

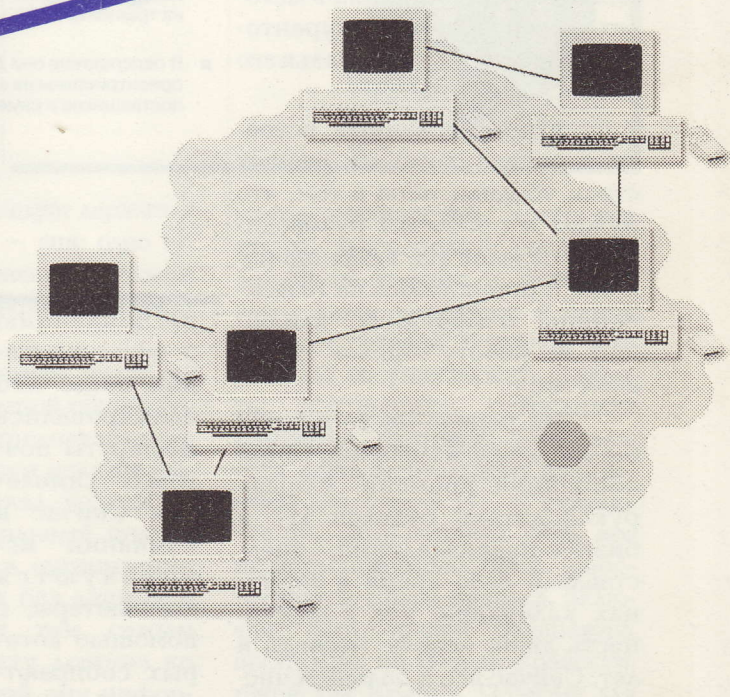
Jet

INFO

**МАТЕРИАЛ
НОМЕРА**

**Открытые сетевые
решения
1990-х годов**

**8
1996**





Открытые сетевые решения 1990-х годов

Запись выступления на семинаре в Доме ученых РАН, июнь 1995 г.

Я работаю в должности директора по информационным технологиям (ИТ) компании Sun Microsystems около 5 лет. В Соединенных Штатах это считается очень большим сроком, поскольку руководителям отделов ИТ приходится сталкиваться с проблемами

ле отраслей и стран растет понимание того, что сверяться нужно не с календарем, а с часами, и вести отсчет времени по минутам и даже секундам.

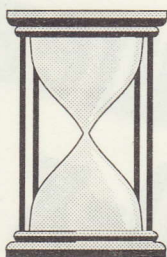
В качестве примера можно привести Federal Express — крупнейшую американскую транспортную и доставку корреспонденции. Ранее деятельность Federal Express строилась следующим образом. Не позднее пяти часов утра клиенты должны были привозить кор-

ваются сведения о количестве и адресах посылок. В результате в компании приступают к составлению расписания движения самолетов в восемь часов утра. Таким образом достигается 15%-ная экономия в числе самолетов и пилотов, и компании не требуется закупать дополнительные самолеты и нанимать новых пилотов.

Многие предприятия общественного питания в США планируют и отслеживают заказы и операции с поставщиками в масштабах всей страны с точностью до минуты. Это позволяет им гораздо более эффективно рассчитывать свои потребности на следующий день, опираясь на знание того, что и в каком количестве покупают

1. Новая концепция времени

- В 1989 году мы имели возможность составлять план материальных ресурсов предприятия (МРП) раз в месяц
- К 1996 году мы должны располагать средствами для пересчета МРП при каждом изменении заказа
- На сегодня календарь уже не является адекватной мерой времени



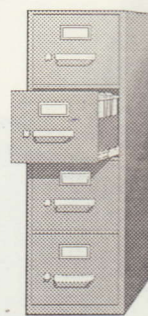
весьма неприятного свойства. Суть в том, что в наше время требуются постоянные и существенные изменения в методах обеспечения конкурентоспособности глобального бизнеса.

Многочисленные встречи с заказчиками из разных стран убедили меня в том, что нужды и проблемы руководителей отделов ИТ везде одинаковы. Я остановлюсь на 10 важнейших изменениях, влияющих на работу отделов ИТ во всем мире.

На первое место я хотел бы поставить изменение самой концепции времени. Раньше руководители отделов ИТ и бизнес-подразделений представляли себе время в терминах календаря, как совокупность дней, недель, месяцев и лет. Сейчас все в большем чис-

2. Новые структуры данных

- На сегодняшний день наши информационные системы ориентированы на транзакции
- В перспективе они должны быть ориентированы на заказчиков, поставщиков и служащих компании

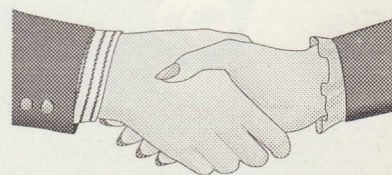


респонденцию в офис компании, после чего планировались маршруты почтовых самолетов. Сейчас в компании используются компьютеры, с помощью которых собираются и обрабаты-

сегодня. Целью является план материальных ресурсов предприятия (МРП), в котором отражается объем закупок, поставок, отгрузки и т.п. В 1989 году мы в нашей компании составляли такой план раз в месяц. В 1996 году мы пла-

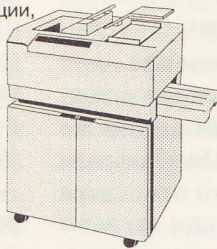
Следствие:

- Для информационных систем межоперабельность важнее предоставляемой функциональности
- Управление этими системами значительно сложнее, чем их построение



3. Новая модель распределения данных

- Мы называем ее "публикация и подписка"
- Основная идея - предоставление всем пользователям именно той информации, которая им необходима
- Централизованное хранилище данных подобно центральному архиву



нируем составлять такие планы с учетом каждого изменения заказов по всему миру. Это является фундаментальным нововведением. Мы должны осознать, что события происходят не день за днем, а ежеминутно и ежесекундно.

Второе изменение непосредственно связано с первым и состоит в необходимости реорганизации баз данных. Большинство компьютерных систем создавалось в процессе автоматизации деятельности отделов, существовавших в компании. Имелся отдел по обработке заказов, отдел по работе со счетами и т.д. Соответственно, создавались отдельные информационные системы по заказам, счетам и т.п. Однако, для поддержания конкурентоспособности следует осознать, что информация группируется не по типам выполняемых операций, а по субъектам, к которым относятся операции. Требуется информация о заказчиках и поставщиках, а не о самих заказах. В этой связи необходимо полное переосмысление структуры информационных систем.

Мы можем наблюдать ростки еще одного изменения, состоящего в переходе к распространению информации по принципу "подписывайся и получай". В качестве пояснения приведем аналогию с копировальной машиной. Если сотрудники организации не имеют свободного доступа к копи-

вальным машинам, приходится создавать централизованные хранилища информационных материалов, где работники могут найти и получить необходимые бумаги. Но как только появляются копи-

ще говоря, понимаются не отдельные люди, а отделы или иные подразделения, но тенденция состоит в создании баз данных каждым сотрудником, которому они необходимы.

Четвертым, и очень важным, изменением является переопределение самого понятия сети, размывание границы между сетями. В качестве примера традиционного понимания сети можно привести Кремль.

Строилась стена, за которую доступ обычным людям был закрыт. Но как только кто-либо получал доступ в пределы Кремля, ему сразу же начинали доверять. Новая модель сети больше похожа на Москву, как на

Публикация и подписка

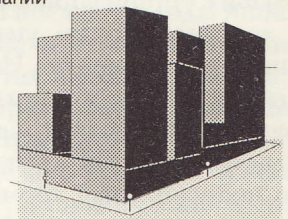
- Ориентация на изменения, нежели на базы данных (на сегодня - пакетные)
- Принадлежность баз данных приложениям и их децентрализация
- Предоставление надежной среды передачи сообщений в целях обеспечения доставки изменений
- Целостность данных - забота получателя, а не отправителя информации
- Централизованная служба имен и адресов для поддержания целостности (непротиворечивости) данных

ральные машины, такие хранилища быстро исчезают. Причина в том, что эффективнее, когда каждый сотрудник получает все необходимые ему материалы, а не когда один отдел пытается решить, какие документы и в каком количестве следует держать в хранилище. Это — еще одно существенное изменение в подходе к построению информационных систем.

Традиционным является построение большой общей базы данных и предоставления доступа к ней всем пользователям. На наш взгляд, в ближайшие 5 лет правильный подход будет состоять в организации индивидуальных баз данных и предоставлении тем самым каждому субъекту доступа ко всей необходимой ему информации. Под субъектами, вооб-

4. Новая концепция сетевых решений

- Электронная рабочая среда
- Общая среда для заказчиков, поставщиков и служащих компании
- Замковая модель больше не годится



некое сообщество. Здесь нет стены, здесь нельзя разделить людей на плохих и хороших, здесь люди находятся в одной среде — сетевой. В единой сети должны сосуществовать клиенты, поставщики, сотрудники компании, партнеры. Это является фундаментальным изменением для отдела ИТ. Я считаю, что на сегодняшний день все информационные сети, работающие в сфере бизнеса, включая сеть нашей компании, построены по модели крепости, такой как Кремль. Поэтому для успешного развития мы должны

понять, как изменить информационные структуры. В этом мы видим основную задачу руководителей отделов ИТ в ближайшие несколько лет.

Пятым изменением является значительное сокращение жизненного цикла прикладного программного обеспечения. Подобно многим, я начинал работать на мейнфреймах. Когда я впервые занялся прикладными программами, я узнал, что средний срок их службы составляет 15 лет. Иногда до сих пор говорят о 20-летней продолжительности

40% изменении того, что ежегодно разрабатывается. Поэтому следует переосмыслить подход к разработке приложений, ибо сегодня они меняются чаще, чем в прошлом.

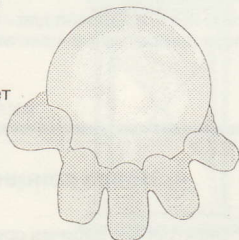
Шестое изменение состоит в том, что системы должны быть глобальными в той же степени, в какой глобальным стал сам бизнес. Это относится ко всем частям России, ко всем странам мира. Во многих компаниях поняли, что изменения в глобальных системах происходят слишком медленно и стоят слишком дорого. Если жизненный цикл системы составляет 15 лет, ее, вероятно, стоит переоснащать и оптимизировать на локальном уровне. Но если вы захотите менять систему каждые 2-3 года, то быстро увидите, что многочислен-

бализация деятельности в течение ближайших пяти лет. В основном это были американские компании. Однако, сегодня японские компании тратят на информатизацию больше средств, чем американские. Это является еще одним глобальным изменением с точки зрения отдела ИТ.

Многие из нас начинали работать на мейнфреймах, будучи уверенными, что очень важно устранить избыточность данных и обеспечить ситуацию, когда все пользователи работают с одним и тем же экземпляром данных. Например, если необходимо проверить чей-то платежный баланс, то, независимо от географического положения пользователя, следует обращаться по определенному адресу на диске. В принципе это правильно, но очень дорого. При работе в распределенной среде, у руководителя отдела ИТ появляются новые возможности, которые необходимо принять во внимание.

5. Сокращение жизненного цикла приложений

- Двадцать лет назад нормальным считался 20-летний жизненный цикл приложения
- Сегодня на Уолл-стрит может идти речь и о 20-минутном цикле
- Сегодня в большинстве случаев жизненный цикл приложений может измеряться месяцами, подобно циклам в бизнесе

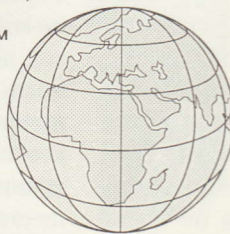


жизни приложений. Во всех больших организациях имеются так называемые унаследованные приложения. Недавно у нас в компании находился крупный специалист в данной области, который сказал, что это неправильный термин для подобных приложений. Термин "унаследованный" применим к тому, что ваша бабушка оставила вам в наследство по завещанию; он обозначает нечто хорошее. Правильнее называть унаследованные приложения предшественниками.

Когда мы встречаемся со своими заказчиками, даже такими крупными, как промышленные корпорации с капиталом более 100 миллиардов долларов, они говорят, что, по их мнению, в ближайшее время срок службы приложений будет составлять не более 2-3 лет. Это трудно себе представить, так как речь идет о 30%-

6. Глобальные системы

- Решение глобальных бизнес-проблем требует применения глобальных систем
- Чтобы оправдать свое существование, группы по разработке локальных приложений должны быть коренным образом изменены
- Локальные системы сильно тормозят глобальные изменения и вызывают их удорожание



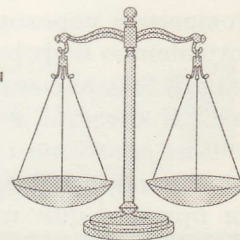
ные изменения с учетом местной специфики невозможны.

За последние два года я посетил более 100 финансовых компаний, и везде в качестве приоритетной задачи называлась гло-

Поясню свою мысль примерами. Как-то мне пришлось побывать в гостях у руководителя отдела ИТ одного из американских банков. В этом банке работают две информационные системы. Одна из них, соз-

7. Новые альтернативы

- Внесение небольших задержек может дать существенную экономию затрат на разработку и функционирование систем
- Таким же образом подобные результаты могут быть достигнуты при допущении небольшого количества ошибок



данная на базе мейнфрейма IBM, поддерживает региональные отделения и установлена в главном офисе. Другая система обслуживает банкоматы, где люди могут в любой момент получить деньги по кредитным карточкам. Перед директором отдела ИТ поставили задачу связать между собой две эти системы, поскольку имелась потенциальная возможность обманного получения денег. Я, например, мог пойти в региональное отделение к кассиру и снять деньги, а затем еще раз снять те же деньги через банкомат. Система срабатывала с опозданием, поэтому я мог дважды получить одни и те же деньги. Все говорили этому руководителю, что, конечно же, необходимо связать эти две системы, чтобы подобное не могло произойти. Подсчитали, однако, что поддержание связи между системами будет стоить свыше миллиона долларов в год, тогда как общая сумма потерь от преступных операций составляет лишь 30 тысяч долларов в год. Поэтому банк отказался от идеи создания единой системы.

Еще один пример. В компании Sun Microsystems все хотели получать отчеты из баз данных мейнфрейма. Мы провели исследование и обнаружили, что примерно 95% деловых нужд можно удовлетворить с помощью данных, имевшихся на конец предыдущего операционного дня. Таким образом,

допуская определенный процент ошибок, можно экономить огромные суммы денег. По информации наших клиентов, отчет, точный на 99%, может стоить в 10 раз меньше, чем отчет со 100% точностью. Этот факт способен существенно повлиять на возможности отдела ИТ по удовлетворению потребностей всей компании.

Восьмым изменением, которое мы наблюдаем, является внедрение мультимедиа. Известно, что люди запоминают 20% прочитанного, 40% увиденного и 70% обработанного. Люди хотят иметь интерактивные средства мультимедиа для существенного повышения эффективности общения в процессе обучения. Если вернуться к идее о постоянных изменениях приложений, то становится ясно, что без улучшения процесса обучения реализовать ее будет невозможно. Следовательно, без развития интерактивных средств мультимедиа мы не сможем менять приложения так быстро, как того требует бизнес. В некоторых американских компаниях полагают, что интерактивные средства мультимедиа необходимы им и для повышения качества сервиса. Множество исследований показывает, что люди реагируют совершенно по-разному на лицо человека и на набор данных. Если вы имеете полную информацию на экране дисплея, вам совсем

не обязательно видеть лицо человека, но поведение людей совершенно различно при общении с человеком или с экраном.

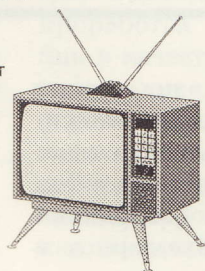
Внедрение глобальных интерактивных средств мультимедиа в такой компании, как Sun, будет экономически выгодным уже в ближайшие 5 лет. Препятствия этому носят скорее политический, нежели экономический, характер. В Европе, в частности, государство взимает с компаний высокую плату за коммуникации, чтобы удешевить домашние услуги связи. Стоимость телекоммуникационных сетей в Европе в 10 раз выше, чем в США, и в 5 раз выше, чем в Японии и других азиатских государствах. Это создает целый ряд финансовых проблем в ЕЭС. Одна группировка говорит, что нужно защищать семьи, пользующиеся телефонными услугами на дому. Другая же группировка утверждает, что в целях сохранения конкурентоспособности Европе следует значительно снизить плату за услуги связи.

В последние годы Япония перешла к стратегии удешевления тех направлений, где ее цены превышали цены США в 5 раз. Все это в ближайшие 5 лет сделает средства мультимедиа экономически выгодными для крупных компаний.

Девятой важной тенденцией, которую мы наблюдаем, является то, что пользователи

8. Привлечение средств мультимедиа

- Мы запоминаем 20% прочитанной, 40% просмотренной и 70% обработанной информации
- Непосредственное общение повышает качество работы и удовлетворяет запросы заказчиков



9. Взаимодействие с пользователями

- Системные ресурсы представляют собой небольшую часть общей стоимости проекта
- Задание параметров систем и их разработка должны производиться в тесном взаимодействии с заказчиками этих систем



настаивают на своем активном участии в системных проектах отделов ИТ. Наш опыт показывает, что в основных проектах от 2/3 до 90% составляют пользовательские ресурсы, а не ресурсы отдела ИТ. Это является серьезной организаци-

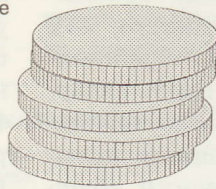
онных решений. Руководители практически всегда сталкиваются с людьми, ратующими за сложность, а не за простоту. Его собственные сотрудники любят сложные решения, поскольку они интересны, требуют от них определенного масте-

рства и влияют на их карьеру. Заказчики и пользователи часто требуют сложности, так как это избавляет их от принятия очень трудных и кардинальных решений — сложные вещи внедряются постепенно и влияют на их карьеру. Заказчики и пользователи часто требуют сложности, так как это избавляет их от принятия очень трудных и кардинальных решений — сложные вещи внедряются постепенно и влияют на их карьеру.

Этот способ мышления в корне отличается от традиционной модели ИТ, которая гласит: "Определите приложения, которые требуется поддерживать, и затем стройте соответствующую инфраструктуру". Однако, если приложения заранее неизвестны, стратегия должна быть иной.

10. Снижение затрат

- Добиваться лучших результатов в кратчайшие сроки при меньших затратах
- Снижать затраты на инфраструктуру и имеющиеся системы с целью высвобождения ресурсов на разработку новых систем, дающих преимущества в конкурентной борьбе



онной проблемой, поскольку сам проект исходит от отдела ИТ, а большая часть ресурсов ему (отделу) не принадлежит. Насколько я знаю, почти каждая крупная неудача в разработке информационных систем связана с тем, что пользователи и разработчики представляли себе проект по-разному.

Десятая тенденция отнюдь не нова, но сейчас она сильнее, чем когда-либо. Практически в любом бюджете отдела ИТ последним пунктом записано намерение осуществлять проекты быстрее, лучше и с меньшими затратами. Следовательно, в отделах ИТ испытывают двойной прессинг — с одной стороны, им необходимо внедрять новшества, а с другой стороны — сокращать расходы.

Я руковожу отделом ИТ уже в течение 5 лет и, по мнению многих, являюсь экспертом в вопросах информатизации. Не знаю, насколько это верно, но я вывел для себя шесть принципов успешной работы руководителя отдела ИТ.

Во-первых, он должен обеспечивать простоту инже-

ленно. Но в конечном итоге простота поможет вам в работе, она будет способствовать деловым успехам. Первая заповедь — изодня в день говорить "нет" сложным решениям, поскольку это абсолютно не то, что требуется.

Во-вторых, по моему мнению, вы должны максимально увеличить число взаимосвязей в организации, в том плане, что все сотрудники должны быть объединены в сеть. Дело в том, что ваша ин-

Как добиться этих результатов?

- Управление информационными технологиями намного важнее и сложнее, чем их внедрение и работа с ними
- Принятые Вами архитектурные технологические подходы определяют Вашу архитектуру управления информационными технологиями



Третьим пунктом является стремление максимально приблизить вычислительные мощности к пользователям. Это положение можно выразить следующей фразой: "Локализа-

Принципы успешного выживания с точки зрения руководителя отдела ИТ

- Простота инженерных решений
- Расширение взаимосвязей
- Сглаживание нагрузок по обработке данных
- Совершенствование архитектурных подходов
- Уменьшение размеров приложений
- Баланс между правами и обязанностями

Для руководителя отдела ИТ ВРЕМЯ, как для финансового директора - ДЕНЬГИ

ция пиковых нагрузок, сглаживание сетевых нагрузок". Опять-таки, главной задачей при нынешнем уровне технологии является управление сетью. Вы можете существенно облегчить управление путем сглаживания вычислительной

нагрузки в максимально возможной степени.

Четвертый момент состоит в необходимости иметь набор правил, которым все должно следовать. Здесь очень трудно выдержать баланс между количеством правил и степенью свободы, которая дается людям для решения их проблем. Когда я стал руководителем отдела ИТ в Sun, я получил "в наследство" стандартный свод правил в шести довольно толстых томах. Сегодня эти правила умещаются на 10 страницах и за их выполнением мы следим очень строго. Убежден, что обеспечить выполнение правил, записанных на тысяче страниц, практически невозможно. Думаю, что отделам ИТ стоит потратить значительное время, чтобы правильно провести грань между свободой, предоставляемой при реализации проектов, и теми положениями, выполнение которых считается обязательным.

Пятый пункт гласит, что мелкие проекты стоят значительно дешевле крупных. Многолетние исследования в США показали, что при увеличении размера проекта в два раза, его стоимость увеличивается в 8 раз. Кроме того, самоуправление команды разработчиков не представляется возможным, если ее размер более 14 или 15 человек. Поэтому, когда вы добавите к такой команде еще двух человек, то следующим шагом станет ее увеличение до 30 или 40 человек, так как лишь 14 человек могут скоординировать свои действия без участия руководителя. Как только численность увеличится, потребуются централизованное планирование и руководители для координации работы. Поэтому мы и стараемся сохранять небольшие размеры проектов.

Между двумя сформулированными положениями существует важная взаимосвязь.

При правильно выбранной архитектуре можно решать реальные задачи в рамках небольших проектов. Подобный подход требует определенной смелости, поскольку по ходу дела проект зачастую выглядит очень неорганизованным, даже хаотичным. Однако, наш опыт показывает, что те команды разработчиков, которые знают, что им нужно делать, и чувствуют, что они в силах это сделать, достигают намеченной цели.

Последним в моем списке из 6 пунктов стоит очень важное положение. Я считаю, что необходимо уравнивать степень ответственности за выполнение проекта с возможностями по его выполнению.

В начале 1980-х годов фирма IBM рекомендовала стратегию, которую она называла "экспорт команд". В IBM считали, что следует выделять команды разработчиков приложений и передавать их бизнес-партнерам. Но когда они попытались осуществить этот подход в среде мейнфреймов, то обнаружили, что на самом деле все по-прежнему программируют вместе. Однако я полагаю, что для компаний со сложной структурой абсолютно необходимо передать разработку приложений в группы пользователей. Я еще вернусь к этому вопросу.

Последнее из общих положений, на которых я хотел бы остановиться, касается роли руководителя отдела ИТ во всей организации. Я два года проработал в Sun в качестве руководителя финансового отдела. Моя работа состояла в контроле над эффективностью расходо-

вания средств компании. Сегодня у меня похожая работа — следить за эффективностью расходования времени. Она состоит в ускорении всех процессов в компании — обработки заказов, ответов на вопросы заказчиков, реакции на мириады проблемы и т.д. В японских компаниях все это уже делали более 20 лет назад.

Фактор времени имеет огромное значение. Если вы хотите, чтобы ваш бизнес развивался успешно, вы должны полностью исключить время из бизнес-процесса. Если вы сделаете это, ваш бизнес станет более конкурентоспособным и прибыльным. Поэтому вы должны в первую очередь сосредоточить усилия на исключении времени из бизнес-процесса и только во вторую очередь думать о его эффективности.

Перед тем как перейти к специфическим мерам, которые мы предприняли в Sun, я постараюсь представить вам свой взгляд на компанию. В 1995 году доход Sun составил около 6 миллиардов долларов. Компания имеет примерно 200 офисов в различных частях света. Специфика Sun состоит в очень высоких нагрузках на работников. Сегодня мы работаем как компания с 13-миллиардным оборотом. Обычно мы получаем 8% доходов с продаж в последний день квартала, а 1/4 дохода — в последнюю неделю квартала. С точки зре-

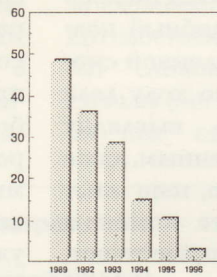
Компания Sun Microsystems, Inc.

- Годовой доход около 6 млрд. долларов
- Глобальный масштаб деятельности и высокие пиковые нагрузки
- Небольшой срок работы компании
- Около 14.000 служащих компании
- Критические маркетинговые сроки - продукты, разработанные в последние 1,5 года, приносят 95% дохода
- Стремительные изменения во всех сферах деятельности
- Жесткая сосредоточенность на продолжительности цикла
- Непрерывная перестройка



Время подведения финансовых итогов

- Объем работ связан с количеством времени
- Уменьшение времени подведения итогов повлекло за собой улучшение инвестиционного климата и управленческих процессов, а также уменьшение финансовых затрат



ния руководителя отдела ИТ это полный кошмар. Мы должны выдерживать очень высокие нагрузки и не можем позволить себе неудач, так как это будет иметь отрицательные последствия для компании.

В феврале 1995 года нашей компании исполнилось 13 лет. Ее штатная численность составляет 14 тысяч человек, но во всем мире около 45 тысяч человек работают на Sun. Маркетинговые сроки для нас имеют чрезвычайно важное значение, поскольку почти весь наш доход обеспечивают продукты, разработанные не более полутора лет назад. В нашей компании мы пользуемся таким параметром, как "время полного цикла". Это количество дней от момента заказа комплектующих у поставщика до получения платы от заказчика. Если вы уменьшите время полного цикла, вы улучшите свой бизнес. В нашем случае такая реорганизация высвободила из оборота свыше 1 миллиарда долларов. В 1989 году время полного цикла составляло 273 дня. Этот показатель являлся средним для компьютерной отрасли, следовательно, наши дела в то время были не так уж плохи. Через 4 года мы уменьшили этот показатель примерно вдвое. Сегодня время нашего цикла составляет половину от аналогичного показателя такой компании, как Tandem, и является лучшим в компьютерной отрасли, находясь в ок-

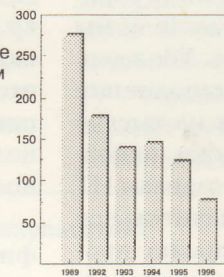
рестности половины среднего значения. Сейчас мы находимся приблизительно на подорожке к сокращению данного показателя еще вдвое в течение двух лет. Но для того, чтобы быть

результатов по окончании года. В 1989 году этот показатель составлял 47 дней. Мы ожидаем, что в 1996 г. он составит 2 дня. Два дня — это рекорд отрасли, установленный компанией Motorola. Сейчас это может показаться несколько абстрактным, но на практике выигранное время можно использовать для изучения, почему вы поступаете тем или иным образом, после чего можно начать

что-то делать по-другому, а от чего-то и вовсе отказаться. Таким образом, мы смогли существенно повысить эффективность нашей работы и снизить издержки.

Время полного цикла

- Стоимостная цепочка
- Поставщики определяют временное регулирование по дням и продажам
- Снижение продолжительности цикла ведет к улучшению деловой ситуации
- Наш балансовый отчет пополнился примерно 1 миллиардом долларов



конкурентоспособным в 1996-1997 годах, время полного цикла должно составлять 60 дней, тогда как на сегодня оно составляет у нас около 140 дней. Мы полагаем, что компании, которые успешно проводят реструктуру, — это те, кто способен определить подобные показатели и сосредоточиться на мерах по их уменьшению.

Понятие времени цикла применимо практически во всех сферах. Приведу еще один пример — число дней, которое требовалось нашей бухгалтерии для подсчета финансовых ре-

История развития бизнес-приложений

- Использование инвентарных ведомостей в дополнение к несовершенным программным продуктам
- Переход к полномасштабному, но эпизодическому планированию требований к материалам с помощью системы Cullient
- Глобальное управление товарными запасами по CAS
- Составление прогнозов и отчетов в системе SunGlobe
- Оформление заказов в системе SalesDesk
- Центры доставки продуктов
- Рост автоматизации планирования требований к материалам

График внедрения системных решений

- Большое количество ЭВМ HP 3000, работающих с ASK
- Внедрение системы Cullient на ОС MVS компании CIM не завершено
- Постепенный переход к системе Cullient в качестве сетевого ресурса, репликация данных
- Существование на сегодня сложной сети из более чем 300 систем, более 200 интерфейсов к мэйнфреймам
- Ликвидация всех ЭВМ HP 3000 с переходом к Oracle
- Замена мэйнфреймов теперь планируется в 1996 году

В начале нашего пути, в частности, в 1989 году, наша компания, имея слабые информационные системы, использо-

Ошибки, которые мы допустили:

- Предположение, что данный проект будет системным
- Недооценка соотношения временных рамок, определяемых внутренними и внешними обстоятельствами: организационные вопросы заняли гораздо больше времени, чем реализация проекта
- Недооценка силы сопротивления сторонников действующей системы любым переменам
- Ориентация на завышенные требования пользователей в ущерб рациональным требованиям
- Надежда на программное обеспечение, которое не существует еще в бета-версиях для реальных заказчиков

вала множество инвентарных документов, чтобы скрыть недостатки автоматизации. Наихудший пример, который я могу здесь привести, состоит в следующем. Мы завозили мониторы в США из Японии морским транспортом. Мы разгружали их в Лос-Анджелесе, затем на грузовиках везли в Сан-Франциско, после чего на других грузовиках доставляли в аэропорт Сан-Франциско для последующей отправки обратно в Японию. У нас не было других возможностей управления этим потоком продукции. Через некоторое время мы внедрили мейнфреймовую систему на IBM-совместимых машинах, где использовали прикладную программную систему Cullinet, версия 2. Мы поместили все инвентарные документы всех подразделений Sun в эту среду.

В 1989 году мы не знали объема своего дохода или прибыли, пока не закрывали "кошелек". Сегодня мы создали систему под названием SunGlobe, которая общается с множеством систем по всему миру и позволяет нам каждое утро получать сведения о нашем текущем рыночном положении.

Сокращая время полного цикла, мы производим и множество других изменений в своей деятельности. Я приведу один пример, который поможет вам понять, почему вы должны думать обо всем, так как часто основная проблема

состоит в том, о чем вроде и говорить не стоит. Оказалось, что главным фактором, влиявшим на время полного цикла в Sun, являлось качество составления спецификаций сейлсменами.

му вы можете представить себе состояние этих систем. Предыстория "взаимоотношений" с HP3000-ASK такова. После многих разочарований, в 1985 году мы остановились на новой среде. Решение об этом принималось в течение двух лет. В конце концов за него проголосовали единогласно. Однако оно оказалось ошибочным. Дело в том, что такого продукта просто не существовало. Коми-

тет проголосовал за обещания сейлсменов. В результате, до того, как мы реализовали решение, компания отказалась от продукта. Вот почему мы пришли к системе распределенных вычислений раньше

Пути реализации:

- Использование технологических принципов в качестве движущей силы при общем бизнес-руководстве системами с одновременной стандартизацией структуры организаций
- Согласование единого архитектурного подхода на самом высоком уровне с распределением реализации элементов проекта по небольшим командам
- Просвещение пользователей по вопросам стоимости новейших программных продуктов и их ориентация на использование стандартных приложений
- Установление прочных взаимоотношений с нашими поставщиками программных продуктов

Когда мы начали работу над проектом и высшие руководители заговорили об этом, никто даже не подозревал, что могут возникнуть проблемы со спецификациями. Но неверно составленные спецификации влияли на заказы. Плохие заказы не могли быть введены в систему. Когда заказчик получал продукцию, оказывалось, что ему требовалось нечто иное. Затем мы снова обсуждали с заказчиком спецификации, так как заказ и поставка не совпадали. Поэтому, когда вы размышляете о времени цикла, вы должны мыслить очень широко, ибо заранее не ясно, что именно влияет на это время.

В начале своей деятельности компания Sun пыталась работать на компьютерах HP3000, используя программную систему ASK. У нас было очень много таких машин. В середине 1995 года мы расстались с последней из них. С 1987 года мы не покупали ни одной новой машины HP3000, поэто-

других. Мы спланировали мейнфреймовый вариант, а затем отказались от него еще до того, как закончили миграцию.

Мы, конечно, должны были использовать сделанные наработки, но в результате мейнфреймовое приложение стало сервером. Сегодня по всему миру мы имеем свыше 300 систем, работающих на компьютерах Sun, и свыше 200 из них взаимодействуют с мейнфреймовыми базами данных. Мы ставим цель прийти менее чем к 50 системам. Такой подход отличается от того, чего мы добивались 20 лет назад.

Я беседовал с одним из наших заказчиков, который преследует ту же цель — иметь 40 или 50 систем. Это японская компания. Я спросил его, сколько систем они имеют сегодня. Он ответил, что свыше 300 тысяч. Они убедились, что в такой ситуации изменения невозможны, и он понимает, почему. Я считаю, что это служит характеристикой той среды, к

которой мы все движемся, и которая подразумевает наличие даже для самой крупной компании 10, 20, 30, 40 или 50 систем, но никак не 10 тысяч, 50 тысяч или 100 тысяч. Это уже совсем другая среда.

Ключевое изменение, которое мы произвели, было почти случайным. Мы обнаружили, что большинство мейнфреймов использовались не для обработки транзакций, а для генерации отчетов. Поэтому мы скопировали мейнфреймовую базу данных в СУБД Sybase, работающую на сервере Sun. Затем мы сказали нашим сотрудникам, которым требовались отчеты, что если они хотят получать отчеты на мейнфреймах, то вот им один счет, а если они согласны воспользоваться сервером Sybase, вот им другой счет, намного дешевле. Мы спрашивали у них, что они хотели бы выбрать. Мы обнаружили, что, как я сказал ранее, к счастью, все отчеты, которые составлялись в компании, могли быть выполнены по окончании дня или на основании итогов предыдущего дня. Текущие данные с мейнфреймов для отчетов не требовались. Таким образом, все больше людей стали запрашивать скопированные данные, и теперь мы имеем по всему миру свыше 20 копий мейнфреймовых баз данных, которые делаются каждый вечер.

Хотелось бы подчеркнуть, насколько важно осозна-

вать различие между обработкой транзакций и запросов. Преимуществом их разделения является значительное удешевление выпуска и тестирования программного обеспечения. Если вы работаете с отчетами в среде, критически важной для компании, вы должны все тщательно проверять, так как нельзя допустить, чтобы отчет разрушил ответственную систему. Однако, если ошибка ничем особенным не грозит, можно отказаться от тщательного тестирования. Практика показывает, что мы экономим гораздо больше денег на выпуске и тестировании программ, чем стоят аппаратное обеспечение и эксплуатация дополнительных систем. В течение года большая часть затрат на новые приложения окупилась.

Суммируем организационные изменения, которые были проведены в компании за последние 5 лет. Во-первых, мы завершили стандартизацию и централизацию нашей инфраструктуры. Мы обнаружили, что это существенно удешевляет инфраструктуру и, что более важно, это резко снижает расходы на разработку, тестирование и выпуск приложений. Мой опыт показывает, что здесь мы экономим по два доллара на каждый доллар, сэкономленный на удешевлении инфраструктуры. Как говорится, "функциональность обходится дешевле свободы", что означает, что гораздо дешевле

предоставить пользователям среду с очень богатыми функциональными возможностями, стандартизованную и централизованную, чем оставлять выбор среды за пользователями. Во-вторых, мы разработали и

проводим в жизнь в рамках компании единые архитектурные принципы. Как-то я услышал такое высказывание: "Взгляните на каждое правило, которое действует в вашей компании для сотрудников отдела ИТ и спросите себя, уволили бы вы работника за нарушение какого-либо из этих правил, и если нет, то исключите это правило". Здесь очень важно регламентировать то, что люди обязаны делать или не делать, а не то, что они могли бы делать в процессе решения конкретных задач.

Третьим принципом является передача ответственности за приложения в бизнес-группы. Это приводит нас к зеркальному отражению того, что имеет место во многих компаниях. Мы централизовали управление настольными системами и децентрализовали управление серверами. После трех лет работы с этой моделью я понял, что ею гораздо проще управлять, чем противоположной.

Сведем вместе шесть основных принципов нашей технологической архитектуры (многие из них уже упоминались).

Первым вновь является "Локализация пиковых нагрузок вблизи пользователей, сглаживание сетевых нагрузок".

Второй принцип: везде, где это возможно, используйте тиражирование, а не разделение данных.

Третий принцип: по возможности производите интеграцию на настольных системах, а не на серверах. Имеется множество пользователей, которые могут сказать, что они работают с приложением под названием, например, "торговый дом". Они рассматривают его как единое приложение, которое полностью работает на одной настольной системе. Но

Три основополагающих принципа

- Стандартизация и централизация инфраструктуры
 - Снижение прямых затрат
 - Существенное снижение отнесенных затрат
 - Функциональность обходится дешевле свободы
- Создание и претворение в жизнь единого архитектурного подхода
 - Принцип "большого правила"
 - Принцип "надо", а не "надо бы"
- Децентрализация приложений по бизнес-группам в рамках данной инфраструктуры и данной архитектуры

Архитектурные принципы

- Локализация пиковых нагрузок, сглаживание сетевых нагрузок
- Репликация данных в большей степени, чем разделение
- Интеграция на уровне рабочих мест пользователей, а не на уровне серверов, а также работа с приложениями для организаций на этом уровне
- Максимальная специализация серверов
- Использование рабочих мест пользователей в большей степени как полноценных элементов сети, а не интерфейсных устройств
- Однородность и масштабируемость снижают затраты как на функционирование приложений, так и на их разработку и сопровождение

с точки зрения отдела ИТ, это на самом деле около десятка приложений. Некоторые из них работают на мейнфреймах, некоторые — на настольных системах, а некоторые — на серверах. Таким образом, пользователь видит единую среду, а отдел ИТ должен поддерживать множество составных частей.

Четвертым пунктом является максимальная специализация серверов. Это является огромным преимуществом, которое можно ощутить только в процессе работы. Другими словами, на одном сервере всегда работает одно приложение, а не несколько. Преимуществом здесь является четкое представление о том, кто является ответственным за данное приложение. Если требуется более крупный сервер — пожалуйста, покупайте. Если требуется более дешевый сервер — покупайте и такой. Если нужна новая версия базы данных — устанавливайте, если захочется работать с новой операционной системой — ваши проблемы. Все решения локализованы в пределах одной группы, занимающейся одним приложением, и руководителю отдела ИТ не придется тратить время на согласования в пределах всей компании. Всеобщей целью является максимальная изоляция изменений.

В качестве последних двух пунктов отметим следующие положения. Мы на самом деле ста-

зок. В США это называется стратегией "толстого клиента", в противоположность "тонкому". ПК являются типичными "тонкими клиентами", а рабочие станции — "толстыми клиентами".

В теории не важно, какого подхода придерживаться. На практике же легче управлять сетями "толстых клиентов", чем сетями "тонких".

Преимуществом такого способа работы является его оптимальность с точки зрения производительности как отдельных лиц, так и компании в целом. Пиковые нагрузки имеют тенденцию к самоограничению, так как пользователи не могут одновременно делать слишком много операций. Пользователи всегда стараются выжать из компьютеров максимум возможного. Если "удар" примут на себя настольные системы, управлять пиковыми нагрузками будет гораздо легче, поскольку нагрузки индуцируются живыми людьми.

Упрощается и управление сетями. По размеру расходов на администрирование настольные системы не отличаются от терминалов, однако для пользователей они представляются как ПК. Вам не требу-

ется использовать настольные системы не только на уровне представления, в интерфейсных целях, но и для настоящей работы и полагаем, что это важно для реализации принципа локализации нагру-

ется выдвигать минимальные предположения о том, что вы станете делать в будущем, поэтому вы обладаете большой гибкостью в отношении изменений. И вы имеете четкое разграничение ответственности — всегда известно, кто за какое приложение отвечает.

Если взглянуть на возможные конфигурации компьютерных систем, среди которых пользователи могут делать вы-

Преимущества данной архитектуры

- Повышение индивидуальной и корпоративной производительности
- Автоматическое ограничение пиковых нагрузок и упрощение управления сетями
- Настольные системы требуют административных расходов терминалов, но имеют характерные особенности персональных компьютеров
- Минимальный набор исходных посылок в отношении требований к будущим приложениям
- Возможность контроля приложений как в процессе разработки, так и в процессе их функционирования

бор, можно отметить пять конкурирующих моделей: мейнфреймы, локальные сети ПК, ПК, работающие совместно с мейнфреймами, сети на базе UNIX-систем, а также сетевые компьютеры на базе Windows NT, которые начинают внедрять Microsoft и которые работают по модели, похожей на UNIX.

Эти модели мы будем оценивать по 4 показателям:

- ОСВ (общая стоимость владения) — сумма затрат на разработку, покупку, управление и эксплуатацию.
- Устойчивость — способность выдерживать изменения.
- Производительность для отдельных лиц.
- Производительность для компании в целом.

Если взглянуть на мейнфреймовые системы (а они, как известно, всегда были дорогостоящими), то окажется, что их высокая стоимость вызвана сильной централизацией. Показатель персональной произ-

Современное состояние рынка

	Совместимость	Устойчивость к ошибкам	Производительность для отдельных лиц	Производительность для организаций
Мейнфреймы	Плохо	Удовлетворительно	Плохо	Удовлетворительно
Локальная сеть на базе ПК	Удовлетворительно	Плохо	Хорошо	Плохо
ПК+мейнфреймы	Плохо	Удовлетворительно	Хорошо	Хорошо
UNIX	Хорошо	Хорошо	Удовлетворительно	Хорошо
Windows NT	Удовлетворительно	Удовлетворительно	Хорошо	Удовлетворительно

водительности у них никогда не был хорошим. Показатель устойчивости — удовлетворительный, но не хороший, поскольку вся система сосредоточена в одном "ящике". Следовательно, при изменении определенных параметров вам придется использовать мейнфрейм в неизвестной среде. Производительность на уровне компании всегда была хорошей, для мейнфреймов всегда существовали замечательные приложения. Проблема состоит в том, что никто сейчас не разрабатывает новых приложений для мейнфреймов. В США больше не ведется обучение студентов для работы на мейнфреймах. Современные приложения выбирают реляционные СУБД из числа гораздо более ЦП-интенсивных, чем те, что в состоянии поддерживать мейнфреймы. Поэтому на сегодня можно характеризовать их производительность на уровне компании как удовлетворительную, но уже не хорошую.

Большинство людей считает показатель ОСВ у локальных сетей ПК удовлетворительным. Важно то, подсчитываете ли вы стоимость для компании или для своего бюджета. Данные по США говорят, что из каждых 20 "белых воротничков" один является администратором систем на ПК. С другой стороны, многие данные свидетельствуют, что активный пользователь ПК тратит от 2 до 4 часов в неделю на обслуживание ПК, и это время обычно не включается в

уровне компаний и в том случае, если работать на ПК в сети Novell, и в случае, если использовать их для других, заранее не известных, целей.

От года до двух назад по общему мнению наилучшим решением считалась модель ПК + мейнфрейм. В большинстве случаев эта модель называется "клиент-сервер". Она давала хорошие результаты с точки зрения как производительности отдельных лиц, так и компаний. Для реализации изменений она вполне подходит, это не совсем просто, но вполне удовлетворительно. Двумя годами позже стали считать, что это решение невероятно дорого и просто непозволительно. Теперь можно наблюдать отступление компаний к исходным позициям и переосмысление своей стратегии в этом вопросе.

Сетевые вычислительные системы на базе UNIX имеют все показатели "хорошо", за исключением индивидуальной производительности, где решения не настолько хороши, как в мире Windows.

Системы NT в той степени, в которой они работают с приложениями Windows, имеют хорошие показатели, но необ-

стоимость работы с сетями ПК, а с точки зрения компании оно определено относится к затратам. Сети ПК хороши в отношении персональной производительности, но показывают слабую производительность на

ходимо 3-5 лет на их совершенствование. При этом требуется много изменений. В силу целого ряда причин будет очень сложно построить большой высокопроизводительный сервер на NT на платформе Intel в течение еще многих лет.

Таким образом, круг замкнулся. Мне представляется, что происходит движение от мейнфреймовой модели IBM 1970-х годов к мейнфреймовой модели 1980-х годов. Однако технологические принципы и реализация будут совершенно иными. Будет использоваться более широкий круг независимых поставщиков ПО, а среда станет более дружественной по отношению к ПК. Будет использоваться ряд сервер-фармов из больших специализированных серверов, вместо того, чтобы иметь один или два мейнфрейма. Вы получите вычислительную систему общего назначения, работающую на терминале, а не на сервере или мейнфрейме. Вы будете иметь более крупную, но по сути ту же модель управления, которую IBM рекомендовала в начале 80-х годов. Причиной того, что это нас не удивляет (хотя это не совсем так), является то, что проблема управления остается прежней, и на этом примере можно видеть, что и другие упомянутые проблемы остались прежними.

Важный урок, который я усвоил за последние 5 лет, состоит в том, что управление

Мы замыкаем круг...

- От мейнфреймовой модели 1980-х годов мы собираемся эволюционировать до аналогичной модели управления для информационных технологий, но с иной реализацией
- Больше число независимых поставщиков
- Большая совместимость, нежели отчужденность от ПК
- Набор специализированных серверов
- Универсальные вычислительные возможности "терминала"
- Сохранение прежней модели управления

информационными системами гораздо важнее, чем сама их работа. Все мы постоянно разрываемся между тремя конкурирующими силами. Одна из них — это что позволит сделать технология. Вторая — чего захочет пользователь. Третья — что все это должно работать 24 часа в сутки, 7 дней в неделю, с надежностью 99.999%, в глобальном масштабе.

Основной вывод — я не думаю, что проблемы сильно изменились за последние 15 лет.

Изменились методы их решения.

По завершении доклада У. Дж. Радучел ответил на следующие вопросы.

Не могли бы Вы более подробно остановиться на проблеме "тонких" и "толстых" клиентов. Я полагаю, что для поддержки "толстого" клиента требуется более квалифицированный технический персонал. Это касается как специалистов по сервисному обслуживанию, так и тех, кто работает с клиентом. Как относятся в компании Sun к проблеме "толстых" клиентов?

И рабочие станции, и ПК могут быть как "тонкими", так и "толстыми" клиентами. Я видел реализации обоих вариантов. Единственным вариантом работы "тонких" клиентов в настольных системах является работа на уровне представления. Есть дисплей и пользователь. Пользователь считывает данные с дисплея, затем вводит в настольную систему свою информацию. Далее имеется уровень приложений, который работает на сервере. Основная сложность заключается в реализации трехуровневой схемы "клиент-сервер", а не в самой ее архитектуре. Вы можете разработать программное обеспечение на трех уровнях, а реализовать его на двух, либо разработать программное обеспечение на трех уровнях и

реализовать его также на трех уровнях. Трудность заключается в наличии промежуточного сервера, который требует соответствующей настройки. Если вы представите свою сеть как систему скоростных магистралей, то подобный сервер приложений можно представить как кольцевую дорогу, помещенную в самый центр скоростной магистральной системы. Серверы приложений трудно настраивать, обычно их настройки влияют на всю сеть.

Этот вариант и более дорогостоящий, так как пиковую мощность приходится приобретать как бы несколько раз. В то же время, в реальной жизни, если имеется, например, восемь пользователей и три сервера (один для выписки счетов, другой для оплаты счетов, третий для приема заказов), то, в зависимости от даты, все будут работать с одним из них, поэтому нет необходимости одновременно поддерживать пиковую мощность на всех трех серверах. Из разговоров с людьми, которые реализовали такую модель, я сделал вывод, что трехуровневая модель связана с очень высокими эксплуатационными расходами, поскольку требуется постоянно поддерживать баланс сетевой нагрузки на низком уровне. Все финансовые среды (биржи, брокерские фирмы) являются "толстыми" клиентами.

Действительно, управлять настольными системами весьма непросто. Мы используем и продаем технологию, которая называется "автоклиентом". Автоклиенты используют локальный диск как кэш тех данных, которые находятся на серверах. В этом случае настольный клиент имеет нулевую стоимость администрирования. Это просто фрагмент технических средств. Если он ломается, вы можете заменить его другим, он не требует ни секунды на выполнение допол-

нительных операций. С логической точки зрения, наши настольные системы не отличаются от серверов приложений — они стандартизованы и работают в устойчивой среде. Но вы получаете преимущество, состоящее в том, что все вычисления производятся на вашем столе; тем самым облегчается управление сетью.

Данные, которые мне приходилось видеть, а это в основном данные по США, говорят о том, что затраты на управление сетью лишь на одну четверть или даже менее чем на одну десятую приходятся на оборудование. Все остальные расходы связаны с управлением людьми и сетевым окружением. Летом 1994 года я присутствовал на семинаре в Next World. Я выступал вместе с пятью другими руководителями отделов ИТ на тему о том, как заставить систему "клиент-сервер" работать. Все эти пять руководителей рассказали свои истории успешной эксплуатации системы "клиент-сервер", в основном в стандарте NT. И что самое интересное — у всех пятерых клиенты были "толстыми", тогда как во всех других фирмах клиенты были "тонкими". Я, может быть, не сумею достаточно понятно объяснить, но у меня есть сильное убеждение, что вычислительную нагрузку нужно приближать к пользователю всегда, когда это возможно. Но я также уверен, что с точки зрения централизованного администрирования, ситуация с "толстыми" клиентами не хуже, чем в случае, когда имеются терминалы и сервер. Централизованной базой данных легко управлять, а пользователи согласятся потратить некоторое время на стабилизацию работы своего (именно своего) компьютера.

Когда вы разрабатываете систему в архитектуре "клиент-сервер", клиент всегда полу-

чается "толстым", как бы вы ни старались добиться обратного. Ему не надо в этом помогать. У него всегда есть внутренняя потенция к расширению. И если мы говорим о принципах разработки, то основной принцип должен состоять в том, чтобы пытаться сделать клиента "тонким", а о своем весе он позаботится сам, без нашей помощи. Здесь у меня та же логика, что и в американских вооруженных силах, когда был введен пороговый вес для генералов Пентагона, и в результате многие из них были уволены. Поэтому я полагаю, что задачей руководителя отдела ИТ должно быть определение этого порогового веса для клиента, после чего его должны увольнять.

Мы не позволяем пользователям влиять на то, что находится на клиенте, они не управляют клиентами. Поэтому с точки зрения пользователей клиент является очень "тонким". С точки зрения вычислений он является тонким, так как мы решаем, какие вычисления будут производиться на настольной системе. С информационной точки зрения он также является очень "тонким", практически нулевых размеров. Все, что есть у него на диске — это кэш данных и программ, хранящихся на сервере, у клиента нет ничего индивидуального. Как говорится, "в этой корзине яйца не лежат". Таким образом, с точки зрения пользователя или отдела ИТ, это очень "тонкий" клиент. Но с точки зрения сети и сетевого управления это "толстый" клиент. И это именно та модель, которая способна работать.

Имеется два пути решения проблем управления. Я считаю, что правильным будет один из ответов. Но это мое личное мнение, и я не могу его доказать. Один ответ — это очень "тонкий" клиент, когда вы пытаетесь максимизиро-

вать информационные потоки к серверу, способные пройти по тонким проводам. Это структура, которую вы должны использовать, если у вас очень много клиентов, подключенных в разных местах, и низкая пропускная способность линий связи. Если же эти пользователи объединены в какие-то логические группы, то вы можете передать данные на серверы, которые обслуживают эти группы. Мы использовали обе модели, так как приобретенное нами программное обеспечение поддерживает иногда только одну модель.

Наш опыт показывает, что второй способ, связанный с копированием данных, гораздо лучше первого, так как он позволяет сохранять работоспособность даже в случае аварий в сети. Об этом стоило бы рассказать тем, кто собирается поехать в Нидерланды. Мы отправляем свои продукты во все страны Западной Европы с нашего склада в Нидерландах. Когда мы отгружаем продукт со склада, то на практике мы требуем только накладную, составленную на местном языке и в местной валюте. Таким образом, мы организовали эту систему так, что она может работать без телефонной связи. Если сеть полностью находится в рабочем состоянии, она функционирует в режиме реального времени. Но в Голландии, так же, как и во всем мире, фермеры не ладят с компьютерной техникой. И в начале лета 1995 года восемь фермеров перерезали все телефонные линии в обоих направлениях. Такая же история случилась осенью 1994 года в США, когда фермеры разрушили линии оптоволоконной связи между США и Азией. Я не представляю себе, как управляться с фермерами во всем мире. Поэтому мы постарались построить нашу систему таким образом, чтобы все наши со-

трудники могли работать с самыми свежими из доступных данных, пока сеть не будет восстановлена. Практика показала, что таким образом можно строить очень надежные, критически важные приложения.

Я считаю, что лучше всего работать с локальными копиями данных. До сих пор многие информационные системы разрабатываются на основе того же подхода, который существовал в середине 70-х годов, однако с тех пор компьютеры стали в тысячи раз дешевле. Мы считаем, что при разработке современных систем следует принимать этот факт во внимание. В 1995 году мы заменили мейнфреймы серверами Sun-Oracle. Суммарная стоимость этого проекта, включая подготовку персонала, составляет 6-7 млн. долларов, причем на оборудование приходится менее 20%. Остальную сумму составляют затраты на подготовку персонала, загрузку данных в базы, тестирование, внедрение новых версий программ и другие расходы.

Может быть, лет через 5 или 10, когда надежность сетей повысится, когда сети станут дешевле, а управлять ими будет проще, все будет происходить по-иному, но сегодня я рекомендую именно тот подход, о котором говорилось выше. К сожалению, имеются приложения, которые не работают так, как нам хотелось бы, и поэтому приходится поддерживать и другие модели, но в целом наше стремление остается постоянным.

Не могли бы Вы объяснить подробнее переход от ориентации на транзакции к ориентации на заказчиков?

Основная тенденция, которую я наблюдаю в США, Японии, странах Западной Европы, состоит в том, что люди более не рассматривают информационные системы с позиций от-

делов, а трактуют их с позиций двух бизнес-процессов. Например, в компании Sun имеется два основных бизнес-процесса. Первый из них называется "оформление заказов", он начинается с того момента, когда продавец выписывает клиенту счет на заказ, и заканчивается тогда, когда будут получены деньги за этот заказ. Второй процесс называется "оплата покупки", он начинается с момента, когда Sun заказывает какие-то комплектующие у поставщика, и заканчивается, когда Sun выплачивает поставщику деньги за эти комплектующие. Мы стараемся построить среду управления, которая охватывала бы все этапы этих процессов. Таким образом, структурирование данных по заказам, счетам, заявкам на обслуживание оказывается не слишком полезным. Сейчас только в США мы получаем свыше полумиллиарда долларов в год в качестве платы за сервисное обслуживание организаций. Заказчики в США хотят видеть сервисное обслуживание на таком уровне, что если они звонят в компанию, то на все их вопросы должен ответить один консультант; нельзя отсылать их по разным вопросам к разным людям. Таким образом, мы должны предоставить консультантам полный объем данных о клиенте.

Когда заказчики звонят поставщикам по поводу размещения нового заказа, они хотят получить полную информацию; они хотят знать, был ли доставлен последний заказ, и если нет, то по каким причинам, каковы условия получения компенсации. В магазинах розничной торговли хотят иметь всю информацию буквально о каждой покупке каждого заказчика, хотят знать, что

вы представляете собой как клиент, в какие магазины ходите и что покупаете.

Сегодня сама идея, что вся информация находится в собственности подразделений и относится только к их деятельности, очень быстро отмирает. Люди хотят иметь доступ ко всей имеющейся информации. Например, если я планирую посетить важного заказчика, я хочу предварительно и очень оперативно получить всю информацию о том, имел ли он проблемы с доставкой, с сервисным обслуживанием, что он недавно заказывал, какой был последний сервисный вызов и т.п. Чтобы быть конкурентоспособными в 90-е годы, необходимо иметь всю эту информацию.

На сегодняшний день самый быстро развивающийся класс приложений в США — это хранилища данных. Хранилище представляет собой очень большую базу данных, в которой содержится вся информация о деятельности компании, о ее заказчиках и поставщиках. Все эти данные анализируются с целью оптимизации бизнес-процессов. Поэтому при построении серверов мы исходим из того, что в течение ближайших десяти лет объем баз данных ежегодно будет удваиваться.

Я встречался с клиентами, которые считают, что этого недостаточно, так как, по их мнению, объем данных будет увеличиваться ежегодно более чем в два раза. Я приведу пример человека, который осуществляет продажи по почтовым заказам. У него в компании информационная система в первую очередь идентифицирует звонящего заказчика по его телефонному номеру. Если это удастся — хорошо, если

нет, то она старается определить заказчика по его имени, индексу или телефонному номеру. Если это новый клиент, то система сможет проследить его кредитную историю до того, как будет введен его адрес. Когда клиент делает заказ, система успевает определить, достаточно ли клиент платежеспособен, чтобы приобрести у компании данный продукт. Если клиент не имеет достаточно средств, система осуществляет поиск в файле продуктов того ближайшего продукта, за который клиент в состоянии заплатить. Если клиент все же сделает заказ, то система предоставляет перечень дополнительных услуг, которые можно в этой связи оказать. Компания постоянно определяет, сколько времени тратится на обслуживание каждого конкретного клиента, и делает вывод, насколько клиент удовлетворен работой с компанией. Основой всего служит база данных в 14 гигабайт на платформе Oracle.

Подобная модель ведения бизнеса коренным образом отличается от той, что использовалась два года назад. Такая тенденция характерна даже для промышленных предприятий, когда обслуживающий вас сотрудник должен иметь полную информацию о вас. Эта тенденция важна для создания наиболее благоприятного микроклимата в банках и других финансовых учреждениях, чтобы при каждом обращении вас обслуживали наилучшим образом. Недавно в США произошло слияние двух крупных банков, и основной причиной слияния явилось то, что каждый из банков в отдельности не мог позволить себе затраты на подобную информационную систему.



Информационный
бюллетень Jet Info

Индекс по каталогу
РОСПЕЧАТИ - 32555

Главный редактор: В.А.Галащенко
Технический редактор: С.И.Демочкин

Полное или частичное
воспроизведение материалов,
содержащихся в настоящем
издании, допускается только
с разрешения Jet Infosystems

Jet Infosystems

Россия, 103006, Москва,
ул.Краснопролетарская, 6
тел. (095) 972 11 82
(095) 972 13 32
факс (095) 972 07 91
e-mail: JetInfo@jet.msk.su

Перед Вами один из номеров бесплатного периодического информационно-технического бюллетеня "Jet Info", издаваемого компанией "Инфосистемы Джет" - ведущим российским системным интегратором в области UNIX-систем.

Если Вы еще не являетесь подписчиком "Jet Info", но хотели бы получать его регулярно, просим Вас отправить в наш адрес заполненный купон этого объявления. Отправьте этот купон и в том случае, если Вы сменили свой почтовый адрес или решили изменить форму доставки Вам бюллетеня.

Информация для подписчиков: не забудьте в течение каждого декабря и июля обязательно отправлять нам этот отрывной купон для перерегистрации.



1. Организация _____

2. Фамилия, имя, отчество и должность лица, в чей адрес будет осуществляться рассылка _____

3. Индекс и почтовый адрес _____

4. Телефон и/или факс _____

5. Адрес электронной почты _____

6. В какой форме Вы хотели бы получать бюллетень

- печатный вариант по почте
 электронную версию по электронной почте

7. Прежняя форма и адрес подписки (для подписчиков, меняющих адрес и форму подписки, либо тех, кто проходит перерегистрацию) _____

8. Общее число компьютеров в организации _____

9. Ваша организация использует

Вычислительную технику

- Apple
 DEC Alpha
 DEC VAX
 IBM PC
 HP 9000
 Silicon Graphics
 Sun Microsystems
 другое _____

Операционные системы

- MS-DOS (Windows)
 Novell NetWare
 SCO Unix
 Solaris / SunOS
 UnixWare
 VAX VMS
другое _____

10. Материалы на какую тему заинтересовали бы Вас в первую очередь _____

Мы будем рады, если кто-то из Вас захочет принять участие в подготовке выпусков "Jet Info" в качестве автора. Будем также признательны за любые конструктивные замечания и предложения по всем аспектам подготовки, выпуска и распространения нашего бюллетеня.