучшее соотношение цена/качество.*

[Unix](#) и его различные потомки признаны в профессиональном мире вполне надежными и производительными операционными системами. Вместе с соответствующим аппаратным обеспечением он, как и прежде, представляет достойный выбор для реализации ИТ-приложений предприятия. Все ранние инсталляции Dbtron/L базировались на нем<sup>9</sup>; многие работают до сих пор.

---

8) Аспекты изоляции и взаимного влияния дежат за рамками этой статьи. Мы готовы однако предоставить их более детальное освещение.

9) Hewlett-Packard HP-9000 с HP-UX



[Linux](#), «наследник» Unix'a, являясь для многих хорошей альтернативой как Unix'у, так и [Microsoft Windows Server](#), нашел свой путь в конфиденциальные сферы предприятия. С 1999 СУБД Oracle поддерживает Linux в своих новых версиях и развивает их на этой платформе.

Для нас это был эволюционный путь от Unix к Linux. Более 10 лет мы базируем все наши поставки DBtron/L на дистрибутиве [Red Hat Enterprise Linux](#), точнее на его производной [CentOS](#).

---

## ПЛК-управление без OPC

*Надежные материально-потокосые системы зависят от стабильности интерфейса ПЛК.*

Для сопряжения с [ПЛК](#)-системами на уровне управления мы заняли очень определенную позицию. Вместо того, чтобы поддерживать бесчисленные несовместимые фабрикаты<sup>10</sup>, мы концентрируемся на лидерах мирового рынка - Siemens (S5/S7), Allen-Bradley (PLC5/ControlLogix) и AEG Modicon.

Отвергая сложные, подверженные ошибкам компоненты на промежуточном уровне, DBtron/L коммуницирует напрямую с ПЛК.

Там, где конкуренты ставят на [OPC](#)-сервера как связующее звено, мы в DBtron/L используем относительно небольшую Software-библиотеку и обмениваемся данными с ПЛК наиболее кратким и стабильным путем.

---

10) Обозреватели мирового рынка насчитывают более 300 фабрикатов

---

---

## Принципы работы

---

### Бизнес-логика в СУБД

Более чем 20-летний опыт работы с Oracle показывает: СУБД является, как правило, наиболее подходящим местом для имплементации функций.

#### Универсальная среда программирования

Существенным преимуществом Oracle является не только безопасное и гибкое управление данными любого сорта, но и возможность сохранения программного кода<sup>11</sup> в базе данных и там же, в непосредственной близости от данных, его запуска. При этом возможности Oracle намного превышают возможности конкурентов, будучи предоставленными для полной имплементации производственной логики и разнообразного ПО предприятия.

В DBtron/L мы также храним большую часть программных кодов внутри БД. При каждой миграции СУБД на новую платформу — даже при изменении архитектуры процессора или ОС — безболезненно мигрируют и программные коды.

---

### Воспроизводимость

Для того, чтобы как можно лучше понять причины проблем, возникающих у клиента, мы воспроизводим в нашем отделе разработок конфигурацию, подобную клиентской. Раньше она включала в себя всю компьютерную систему вплоть до идентичной версии операционной системы и СУБД.

Сегодня благодаря виртуализации аппаратной части стало существенно легче и, главное, дешевле воссоздать подобную конфигурацию.

---

11) Написанном на [PL/SQL](#) или [Java](#)

---

---

---

## Ответственность

Мы отдаем себе отчет о значимости нашей IT-системы для каждого клиента. Все наши разработчики чувствуют высокую ответственность за безотказную работу системы, что проявляется при осуществлении поддержки. Мы устранили всяческие зазоры между разработкой и сопровождением, которые могли бы создать ненужные задержки в решении проблем клиентов.

---

---

## Заключение

[Парето-принцип](#), или Парето-эффект, или правило 80/20, гласит, что 80% результата достигается за 20% времени проекта. Оставшиеся 20% требуют большую часть усилий и являются причиной большинства ошибок.

Неизменно фокусируясь на существенном, тщательно взвешивая все затраты (также в области Know-How) и выгоды (гибкость, надежность), мы добиваемся оптимального решения для промышленно используемой вычислительной системы .