

IBM Software Group



Превращая создание продукта в конкурентное преимущество:

СИСТЕМНЫЙ ИНЖИНИРИНГ



*Анатолий Волохов,
Software Group, Rational-Telelogic Solutions,
anatoly.volokhov@ru.ibm.com*

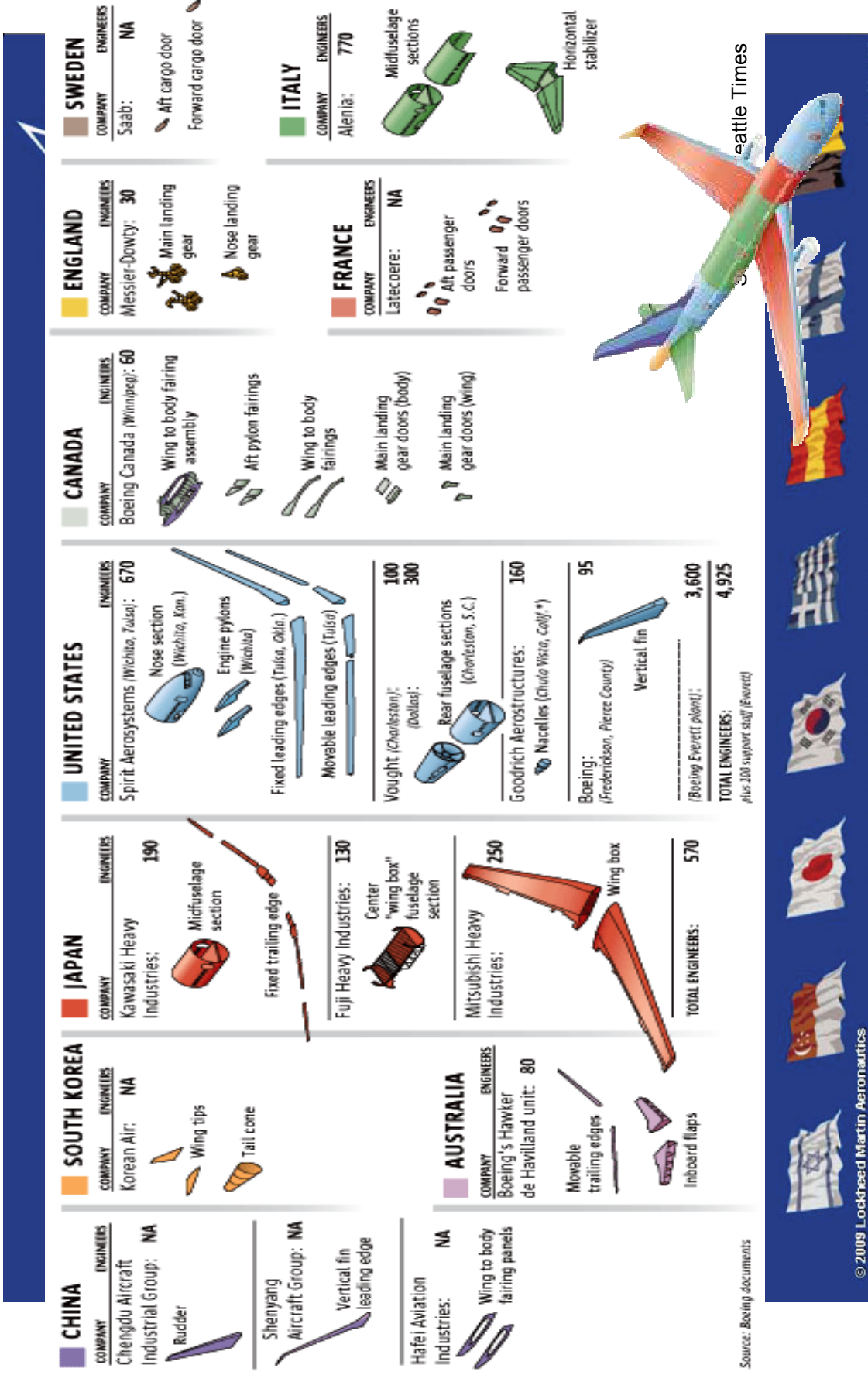
Системный инжиниринг (System Engineering). Определения

- **Системный инжиниринг** - ориентированный на изделие подход, отвечающий за создание и выполнение процессов, обеспечивающих удовлетворение нужд заказчиков и непосредственных пользователей изделия
- Системный инжиниринг – есть **междисциплинарный** подход, используемый для контроля за работками **сложных, инновационных** изделий и **систем** *
- **Система** – это набор компонентов (*которые и сами могут быть системами*), соединенных и связанных определенным образом так, чтобы обеспечить функционирование, которое сами по себе компоненты обеспечить не могут.
- **Летательный аппарат, атомная станция, судно, нефте- и газоперерабатывающий комплекс, программное приложение** – есть система.. и ее основные компоненты (*например, в отношении самолета - корпус, крылья, система управления, силовые установки, программное обеспечение...*), также являются системами (*или оборудованием*)





Самолет – это сложная авиационная система



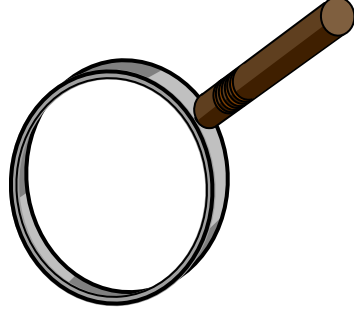
Source: Boeing documents

© 2009 Lockheed Martin Aeronautics

Зачем нужен Системный Инжиниринг

Системный Инжиниринг отвечает за **всю картину в целом**, обеспечивая выполнение требований в течение всего жизненного цикла изделия.

- Повышается вероятность успеха создания Системы
 - ▶ Понимание природы Системы и ее поведения в окружающей среде
 - ▶ Определение характеристик Системы с точки зрения пользователя
- Уменьшаются риски принятия неправильных решений
 - ▶ Выявление и оценка возможных рисков
 - ▶ Поиск неопределенностей и изменяемых параметров
 - ▶ Учет требований нормативных документов и регулирующих организаций
- Уменьшение общей стоимости жизненного цикла изделия
 - ▶ Улучшение процесса принятия решений в планировании, разработке, эксплуатации



- Такой подход начинается с понимания **потребностей заказчика**, определения **функциональности изделия** и обязательных **запланированных проверок** (*аттестаций, приемочных испытаний, контроля*) на самых ранних стадиях жизненного цикла создания изделия

К чему ведет неиспользование системного инжиниринга

Проблемы с проектами

- Только **28%** проектов отвечают запланированным срокам и бюджету

Прямые убытки

- Выпуск изделия на рынок всего **на 6 месяцев позже** может стоить компании **трети** планового пятилетнего показателя возврата инвестиций

Исправление ошибок

- Более **45%** бюджета на разработку, может «уйти» **на исправление и переделки**
- От **35 до 50%** общего объема работ тратится на исправление ошибок в дизайне
- Исправление ошибок, обнаруженных на этапе эксплуатации, обходится в **200 раз дороже** ошибок, обнаруженных на ранних этапах



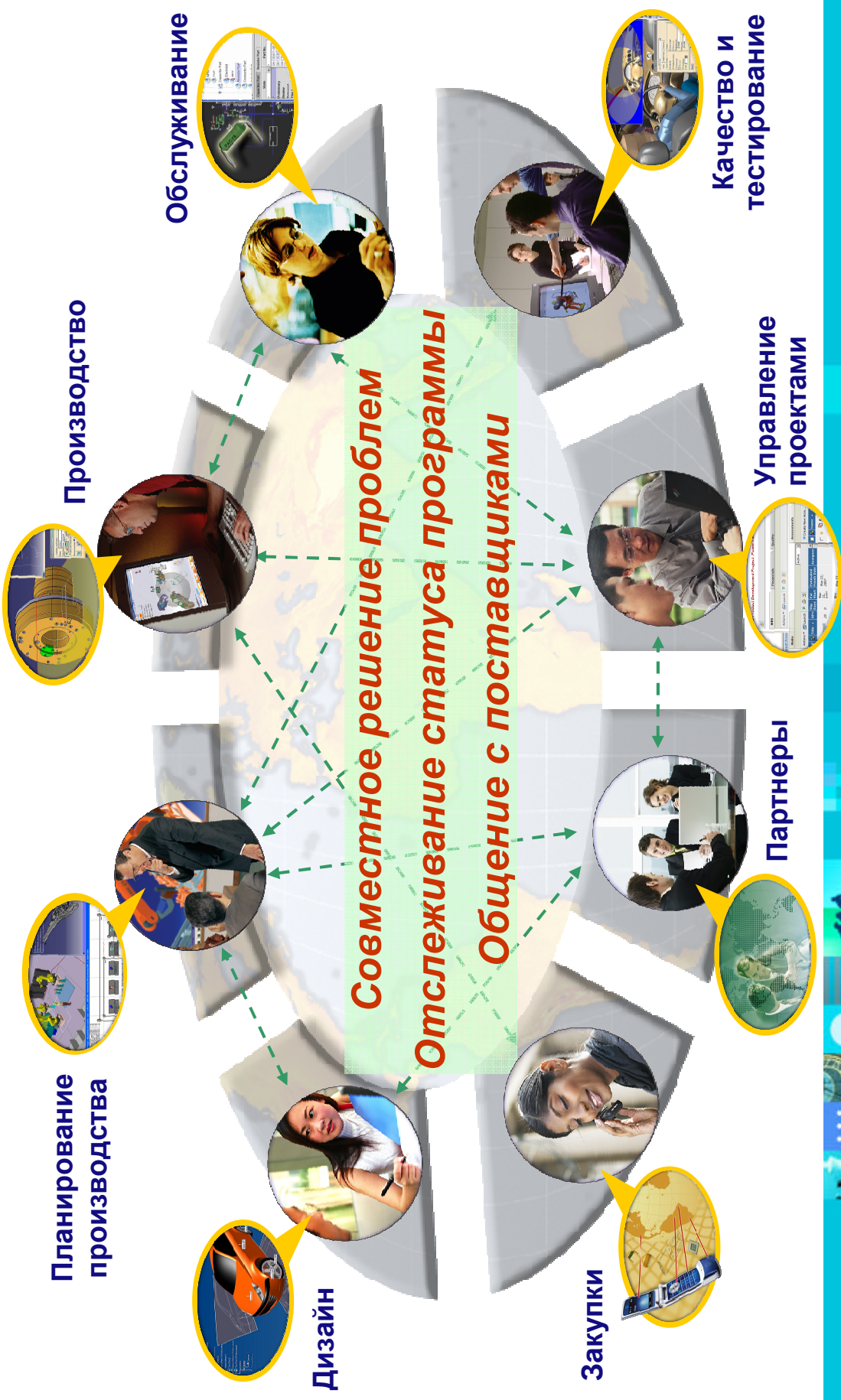
Sources:

- Don Reinertsen, McKinsey, 1983
- Standish Group, 2001
- Leffingwell & Widrig, "Managing Software Requirements," Addison Wesley, 1999
- Effective Requirements Practices, NASA
- IAG Consulting, 2008
- Dynamic Market Limited, 2007



Почему это происходит?

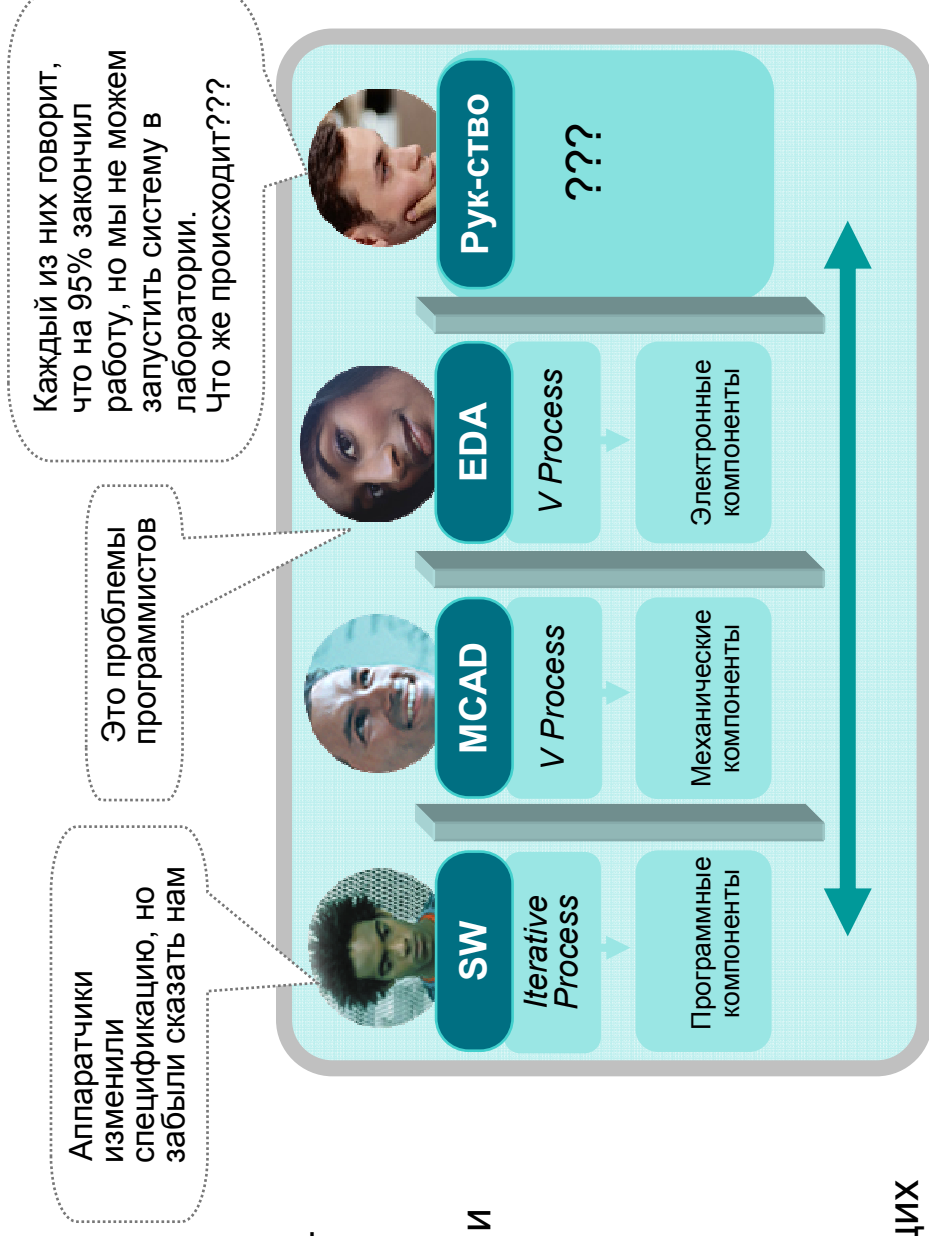
Пользователи разделены разными приложениями и средствами коммуникации



Почему это происходит?

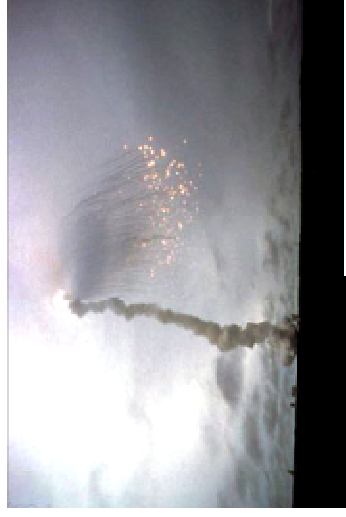
Вертикальная структура организации разработки – уже большая проблема

- Несогласованность процессов и правил
- «Know-how» – только в пределах своей вертикали - снижение общей производительности
- Сложность сотрудничества и обмена данными и результатами
- Заниженные возможности отслеживания оперативных решений и меняющейся информации
- Низкий уровень повторного использования существующих решений и артефактов



Аварии все еще продолжают беспокоить производителей и PLM не является панацеей от всех бед..

- **Аэрокосмическое агентство**
На 40с полета бортовой компьютер прототипа стратегической ракеты стоимостью \$1 млрд. ошибочно выдал команду на самоуничтожение
- **Производитель автомобилей**
Компания вынуждена была отозвать 75 тысяч автомобилей, которые из-за ошибки в ПО вдруг начали тормозить на большой скорости
- **Производитель мед. оборудования:**
Из-за некачественного ПО пришлось изымать из эксплуатации 42 тысячи дефибрилляторов



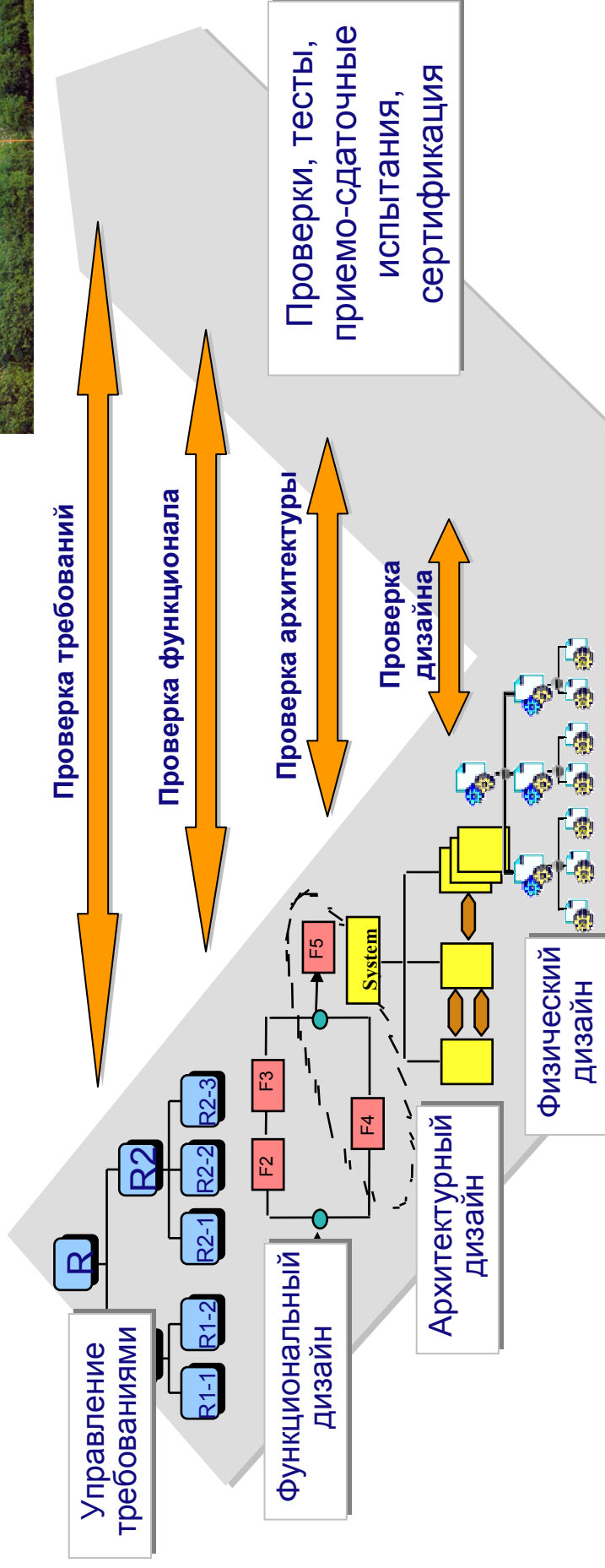
Решения для системного инжиниринга

Системный инжиниринг требует взаимодействия множества инструментов и дисциплин:

- Управление совместными бизнес-процессами:
 - Управление портфелями
 - Управление программами
 - Управление требованиями
 - Управление изменениями
 - Управление конфигурациями
 - Управление поставками
 - Управление потоками задач
 - Контроль за соблюдением стандартов
- Моделирование и тестирование систем и изделий (*в зависимости от их типа*):
 - ▶ UML & SysML
 - ▶ «Белый» ящик
 - ▶ Rhapsody
 - ▶ «Черный» ящик
 - ▶ Simulink
 - ▶ Нагрузочное
 - ▶ Modelica
 - ▶ Функциональное
 - ▶ MCAD & ECAD
 - ▶ Автоматизация



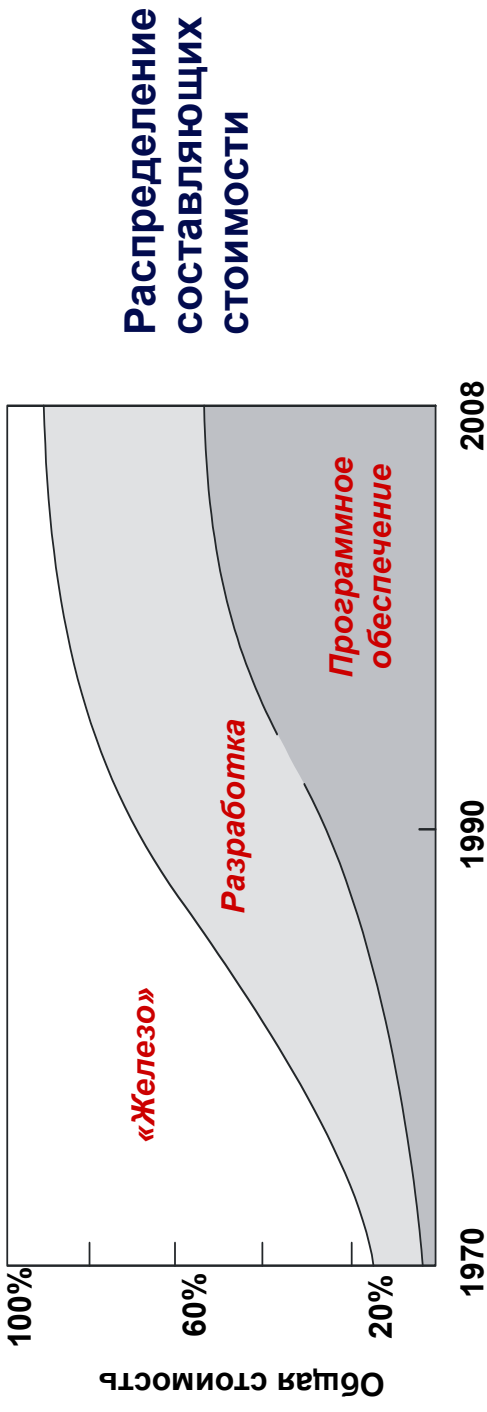
V-модель - ИЗВЕСТНЫЙ ПОДХОД.



- Разработка продукта в полном соответствии с требованиями
- Учет изменений на всех уровнях разработки
- Тесты, проверки, сертификация проверят требования
- Обеспечивается сквозной мониторинг производства продукта
- **Конечный продукт соответствует требованиям на все 100%**



Значительно возрастает доля программной составляющей и это ведет к существенным проблемам

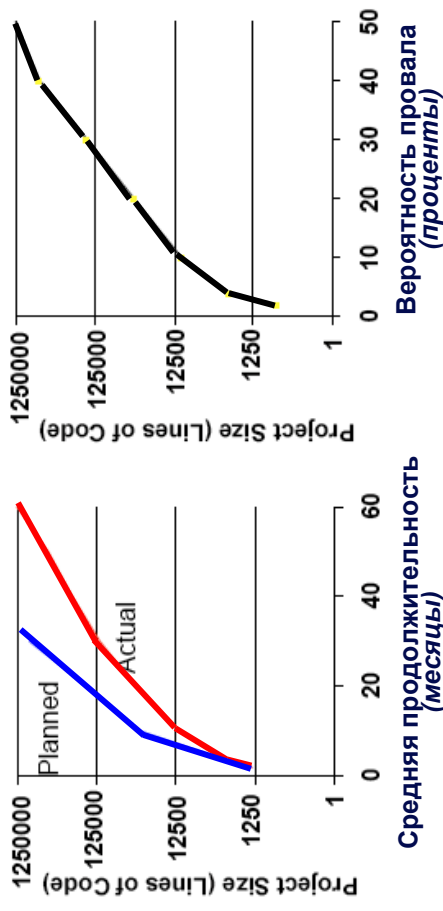


Рост софтверных составляющих

Тип самолета	Год	Функции, контролируемые ПО (%)
F-4	1960	8%
A-7	1964	10%
F-111	1970	20%
F-15	1975	35%
F-16	1982	45%
B-2	1990	65%
F-22	2000	80%

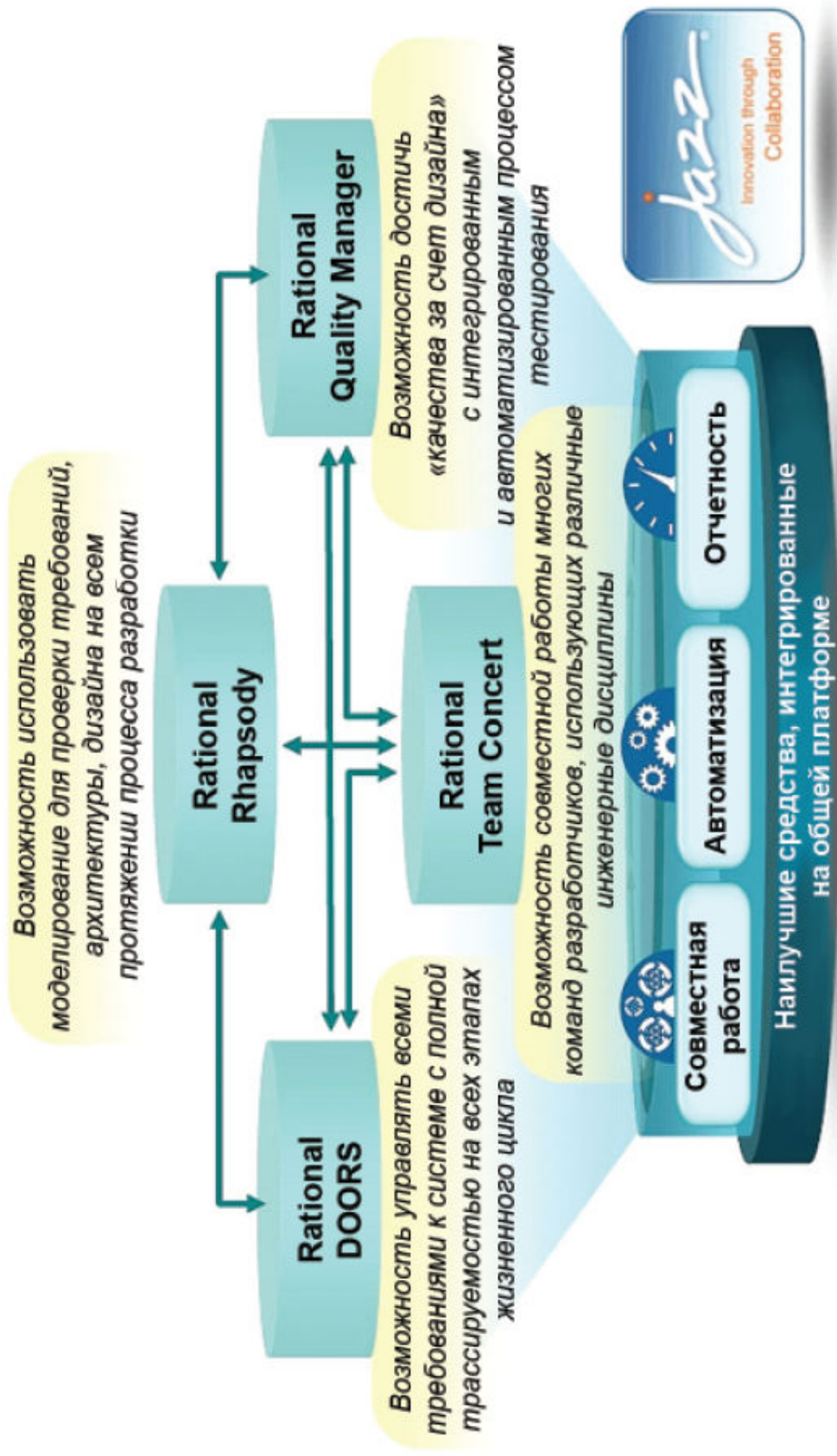


Влияние на бизнес





Инструментальная поддержка V-модели



IBM Software Group



Превращая создание продукта в конкурентное преимущество:

Инжиниринг требований



*Анатолий Волохов,
Software Group, Rational-Telelogic Solutions,
anatoly.volokhov@ru.ibm.com*

Разработка без управления требованиями – непредсказуемый результат

Если вы еще только начинаете задумываться об улучшении ваших процессов, то помните, что начать стоит именно с **процесса управления требованиями**, потому что здесь действует простой принцип : *что посеешь, то и пожнешь*.

Если требования плохие, то все остальные ваши усилия, процессы и инструменты лишь помогут вам как можно быстрее создать **плохой продукт**.



К чему все это приводит ...

... и как найти выход из этого нерадостного положения?



Проблемы бизнеса

Продукт не удовлетворяет заказчика	46%
Поздний выход на рынок / упущенный спрос	33%
Слабая коммерциализация / раскрутка	26%
Качество продукта	24%
Ценовая политика	23%
Нечеткое позиционирование продукта	19%

The CIO's Guide to the PERFECT Launch: Translating Innovation to Business Benefit, AMR Research, 2005



Организационные возможности

Улучшить связь и взаимодействие между дисциплинами / доменами 71%

Повысить доступность требований 49%

Уметь прогнозировать поведение системы до тестирования 46%

Внедрить новый или переделать имеющийся процесс разработки, чтобы охватить множественные дисциплины / домены 43%

(... что-то, не имеющее отношения к данной теме ...) 39%

Aberdeen Group, System Design: New Product Development for Mechatronics, Michelle Boucher, David Houlihan, January, 2008



Вся наша жизнь – это работа с требованиями..... В БЫТУ

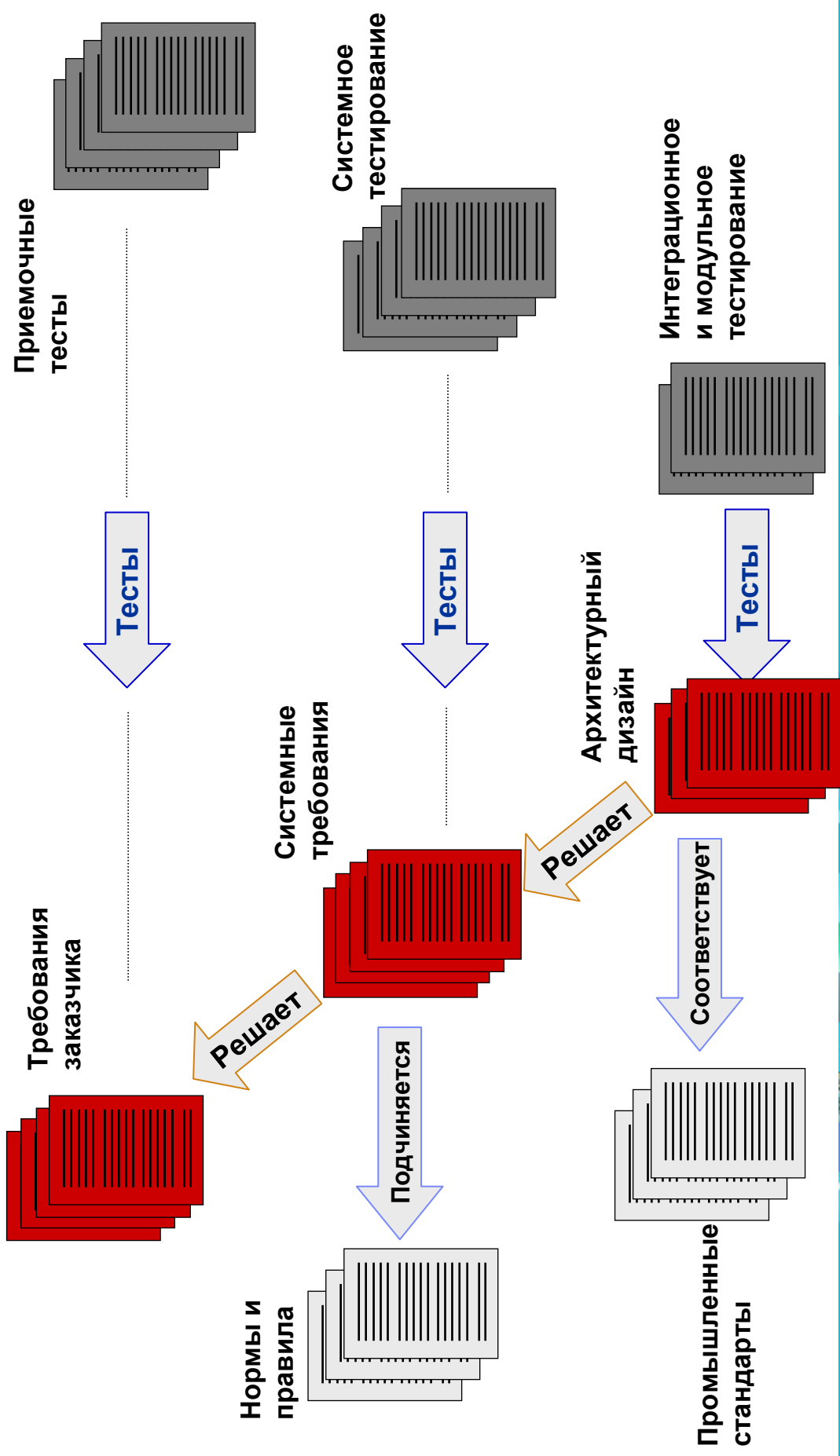
- Если жена звонит вам и просит по пути с работы домой купить – хлеб, сахар, масло, молоко и вино - то это ничто иное, как **сбор и формирование требований**
- Если в разговоре выясняется, что она имела ввиду:
 - Хлеб белый – 1 батон
 - Масло растительное - 1 бутылка
 - Сахар-песок– 2 пачки
 - Молоко топленое – 1 литрто это ничто иное, как **декомпозиция и детализация требований**.
- Разговор заканчивается тем, что вы приходите к выводу, что вино вы покупать не стоит, потому что в гости придет друг, которому пить противопоказано ... - это ничто иное, как работа с ограничениями или **учет ограничивающих факторов**
- Если вы решаете, что хлеб, масло и сахар вы купите в одном магазине, а вот молоко в другом - это выглядит как **структуризация требований**
- Если чуть позже жена вновь звонит вам и говорит, что она передумала и просит вместо молока купить сметану, то это **изменение требований**
- Если уже дома вы сверяете купленное с тем списком, что диктовала жена, - это **проверка реализации требований** или тестирование
- И если обнаруживаете, что вместо сметаны вы все-таки купили молоко - это значит, что вы **не учли изменение**, ранее внесенное в одно из требований





Вся наша жизнь – это работа с требованиями.... НА РАБОТЕ

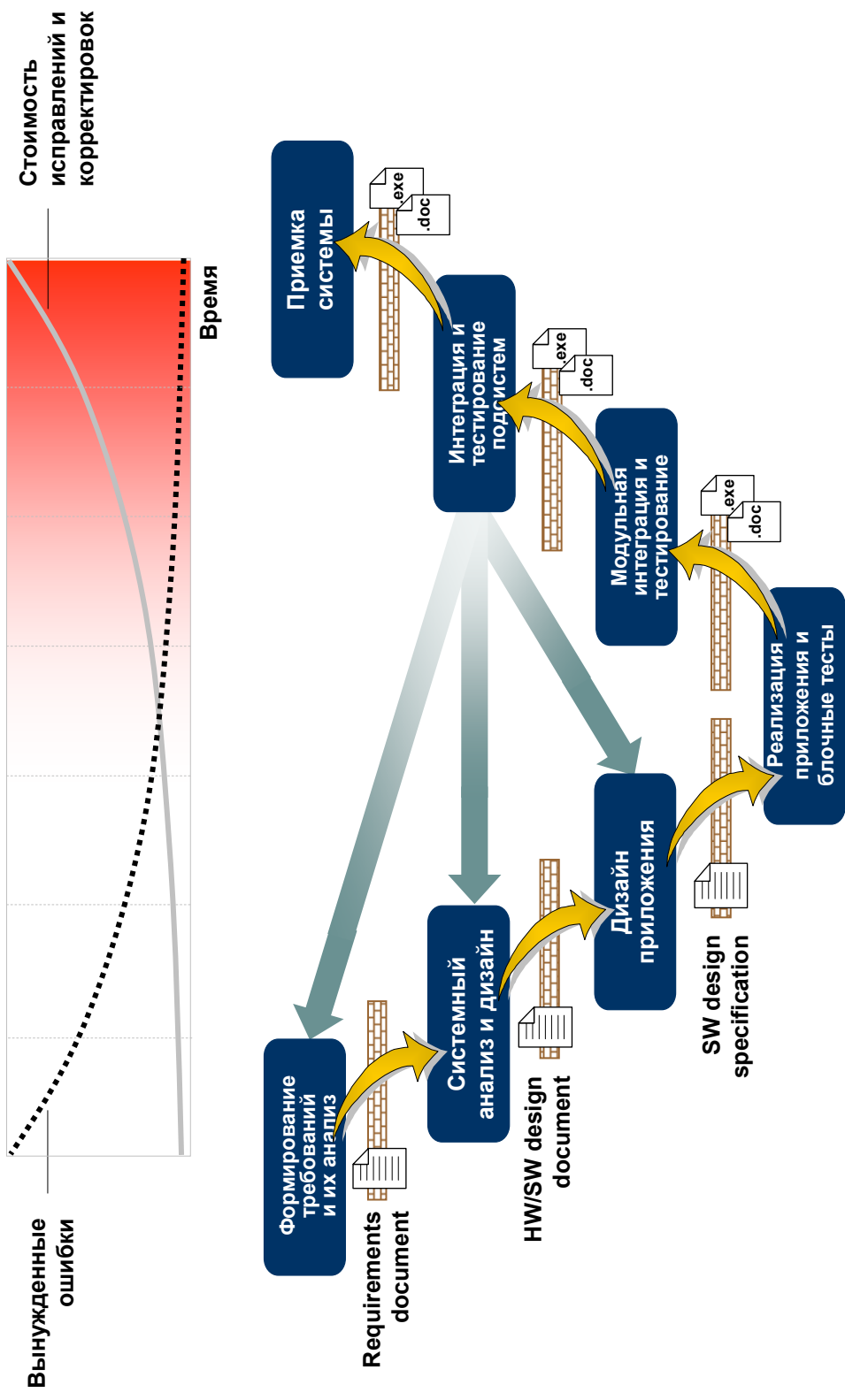
упрощенная модель проекта





Но не все так просто ...

Люди с разным мышлением с трудом понимают друг друга



Инжиниринг требований – составная часть системного инжиниринга

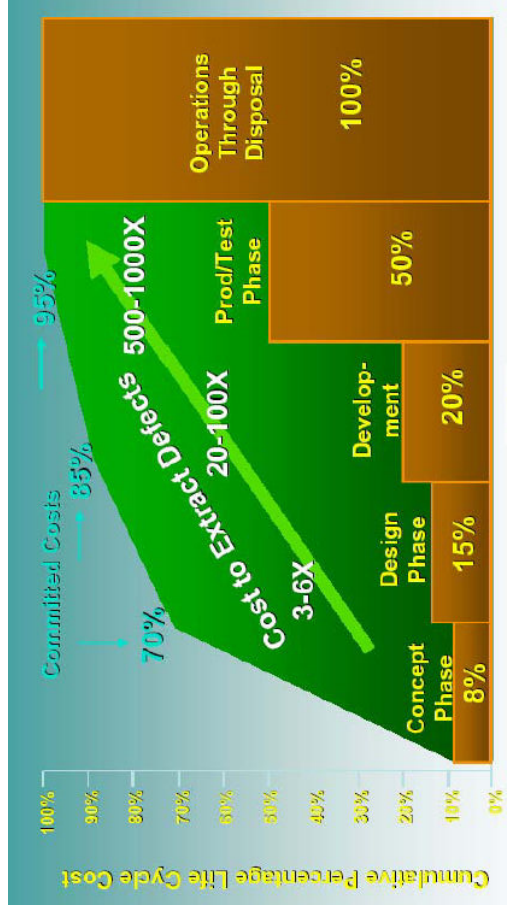
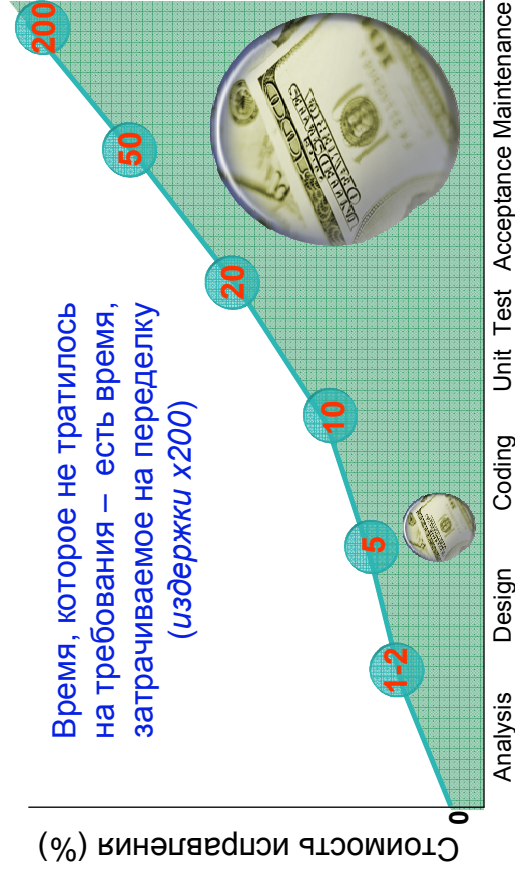
Более **80%** разработок заканчиваются плачевно только из-за неудовлетворительного формирования требований, их анализа и управления им

IDC, November 2007

Почти **80%** ошибок вносятся на стадии формирования требований

NASA, 2006

Проблемы в работе с требованиями ведут к излишним доработкам и переделкам, плохому качеству, задержкам и провалу проектов



Source:

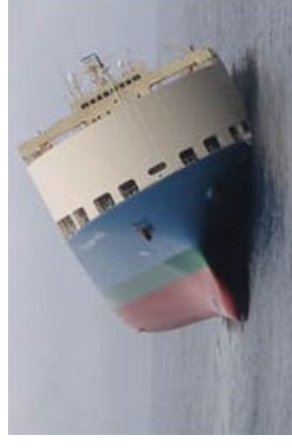
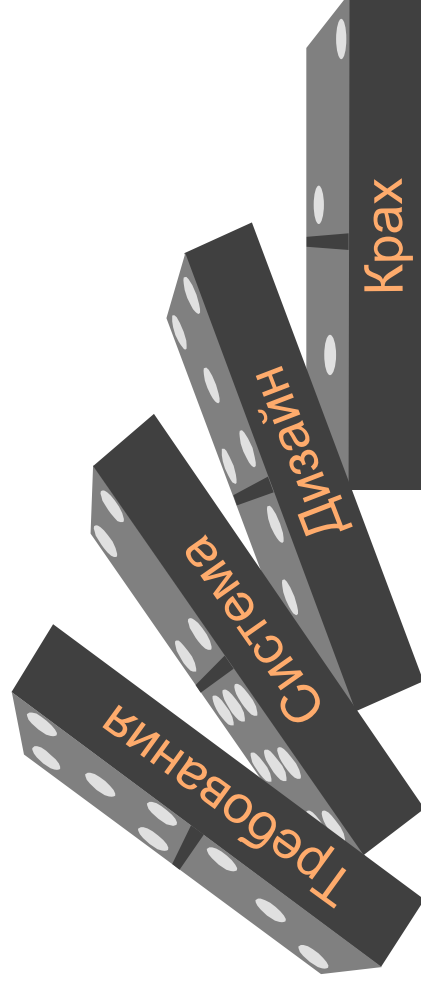
- Defense System Management College, DAU, 2004



Очень важен правильный старт и эффективная работа

- Около 60%-70% общего числа всех проектов заканчиваются плачевным результатом только из-за неудовлетворительного формирования **требований**, их анализа, управления и контроля

- *Meta Group, 2003*

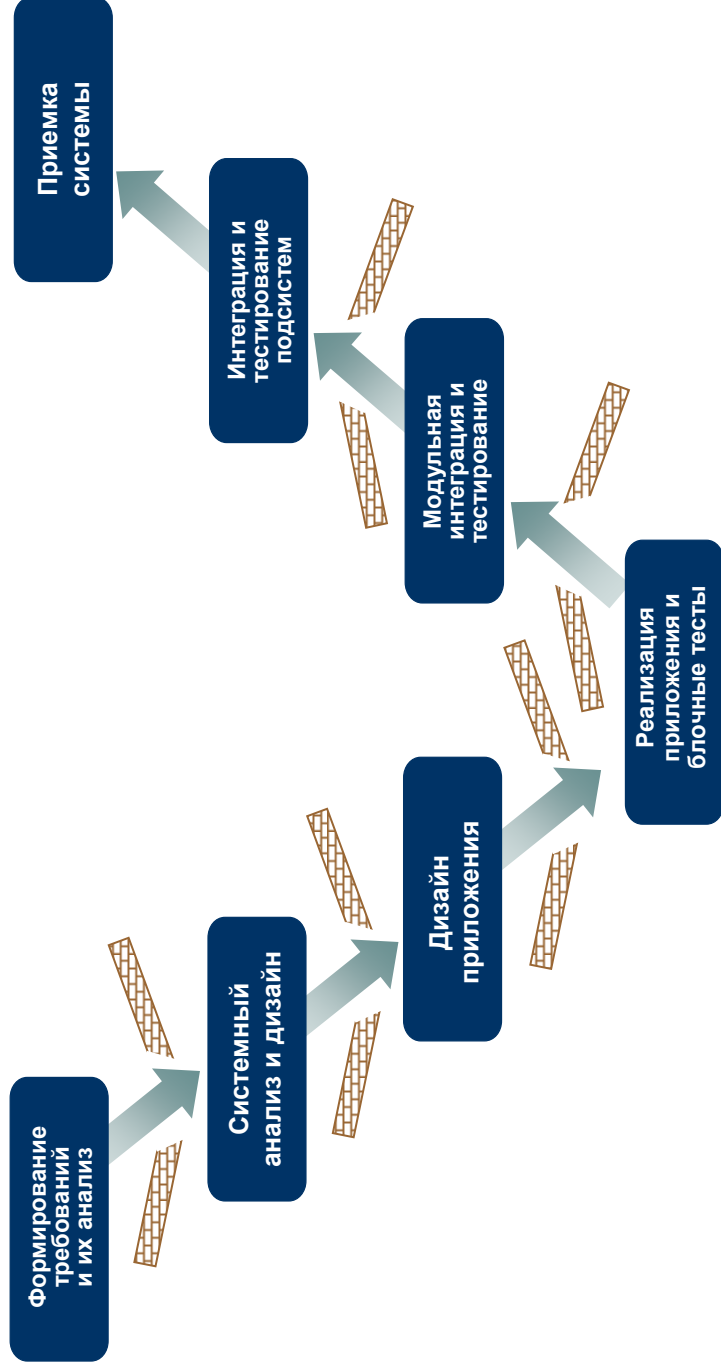


- **Задержки** - средний проект «опаздывает» на 220%
- **Неэффективная работа** - 40-50% времени специалиста тратится на задачи, не связанные с выполнением его непосредственных обязанностей: *поиск нужных документов, отслеживание изменений...*

Инжиниринг требований ломает барьеры

Инжиниринг требований

Тест инжиниринг



Необходим системный подход к работе с требованиями, чтобы производить успешный и прибыльный продукт

Инжиниринг требований =
 Формирование требований + Управление требованиями

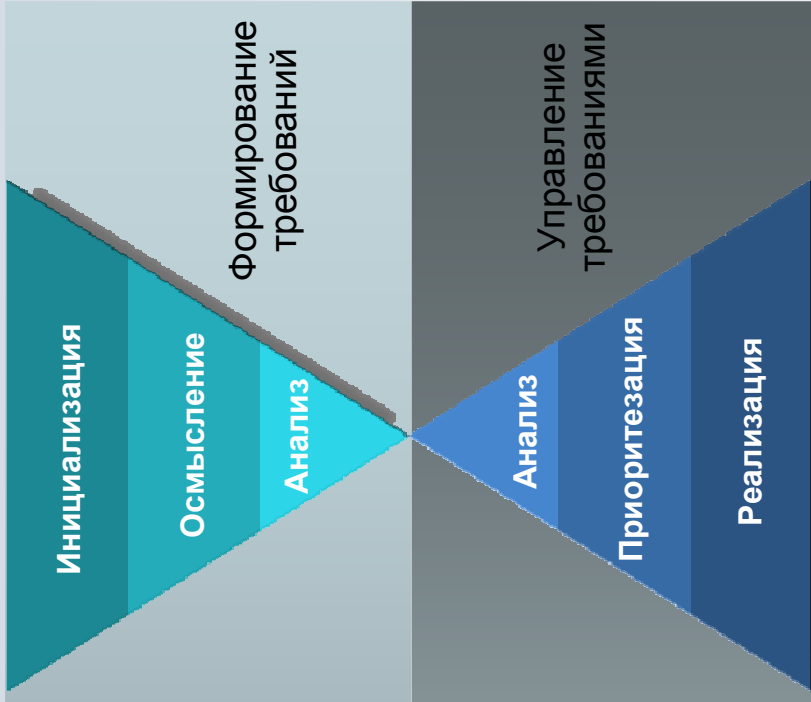
Are we solving the *right* problem ?

Собирать, извлекать, фиксировать, преобразовывать, конкретизировать, обсуждать, анализировать требования, используя различные подходы, метода и нотации

Возможность специалистам по бизнесу работать над требованиями вместе с технологическим экспертами

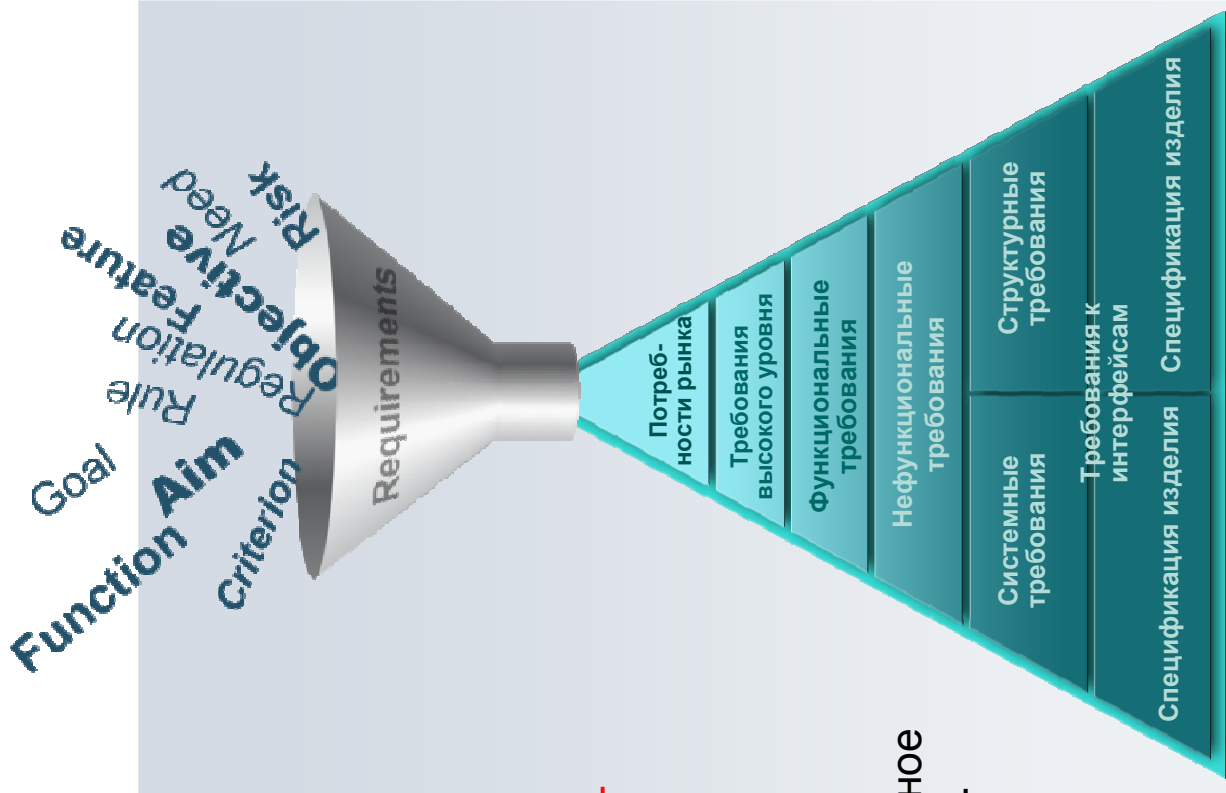
Are we solving the problem *right* ?

Систематизировать и структурировать требования, строить взаимоотношения между ними, используя атрибутику, линкование и трассировку. Контролировать и анализировать изменения и управлять ими



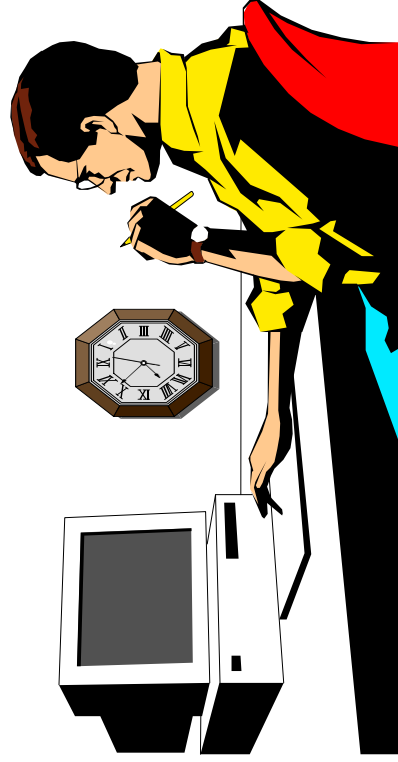
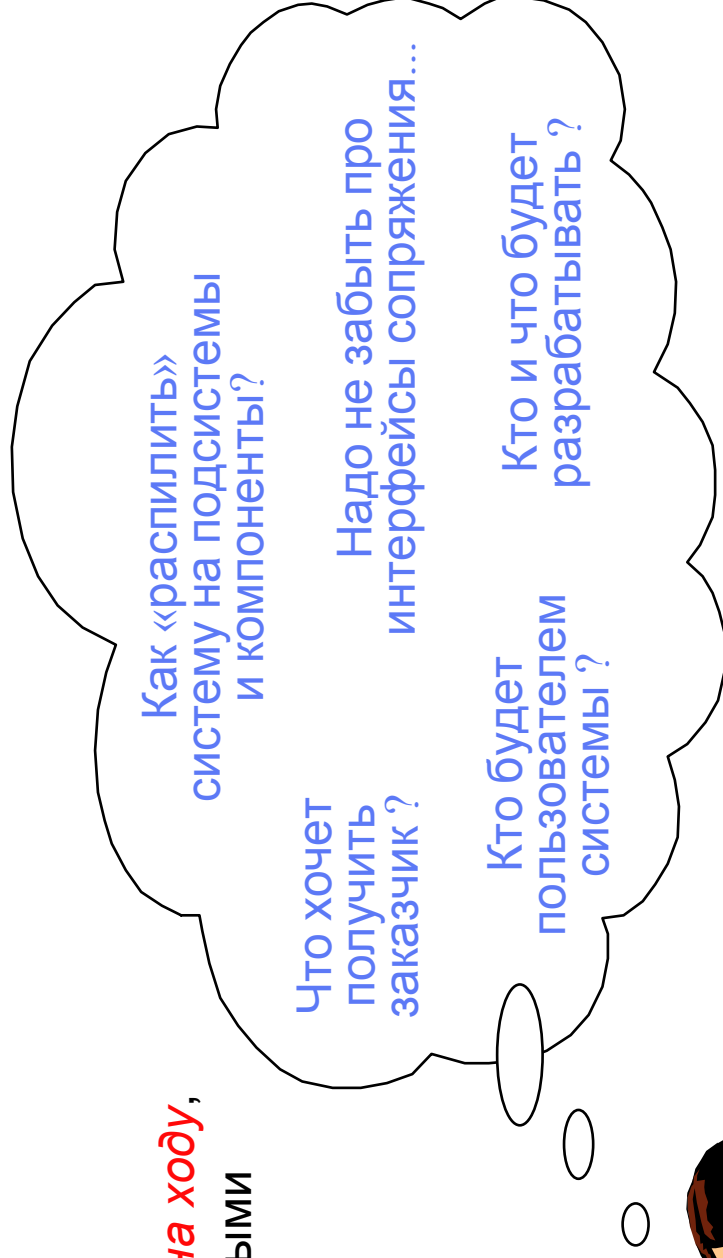
Что такое “требование” ...

- Требование - есть единичная задокументированная необходимость
- Требование - описание того, каким должен быть определенный продукт
- Описание того, **ЧТО каждый компонент ДОЛЖЕН ДЕЛАТЬ** и **КАК компоненты будут ВЗАИМОДЕЙСТВОВАТЬ**
 - ▶ Функциональные требования описывают точное поведение (*функционалирование*) системы, т.е. «ЧТО система должна делать»
 - ▶ Нефункциональные требования описывают насколько хорошо это поведение должно исполняться (но избегайте слова – КАК)



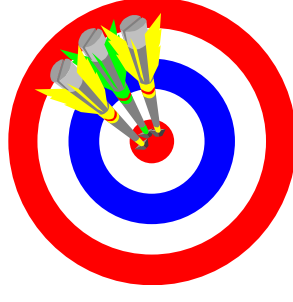
Успешный проект должен получить свои требования до начала работ по его реализации

Решения, принимаемые **на ходу**, не могут быть оптимальными



За ошибки, сделанные ВЧЕРА, и исправления, вносимые СЕГОДНЯ, все равно **придется расплачиваться ЗАВТРА**

Признаки хорошего требования

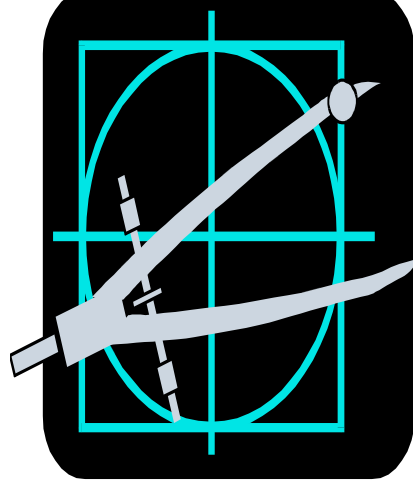


Каждое индивидуальное требование должно быть:

- **Корректное** с технической и юридической точек зрения
- **Полное** выразить утверждение или законченную идею
- **Четкое, однозначное** недвусмысленное и не сбивающее с толку
- **Совместимое** согласующееся и не конфликтующее с другими требованиями
- **Проверяемое** чтобы подтвердить, что результат соответствует требованию
- **Трассируемое** уникально идентифицированное и отслеживаемое
- **Выполнимое** чтобы реализоваться в рамках запланированного бюджета и сроков
- **Модульное, блочное** изменяться без чрезмерных последствий для всего проекта
- **Инженерно-независимое** не должно содержать описания конкретного решения
- **Позитивное** сформулировано в утвердительной форме

Как создавать хорошие требования...

- Каждое требование должно выглядеть как законченное предложение, содержащее **подлежащее и сказуемое**, и при этом :
 - ▶ отражать предметную принадлежность (*требование относится к пользователю или к системе*)
 - ▶ содержать утверждение (*логическое условие, действие, предполагаемый результат*)
- Формулируя требование, необходимо использовать :
 - ▶ либо глагол **должен**, когда требование является обязательным,
 - ▶ либо глагол **может**, когда требование является дополнительным или факультативным
 - ▶ возможны и вариации этих глаголов, но при соблюдении смысловых мер предосторожности
- Законченное требование должно точно формулировать **конечную цель** или определять **желаемый результат**
- Требование должно содержать критерии и оценки его успешной реализации или другие аналогичные индикаторы качества, которые можно было бы измерить: **невозможно контролировать то, что нельзя измерить**



Анатомия хорошего требования пользователя

Указывается предметная принадлежность

Используется определенный глагол

Неподготовленный пользователь должен иметь возможность создать отчет менее, чем за 3 минуты

Декларируется позитивный конечный результат

Измеряемый критерий производительности

Наибольшая проблема – суметь прописать **ВСЕ** эти составляющие для **КАЖДОГО** требования, которое вы формулируете

Требования должны быть структурированы... А зачем?

Требования должны быть структурированы, чтобы можно было увидеть:

- Контекст - *общие характеристики среды, к которой относятся требования*
 - ▶ Позволяет охватить взглядом всю картину целиком
- Дублирование требований... и избежать этого, чтобы:
 - ▶ Не выполнять дважды одну и ту же работу
 - ▶ Избежать конфликтных ситуаций на стадии разработки
 - ▶ Не способствовать удорожанию стоимости поддержки продукта в последующем
- Пропущенные требования
 - ▶ Отсутствие требования ведет к потере функциональности системы
 - ▶ Может привести к изъянам в области реализации нефункциональных требований (напр., *производительность, надежность, простота использования и т.д.*) – которые практически уже не поддаются исправлению, если проект завершен и система создана.

Помните эффект домино??
Это начинается здесь!!!



Пользователь #1.

Оглавление документа с требованиями

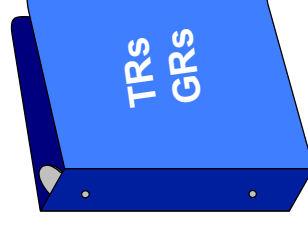
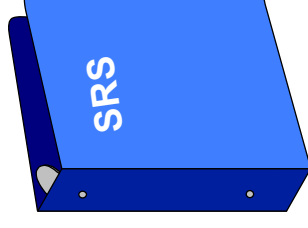
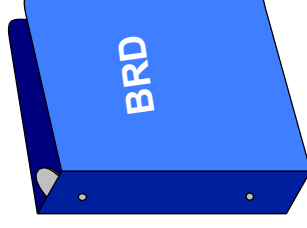
- 1.0. Общее описание
 - 1.1. Характеристики продукта
 - 1.2. Контекстные или системные диаграммы и схемы
 - 1.3. Условия эксплуатации
 - 1.4. Характеристики пользователя
- 2.0. Допущения и зависимости
- 3.0. Специфические требования
 - 3.1. Функциональные требования
 - 3.2. Нефункциональные требования (в порядке важности)
 - 4.0. Верификационные замеры и проверки
 - 5.0. Заметки (дополнительная информация, не имеющая отношения к требованиям)



Пользователь #2.

Типы документов с требованиями

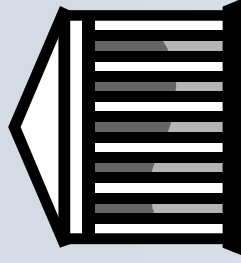
- Business Requirements Document (BRD)
- Market Requirements Document (MRD)
- User Requirements Document (URD)
- Statement of Work (SOW)
- Operational Concept Document (OCD)
- Interface Control Document (ICD)
- System Requirements Specifications (SRS)
- Technical Requirements Specification (TRS)
- **Constrains & Restrictions Document (CRD)**



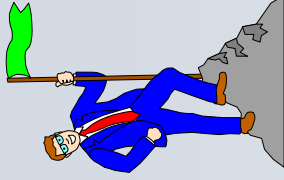
Источники требований. Сложные системы получают требования из многих источников



Пользователи



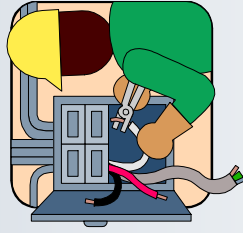
Бизнес



Конкуренты



Заказчики



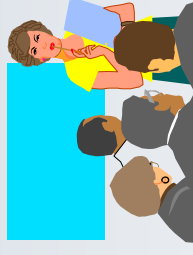
Эксплуатация



Sales



Help Desk

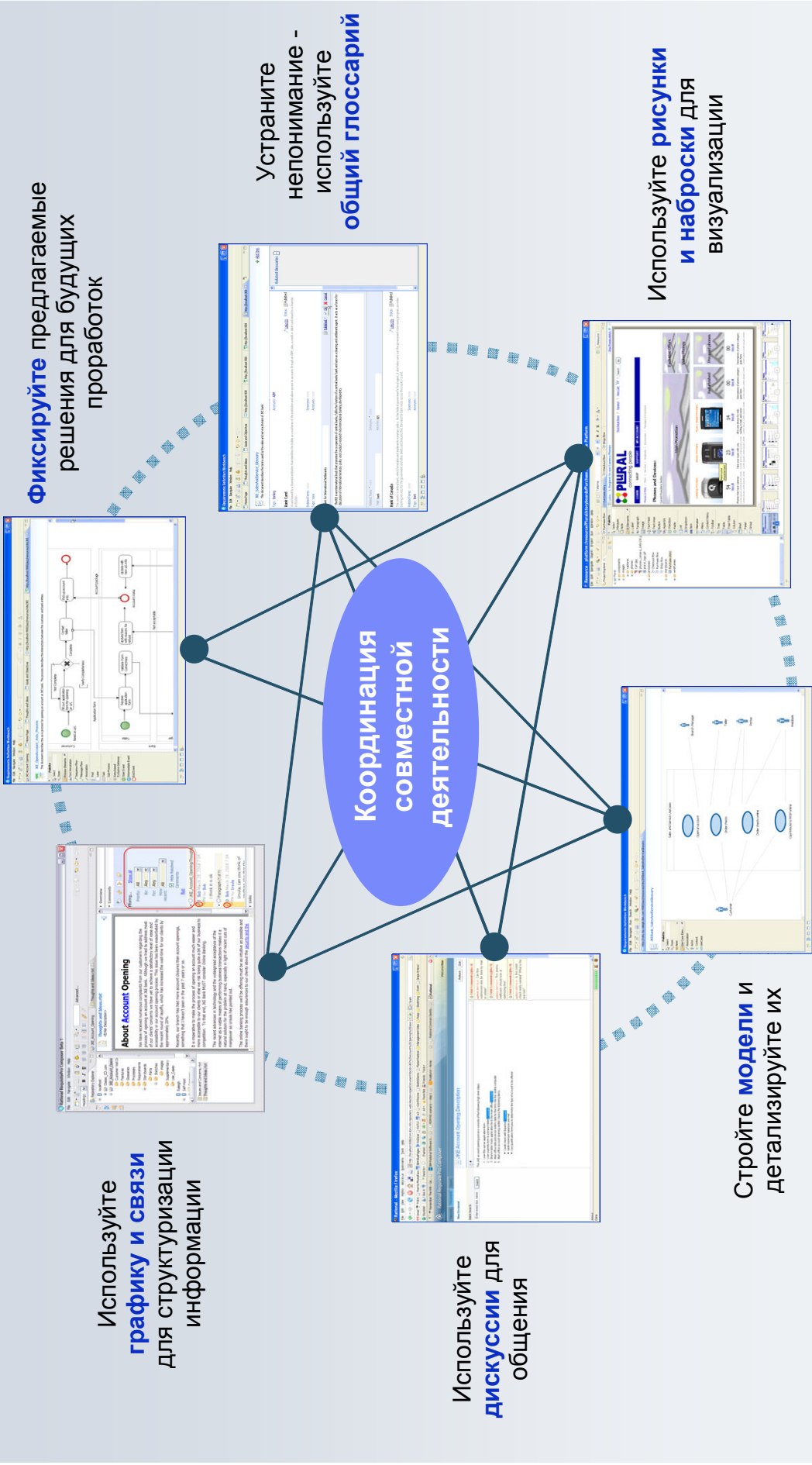


Обучение

Принимайте во внимание мнение **ВСЕХ** возможных пользователей.
Даже **ОДНО** неучтенное требование может привести к большим
проблемам или печальному результату



Эффективное формирование требований основывается на совместной работе, объединяющей различные точки зрения



Совет: как создавать требования

- Принимайте во внимание различные источники (внутренние, внешние, Web). Идентифицируйте типы и группы пользователей. Общайтесь с каждым.
- Попытайтесь заставить пользователя выразить свои мысли в терминах процессов и данных, используемых ими на каждом шаге разработки
- Записывайте каждое требование как полное предложение, сформулированное в утвердительное форме
- Не забывайте о прошлых ошибках и старайтесь обойти их хорошими и простыми альтернативными вариантами требований
- Постарайтесь выяснить природу возникновения требования.
Не стесняйтесь на некоторые требования заказчика спросить - **ПОЧЕМУ?**
- Никогда не оставляйте попыток улучшить формулировку требования.
Остановитесь только когда каждый скажет, что понял, что имеется ввиду
- Не жалейте времени сформулировать требование как можно более однозначно и недвусмысленно. Скорость работы многих специалистов - **одна страница в час**. Потратьте и вы хотя бы столько же времени, чтобы создать хороший документ с требованиями – это окупится сторицей.

Мы часто слышим – а зачем нужно управлять требованиями?

Попытаемся ответить на это вопрос вашими же словами:

- Текущий статус проекта никогда не ясен до тех пор, пока мы не поймем, что уже опаздываем и не укладываемся в плановый график
- Создается впечатление, что техническое задание всегда изменяется в самое неподходящее время
- Изменения требуют много внимания и времени и обходятся нам очень дорого
- У нас в компании большие проблемы в общении между подразделениями – трудно понять кто, что хочет и почему
- Довольно часто случается, что нам приходится переделывать наш продукт, что обходится в немалую копеечку..
- Наблюдаются большие проблемы с тестированием некорректно сформулированных требований
- У нас нет полной уверенности в том, что наши тесты проверяют все модули и подсистемы продукта
- Процесс тестирования требует слишком много времени и денег
- В свои требования наши заказчики зачастую закладывают и решение
- Мы испытываем большие трудности при попытке организовать требования в управляемую и контролируемую группу



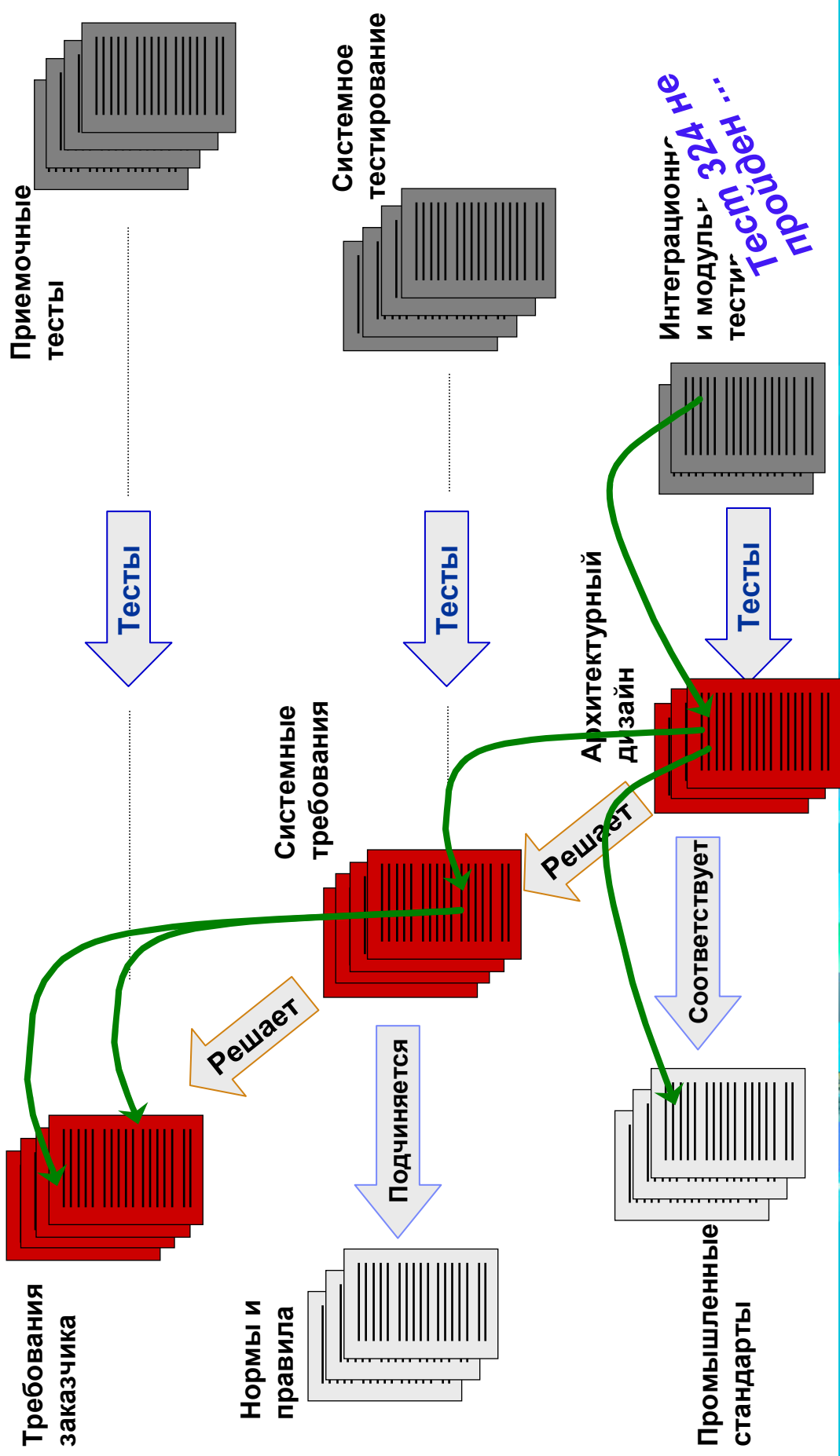
Управление требованиями дает огромные преимущества

- **Информированность** – ясное понимание целей и задач разработки
- **Прозрачность** – руководство может видеть общую картину и статус проекта
- **Тестируемость** – известно что тестировать, чтобы сдать продукт заказчику
- **Интеграция** – все отдельные блоки и модули наконец-то работают вместе
- **Трассируемость** – прозрачность отношений между требованиями
- **Управление изменениями** – оценка последствий вносимого изменения
- **Оптимизация** – разрабатываем и поставляем только то, что заказывалось
- **Качество** – мы хорошо понимаем, как много это значит для бизнеса
- **Удовлетворение** - заказчик и бизнес получают то, что хотели
- **Соответствие** – демонстрация соответствия нормативным документам
- **Анализ** - возможность оперативного принятия решений



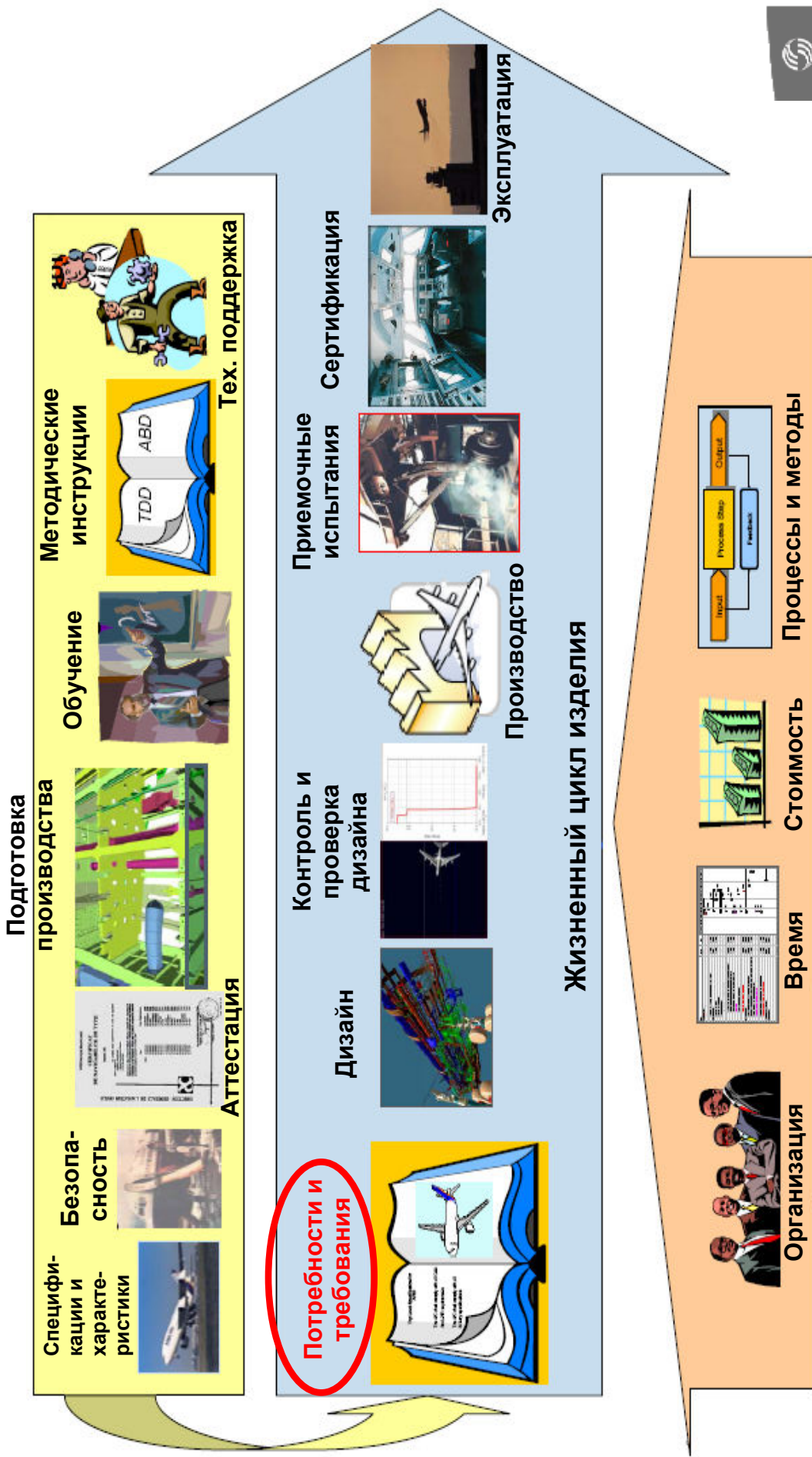


Трассировка (Traceability Analysis)



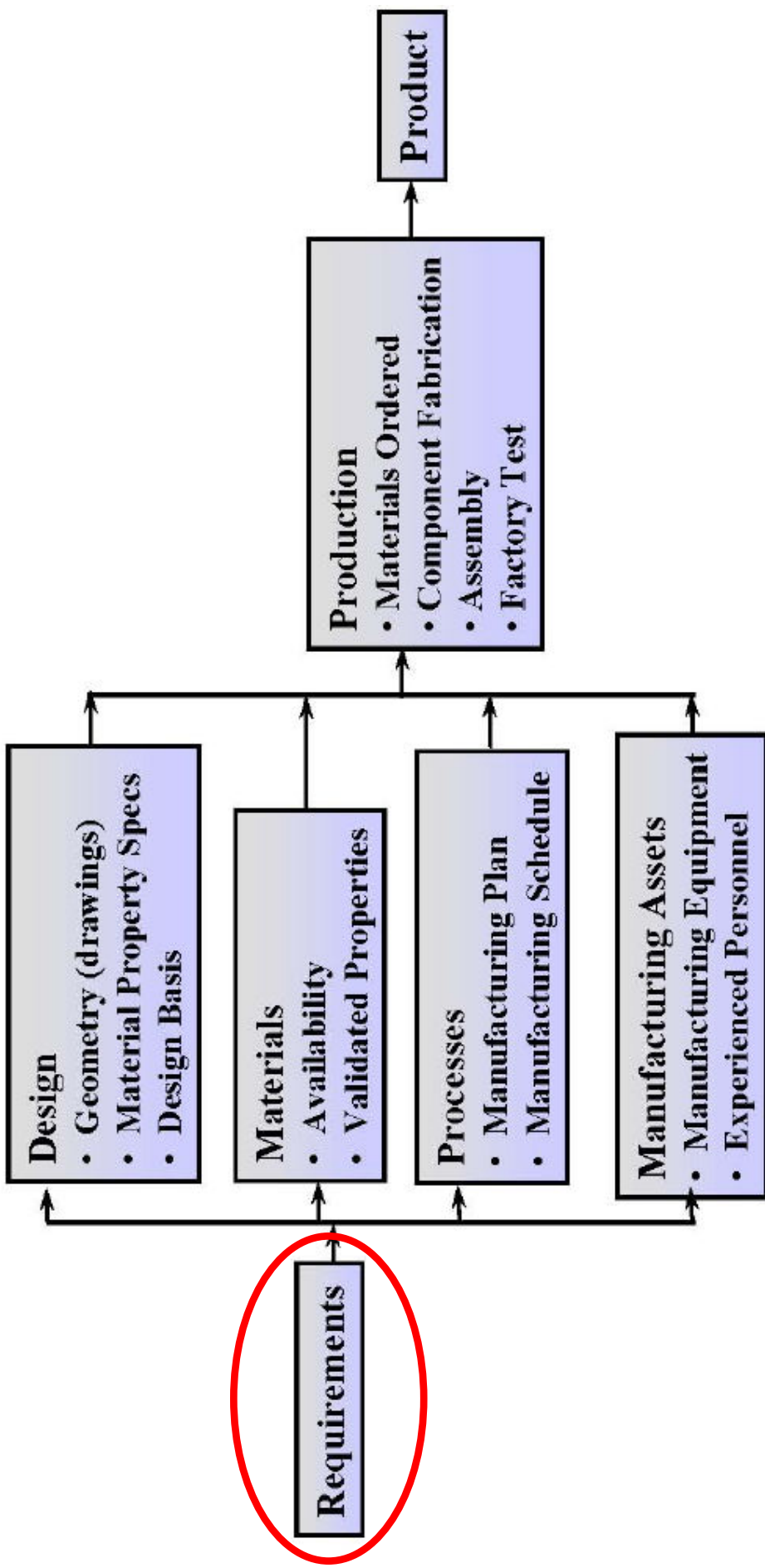
Как это выглядит в AIRBUS

С 2003 года **System Engineering** называется в Airbus - **Requirements Based Engineering**





Как это выглядит в BAE SYSTEMS

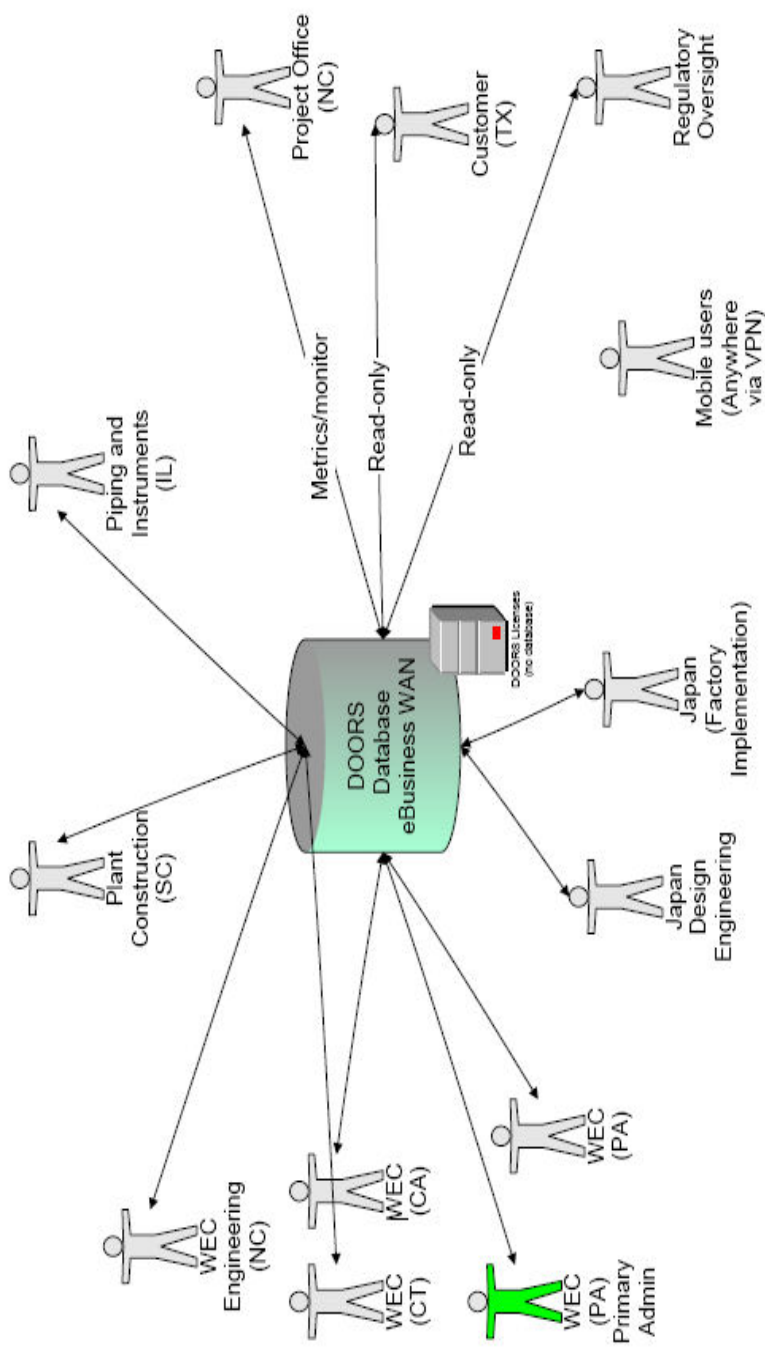




Как это выглядит в Westinghouse Electric

Innovation - Unified RM Database

- Westinghouse is involved in a global, multi-company project where requirements management is a collaborative event
- Hosting a VPN-protected, eBusiness [DOORS database](#) where users from various companies work together
- DWA offers author/reviewer functionality, offering consistent user experience across the geographically dispersed team



Westinghouse Electric устами заказчика.....



Подразделение инструментального контроля компании **Westinghouse Electric** использует **Rational DOORS** для **управления требованиями** в своих новых проектах или проектах по модернизации объектов атомной энергетики по всему миру.

Тысячи требования заказчиков, а также регулирующих, нормативных и лицензирующих требований контролируются и отслеживаются для того, чтобы разработать и создать системы высочайшего качества.

Использование DOORS дает компании Westinghouse возможность более эффективно управлять изменениями требований, создавать конструкторскую документацию высокого качества, создавать прозрачную отчетность для регулирующих органов, заказчиков и аудита

- **Paul Lusardi**, *Requirements Management Team Lead*

Использование инжиниринга требований имеет свои неоспоримые преимущества

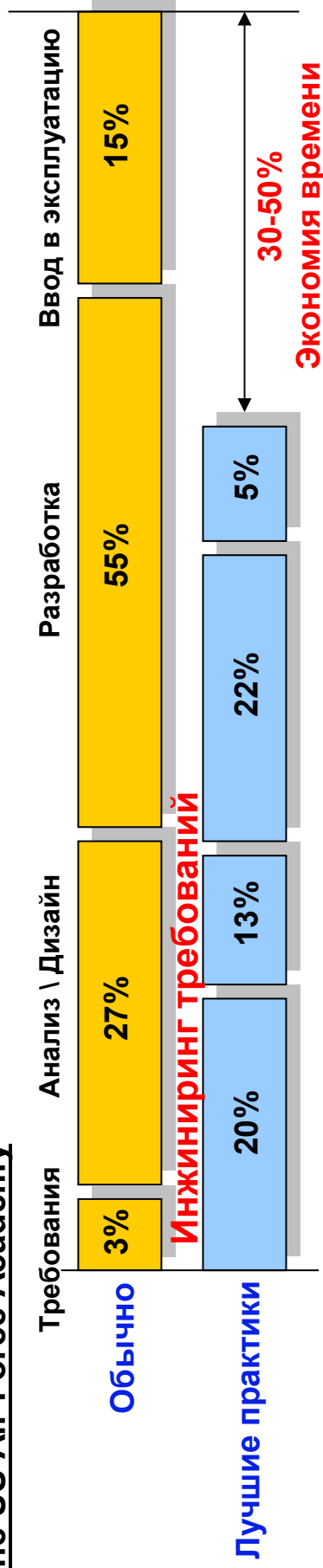
- Почти 100% проектов стали выполняться точно **в срок**
- Ушли проблемы с **перерасходом бюджета**



- Значительно уменьшилось **количество исправлений** (right first time)
- Заметно увеличилась процессная, методологическая, инструментальная и персональная **эффективность** в инжиниринге
- Снизился риск появления **дефектов**
- Инжиниринг требований стал рассматриваться как **конкурентное преимущество**



The US Air Force Academy



Управление требованиями приносит реальную пользу



- Совершенствование цикла разработки современной системы управления ракетным вооружением Tomahawk при использовании **Requirements Management**

Параметры	ДО	ПОСЛЕ
Нестабильность/изменяемость требований: - предварительная проработка дизайна - финальная проработка дизайна	72% 33%	48% 17%
Внесение изменений в требования: - принятые изменения - отклоненные изменения	98% 2%	16% 84%
Время тестирования и приемки: - интеграции - система - приемка заказчиком	9 weeks 13 weeks 22 weeks	4 weeks 6 weeks 10 weeks
Количество дефектов, обнаруженных в готовом изделии	728	165
Время разработки требований к программному обеспечению	10 days	2 days

При отсутствии **Impact Analysis**, большинство изменений попросту принималось

Теперь проведение **Impact Analysis** - вопрос лишь нескольких минут



Требования и качество

- **Качество:**
полное соответствие результата
первоначальным требованиям
- **Цель управления требованиями:**
 - поставка качественного продукта
 - в соответствии с графиком,
 - в рамках выделенного бюджета,
 - отвечающего исходной спецификации,
с полной уверенностью, что все
первичные требования учтены,
проконтролированы и выполнены

IBM Software Group



Преращая создание продукта в конкурентное преимущество:

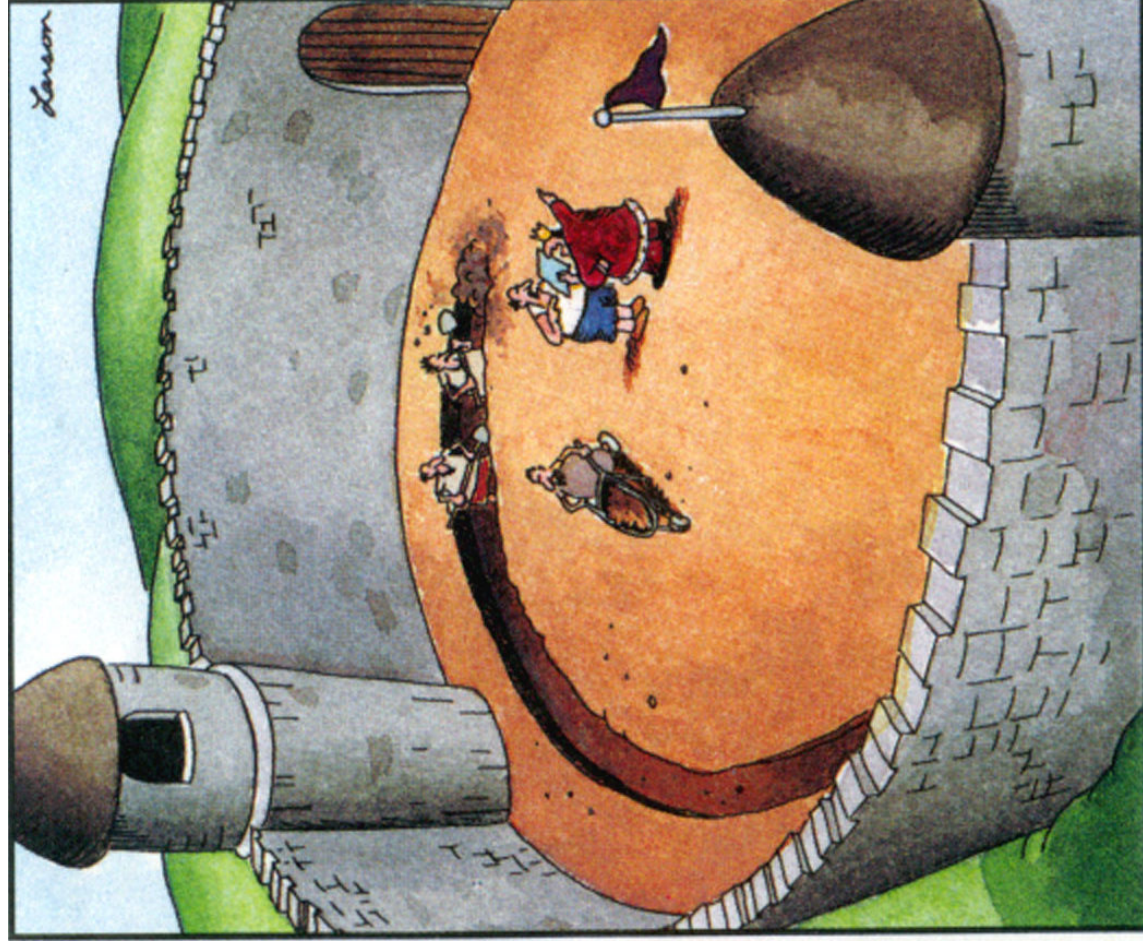
Rational DOORS - инструментальное средство управления требованиями



*Анатолий Волохов,
Software Group, Rational-Telelogic Solutions,
anatoly.volokhov@ru.ibm.com*

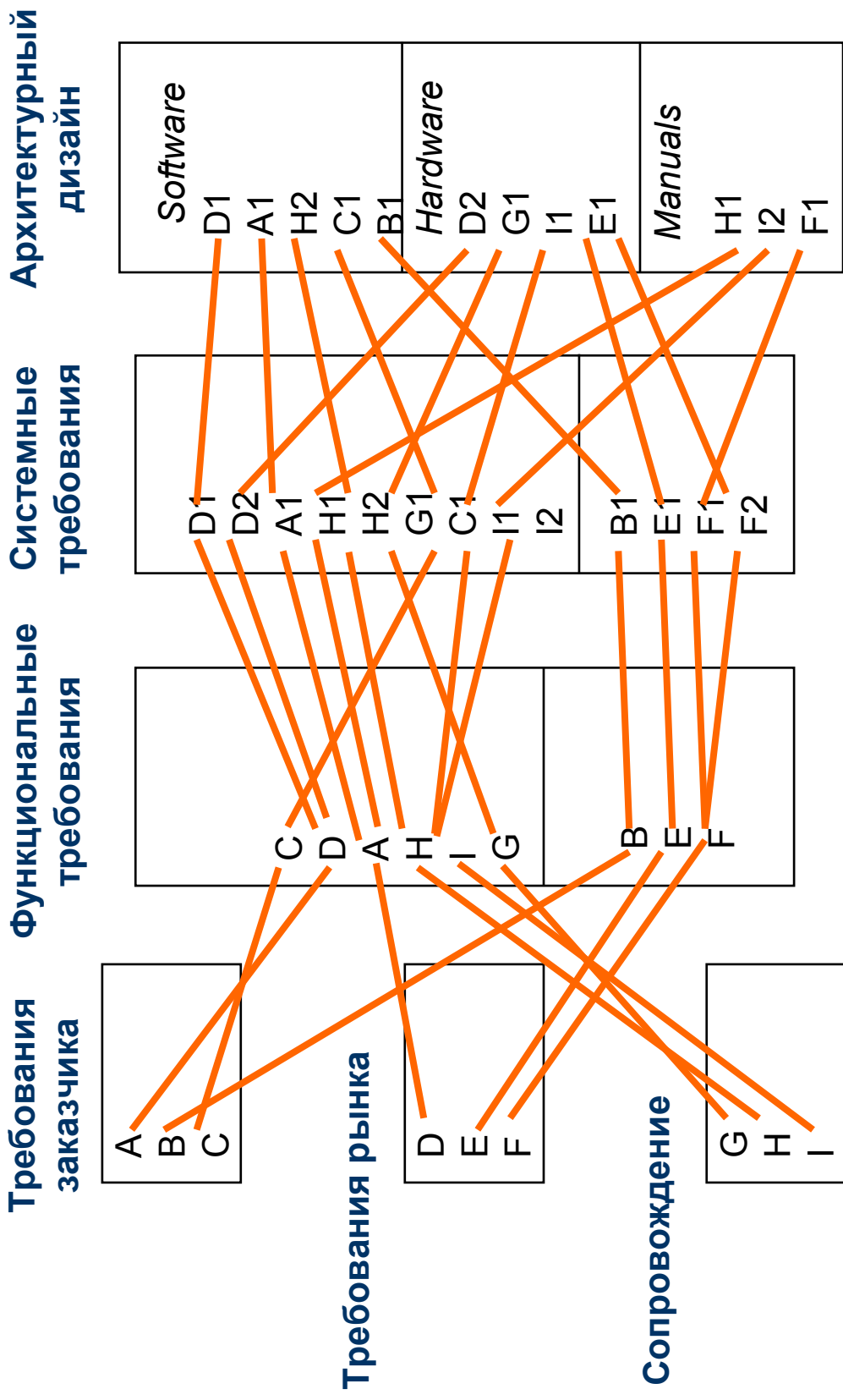


DOORS: В начале было слово...



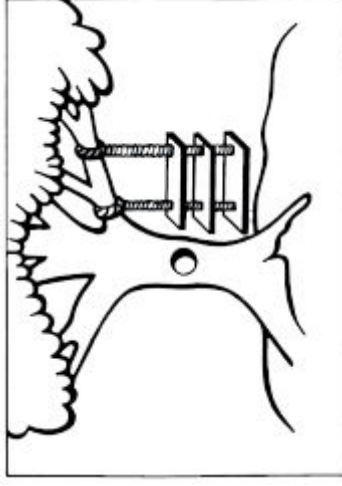


С тех пор все усложнилось...

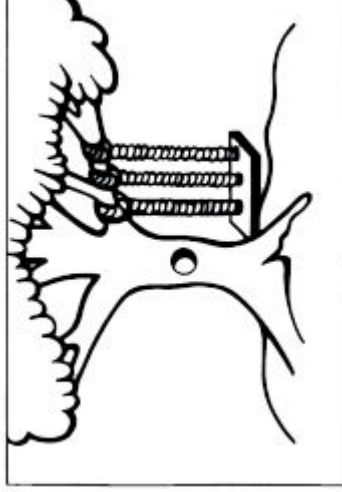


И длинный список заказчиков, подрядчиков, поставщиков ...

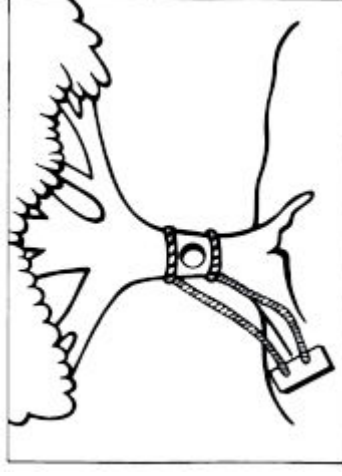
Результат нескоординированной работы



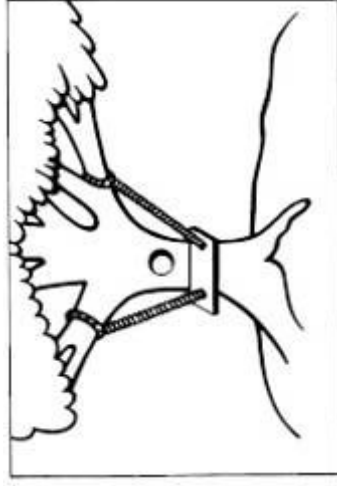
... что хотел
отдел маркетинга



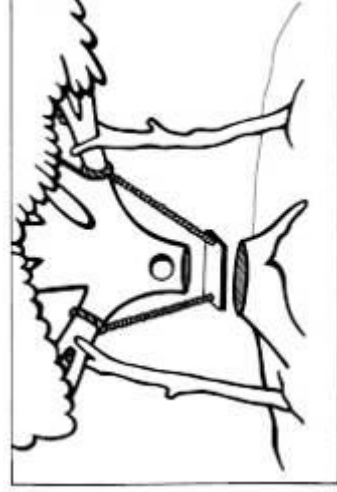
... как это поняли
аналитики



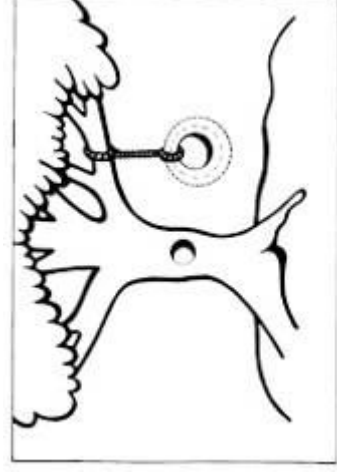
... как это было
сконструировано



... так это было
поставлено



... так это было
собрано



... это реально хотел
иметь заказчик !



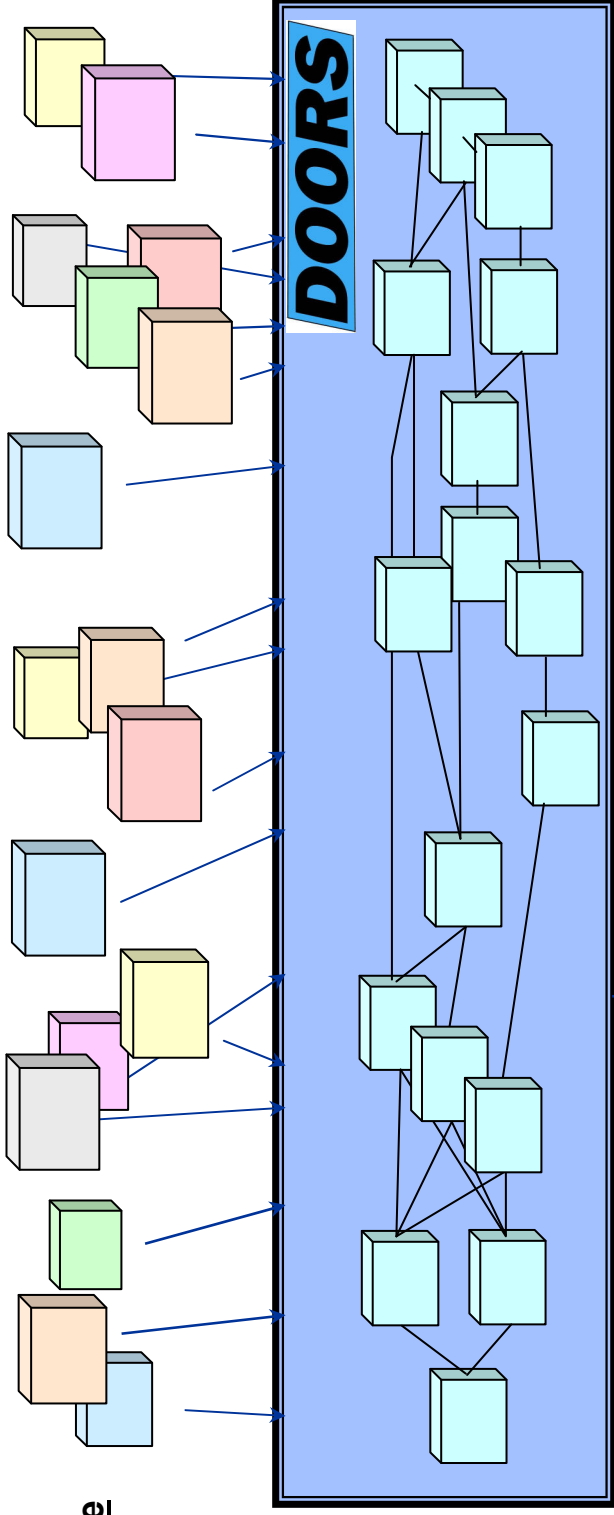
DOORS – ЭТО ВСЕ В ОДНОМ... И СРАЗУ...

Любые
разрозненные

данные

импортируются

Структури-
руются и
линкуются



чтобы
синтезировать
матрицу для
управления
информацией

User requirements for SUV 4x2	Links to Technical Requirements	Design	Links to Tests
3 Requirements This section contains the user requirements.			
3.1 Capability Requirements			
3.1.1 Carrying Capacity			
3.1.1.1 Number of People Four average size adults shall be able to travel in comfort for a period of 3 hours. This level of comfort is defined as being equivalent to the standard of comfort provided by the top 40% of cars produced in 1999.	SR-104 2.14.1.0.1 from /Sports utility vehicle 4x2/Requirements/Functional Requirements The car shall be able to carry 4 average size adults in average comfort for a period of 3 hours. Last modified 11 February 1997	D-342 Full seats shall be created for two passengers in both front and back. D-344 There shall be space for a fifth passenger in the back that will not meet the comfort requirement.	Test Number 18 Market Research Test Result : Passed Test Number 12 Verify Number of People Test Result : Untested
The top level of cars are those in the price range \$20,000 to \$40,000 at 1999 prices.			

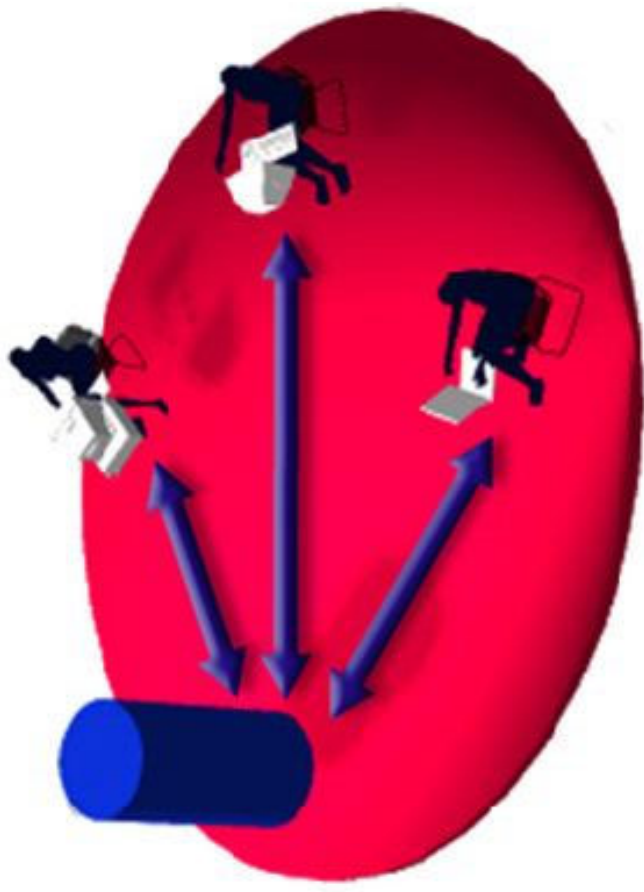


Архитектура DOORS



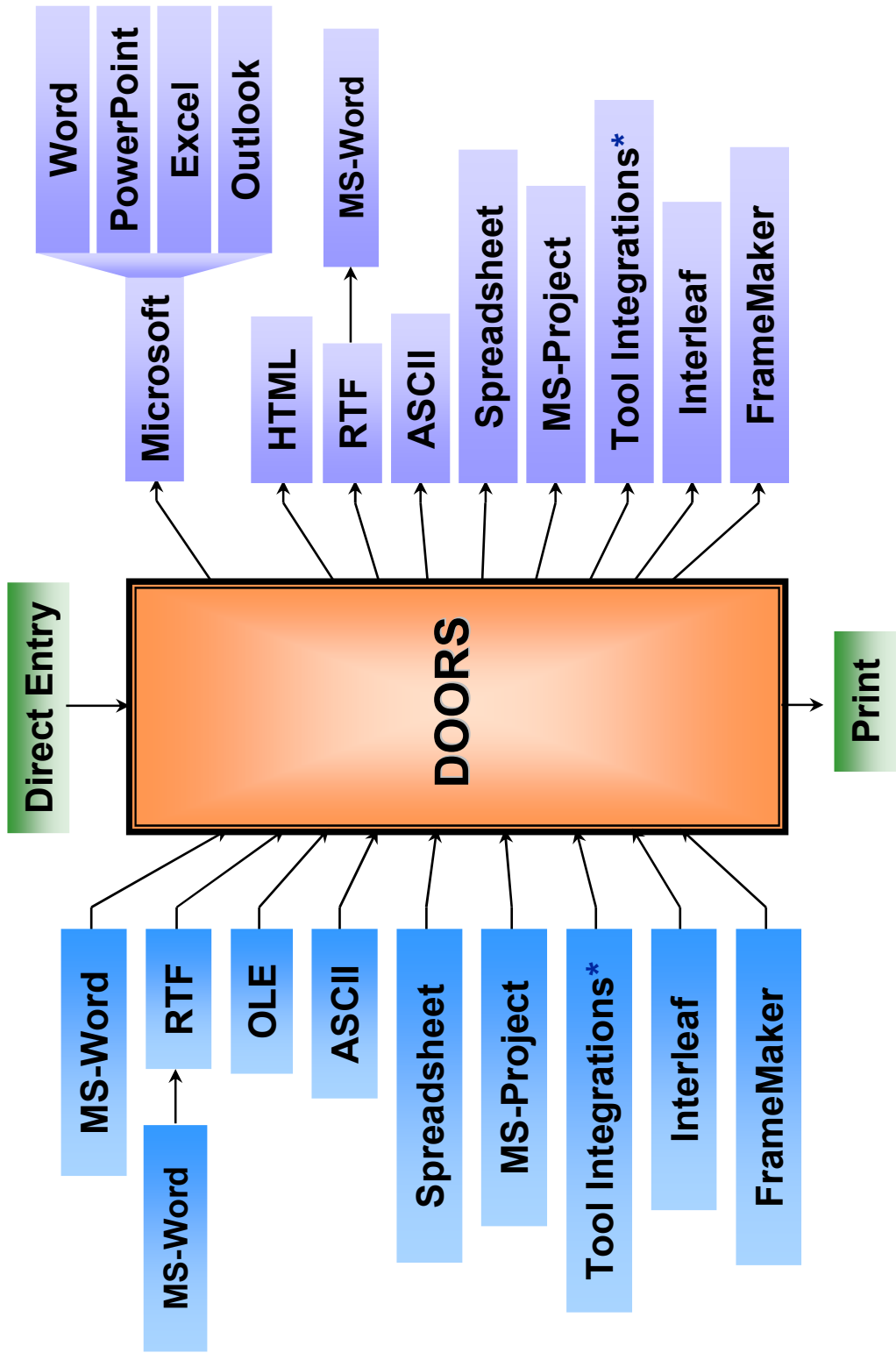
Distributed DWA Users

DOORS

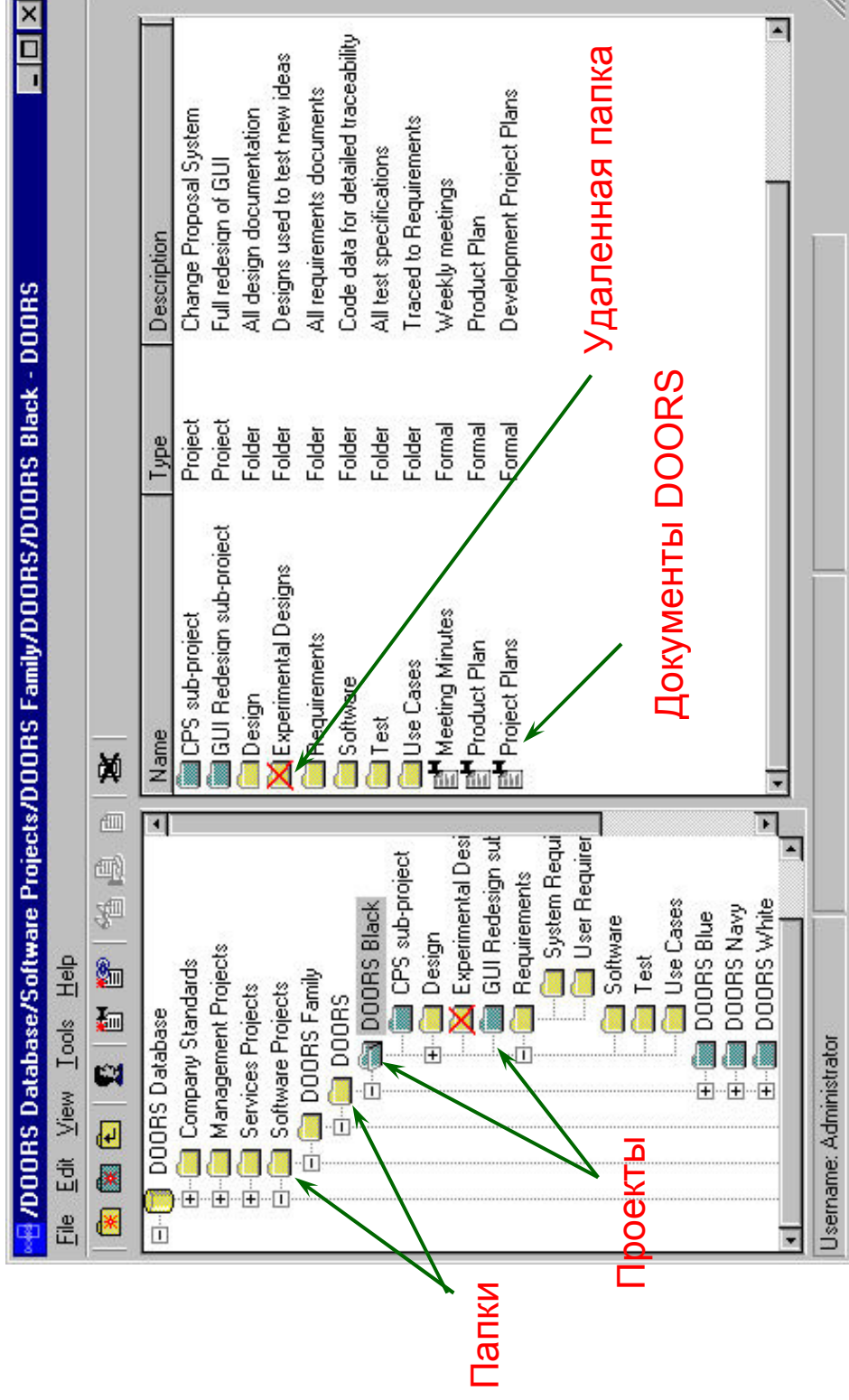




DOORS: ИМПОРТ-ЭКСПОРТ



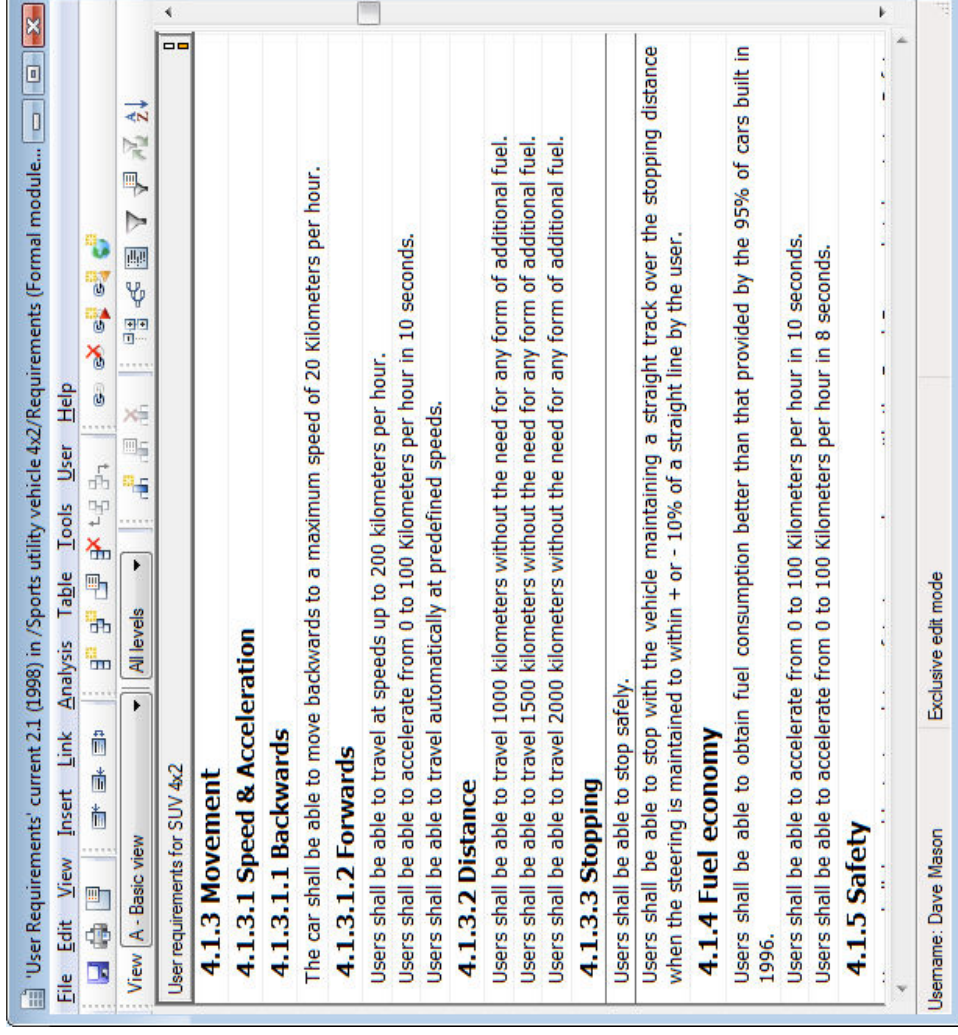
DOORS: база данных



Привычное окружение - легко структурировать проект



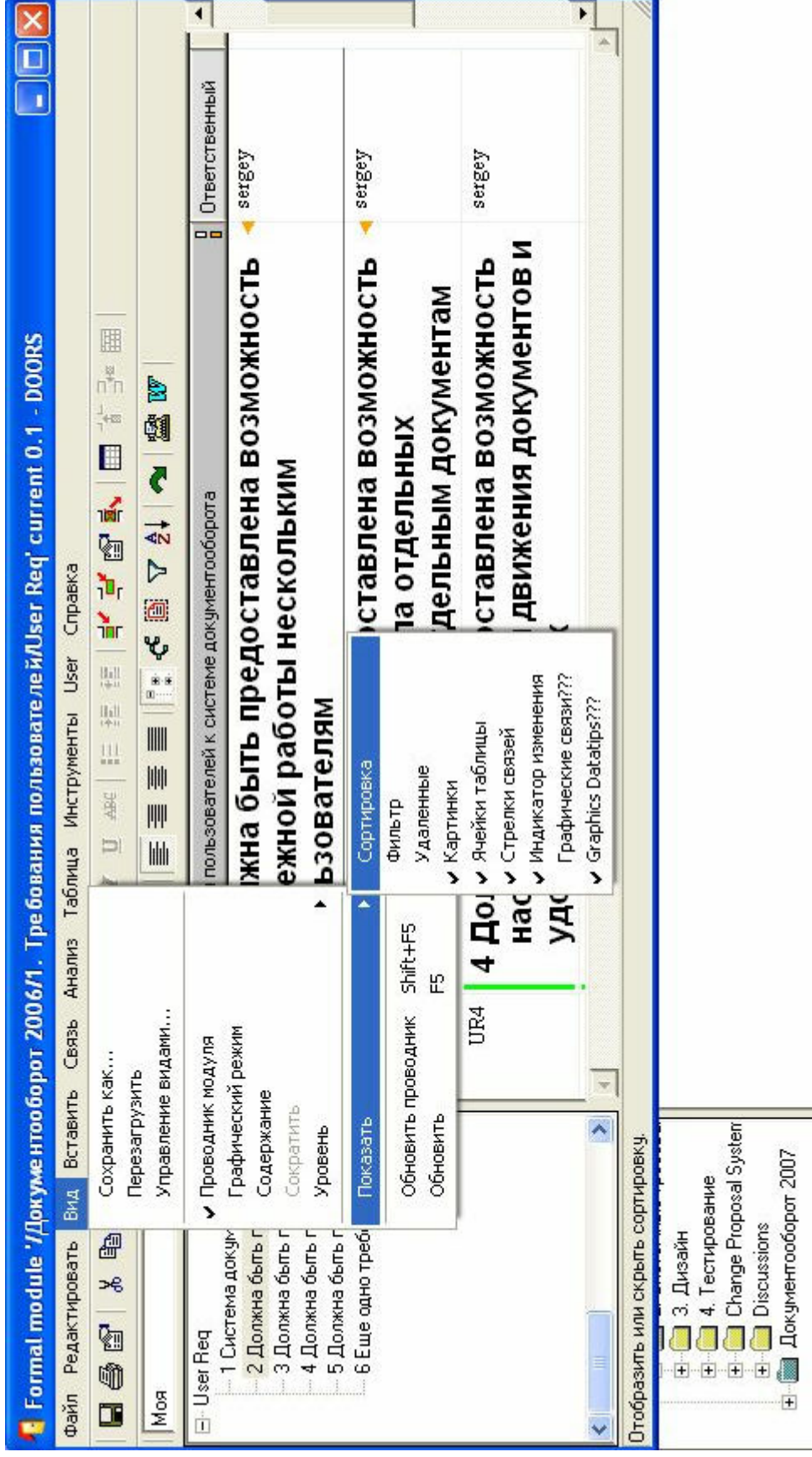
DOORS: ВЗГЛЯД НА ДОКУМЕНТ



Ничего нового – можно сразу начинать

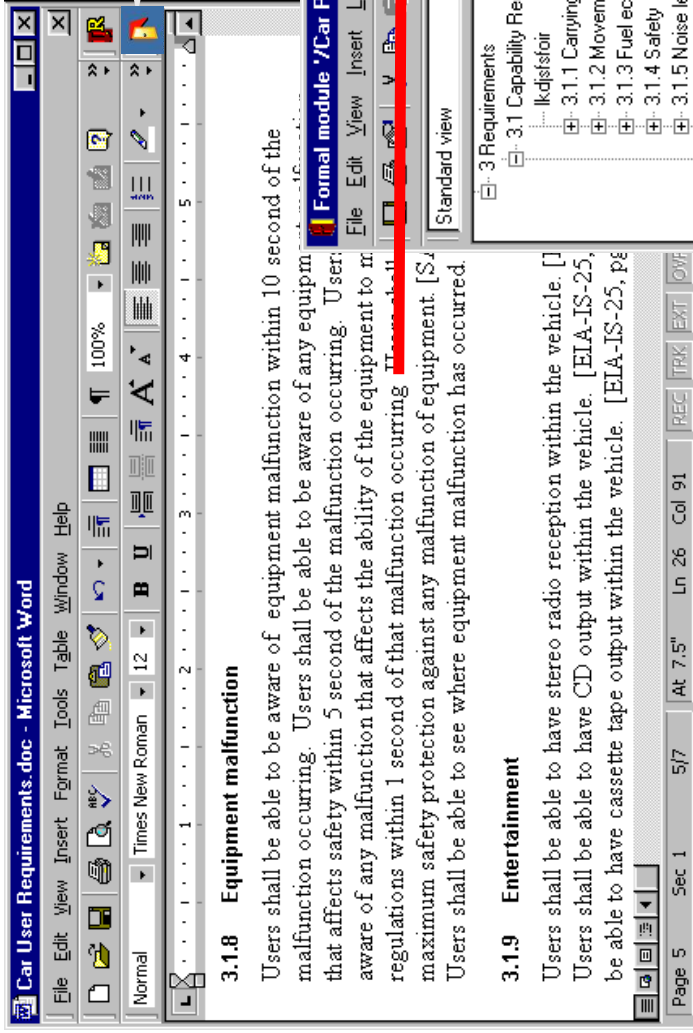


DOORS: поддержка русского языка





Импорт в3 Microsoft Word





ОКНО DOORS

Formal module 'Sports utility vehicle 4x2/Requirements/User Requirements' current 2.1 (1998) - DOORS

Objid	User requirements for SUV 4x2	Test Method
SOW 356	1 Introduction	
SOW 357	These are the initial user requirements for the development of a new sports utility vehicle. All requirements ©2000 QSS, Inc	
SOW 358	1.1 Schedule	
SOW 359	This module contains the user requirements for a new car to be commercially available by 1st September 2002.	Analysis
SOW 3	2 User types	
SOW 4	This section describes the nature of the users of the proposed vehicle.	
SOW 5	2.1 Nationalities	
SOW 6	The car will be used in the countries, UK, North America, Northern Europe, Australia & New Zealand. The European market. file:///C:/Windows/Europe.doc	Analysis
SOW 9	2.2 User sizes	
SOW 10	The car shall be suitable for people minimum and maximum sizes 1.2m to 2m weighing 35 kilograms to 175 Kilograms.	
SOW 11	3 Requirements	
SOW 12	This section contains the user requirements.	
SOW 13	3.1 Capability Requirements	
SOW 14	3.1.1 Carrying Capacity	
SOW 15	3.1.1.1 Number of People	Test
SOW 17	Four average size adults shall be able to travel in comfort for a period of	Analysis

Колонки атрибутов

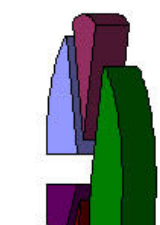
Вся информация в одном окне



DOORS: ВСЕ В ОДНОМ

Formal module: /DOORS Black/Requirements/System Requirements/GUI/ current 0.0 - DOORS

Standard view | All levels | User Interface | Last Modified On

Defined	User Interface	Refined Requirements	Test Case												
	3 Target Users														
×	Users in all industries from commercial to mil-aero.	03 February 2000													
	Users will be of all types from technically proficient to users less computer literate.	20 February 2000	Test 0056-a Ensure MS.Word compatibility. Status: Passed												
	Distribution of user types.	20 February 2000	Test 1034-a Compare to current 2000 industry data. Status: Passed												
			Test 1034-b Verify criteria against previous years. Status: Pending												
	4 Requirements	03 February 2000													
	4.1 Useability	03 February 2000													
×	The user shall be able to enter and edit requirements easily.	20 February 2000	At minimum allow direct typing.												
×	The user shall be able to navigate projects easily.	20 February 2000													
	<table border="1"> <thead> <tr> <th>High</th> <th>Med</th> <th>Low</th> <th>Nil</th> </tr> </thead> <tbody> <tr> <td>12</td> <td>45</td> <td>15</td> <td>3</td> </tr> <tr> <td>127</td> <td>14</td> <td>6</td> <td>1</td> </tr> </tbody> </table>	High	Med	Low	Nil	12	45	15	3	127	14	6	1		
High	Med	Low	Nil												
12	45	15	3												
127	14	6	1												

User name: Administrator

Любые данные, в любом формате



DOORS – ИЗМЕНЕНИЯ И СВЯЗИ

Formal module 'Sports utility vehicle 4x2/Requirements/User Requirements' current 2.1 (1998) - DOORS

File Edit View Insert Link Analysis Table Tools User Help

C - Basic view with details All levels

User Requirements

- 1 Introduction
- 2 User types
- 3 Requirements
 - 3.1 Capability Requirement
 - 3.2 Constraint Requirement
 - 3.2.1 Construction
 - 3.2.2 Terrain
 - 3.2.3 Availability
 - 3.2.4 Lifetime
 - 3.2.5 Security
 - 3.2.6 Accessories
 - 3.2.6.1 A warning triangle
 - 3.2.6.2 A first aid kit shall
 - 3.2.7 Company constr
 - 3.2.8 Cross section di
 - 3.2.9 Fuel input
 - 4 Project Structure
 - 5 Costs
 - 5.1 Expected further costs...

User requirements for SUV 4x2

1 Introduction

These are the initial user requirements
All requirements ©2000 QSS, Inc

1.1 Schedule

This module contains the user req
September 2002.

2 User types

This section describes the nature of

2.1 Nationalities

The car will be used in the countr
Zealand. The European market: file

2.2 User sizes

The car shall be suitable for peop
kilograms to 185 Kilograms.

3 Requirements

This section contains the user requi

3.1 Capability Requirement

3.1.1 Carrying Capacity

3.1.1.1 Number of People

Four average size adults shall be al
comfort is defined as being equivalent
produced in 1999.

User	Session	Date	Modification
Paul Raymond	1	06/02/03 14:15:58	Create Object
Paul Raymond	1	06/02/03 14:16:39	Modify Object Attribute: Object Text
Paul Raymond	1	06/02/03 14:17:17	Modify Object Attribute: Object Text

Details of selected history record

From: The tests of this vehicle should be completed with ~~complete absence of~~permissible failure. ~~during the~~ (defined as acceptable wear and tear) during the first 3000 mile road test.

To: The tests of this vehicle should be completed with permissible failure (defined as acceptable wear and tear) during the first 3000 mile road test.

Restore View change as redlining

Only show entries with

Dates: from: for: Details...

User: Refresh

Export...

Previous Next OK Cancel Apply Help

Username: Paul Raymond Exclusive edit mode

Индикатор
изменений

Идеально для быстрого отслеживания изменений



Создание связей – drag-and-drop

The screenshot shows two windows of the Rational software interface. The top window, titled 'Functional System requirements for SUV 4x2', contains a list of requirements (FR-19 to FR-37) and a table of contents with sections like '2.2.3 Brake car', '2.2.4 Control direction', and '2.3.1 Illuminate external'. The bottom window, titled 'User Requirements' current 2.1 (1998) in /Sports utility vehicle 4x2/Requirements (Formal module)...', contains a list of requirements (4.1.3 Movement, 4.1.3.1 Speed & Acceleration, 4.1.3.1.1 Backwards, 4.1.3.1.2 Forwards, 4.1.3.2 Distance, 4.1.3.3 Stopping, 4.1.4 Fuel economy, 4.1.5 Safety) and a table of contents. Two red arrows indicate drag-and-drop connections: one from '2.2.4.1.1 Direction mechanism' in the top window to '4.1.3.1.2 Forwards' in the bottom window, and another from '2.3.1.1.1 Headlights' in the top window to '4.1.3.3 Stopping' in the bottom window.

как внутри одного документа...

... так и между разными документами



От А до Я - полная картина

The screenshot displays a multi-pane view of a software development project. The top pane shows a state machine diagram for the CD player module. The middle pane lists requirements, and the bottom pane shows test cases and their execution results.

Requirements:

- 3.1.9 Enter: In accordance with Entertainment's of our Cars for The passenger. The passenger. The passenger. Users shall be able to select a CD. Users shall be able to select a CD. Users shall be able to select a CD.
- 3.1.10 Mail: Users shall not be able to select a CD. Users shall not be able to select a CD. Users shall not be able to select a CD.
- 3.1.11 Ser: Users shall be able to select a CD. Users shall be able to select a CD. Users shall be able to select a CD.

Test Cases:

- 1.1 Control radio: Demonstrate that the user can control radio reception.
- 1.2 Control CD: Demonstrate that the user can control track selection.
- 1.3 Control tape player: Demonstrate that the user can start, stop, eject, fast forward, and reverse.
- 1.4 Control Display: Show via inspection that the entertainment system displays random colors to prevent screen burn in. Measure that the screen refresh rate is 20 times per second with a variance +/- 0.25 seconds.

Test Results:

ID	Test Case	Defect	Execution Status
2	1 Screensaver refresh rate	2	Failed
3	2 Screensaver random shapes and colors test		Passed
4	3 Validate display presence		Passed
5	4 Demonstrate CD player		Passed



Трассировка дает возможности анализа

Требования заказчика

Технические требования

Дизайн

Тестирование

User requirements for SUV 4x2	Design	Links to Tests
<p>3 Requirements</p> <p>This section contains the user requirements</p> <p>3.1 Capability Requirements</p> <p>3.1.1 Carrying Capacity</p> <p>3.1.1.1 Number of People</p> <p>Four average size adults shall be able to travel in comfort for a period of 3 hours. This level of comfort is defined as being equivalent to the standard of comfort provided by the top 40% of cars produced in 1999.</p> <p>The top level of cars are those in the price range \$20,000 to \$40,000 at 1999 prices.</p> <p>Five average size adults shall be able to travel in comfort for a period of 3 hours.</p> <p>Users shall have easy entry and exit.</p>	<p>D-342 Full seats shall be created for two passengers in both front and back.</p> <p>D-344 There shall be space for a fifth passenger in the back that will not meet the comfort requirement.</p>	<p>Test Number 18 Market Research Test Result : Passed</p> <p>Test Number 12 Verify Number of People Test Result : Untested</p> <p>Test Number 6 Verify support for Customers Test Result : Untested</p>

Реализацию этого требования как раз упустили из виду



Виды... Атрибуты...

Formal module 'Sports utility vehicle 4x2/Requirements/User Requirements' current 2.1 (1998) - DOORS

File Edit View Insert Link Analysis Table Tools User DOORSrequireIT Help

F - Budget [All levels] [Spent] [Remaining] [Verification Method] [Risk] [Last Modified On]

Obj id	Spent	Remaining	Verification Method	Risk	Last Modified On
User requirements for SUV 4x2					
SOW 37	0	146	No Verification Needed		11 April 1997
SOW 38	0	67		High	27 March 1997
SOW 39	0	79		Medium	03 December 1997
SOW 364	0	79		High	03 December 1997
SOW 40	0	20			11 April 1997
SOW 41	0	0	Demonstration	Medium	03 December 1997
SOW 42	0	20	Demonstration	Medium	26 December 1997
SOW 43	0	95			11 February 1997
SOW 44	0	81			11 April 1997
SOW 45	0	81	Analysis	Low	27 March 1997
SOW 46	0	14		High	
SOW 47	0	14		Medium	1997
SOW 48	0	475		Low	11 April 1997
SOW 49	0	475			11 February 1997

Username: Paul Raymond | Exclusive edit mode

Практически неограниченное число атрибутов (колонок)

Project Plan [All levels] [Icons for undo, redo, copy, paste, etc.]

ID	REQ	SRS - Requirement Specification	Status	Priority	Release	Release Date	Team
SRS1 1	X	Dataflow Diagrams (DFD) considers both data and processing. It's modeling can be described in a number of ways:	In Design	Low	Version 1	24/06/02	Tea
2.1.1 links to links							
SRS2 43	✓		In Design	Med	Basis		Tea
SRS1 2	X	What functions must the system perform? What are the interactions between those functions?	Done	High	Version 2	14/06/02	Tea
SRS1 3	X	What transformations must the system carry out? What inputs are transformed into what outputs?	In Design	High	Basis	24/06/02	Tea
SRS1 4	X	What kind of work does the system do? Where does it get the information to do its work? Where does it deliver the results of its work?	In Design	Low	Basis	24/06/02	Tea
SRS1 5	X	Dataflow diagrams consist of processes, data stores, flows, and terminators:	In Design	Med	Version 1.1	24/06/01	Tea
SRS1 6	X	Processes	In Design	Low	Basis	24/06/02	Tea
SRS1 7	X	Flows	In Design	Low	Basis	24/06/02	Tea
SRS1 8	X	Data stores	In Design	Med	Version 2	14/06/02	Tea
SRS1 9	X	Terminators	In Design	High	Basis	24/06/02	Tea
SRS2 0	X	Process are shown by the circles, or "bubbles" in the diagram. They represent the various individual functions that the system look out. Functions transform inputs into outputs.	In Design	Low	Basis	24/06/02	Tea
SRS2 1	✓	Flows are shown by curved, directed arrows. They are the connections between the processes (system functions), and they represent the information that the processes require as input and/or the information generate as output.	In Design	Med	Basis	24/06/02	Tea
SRS2 2	X	Data stores are shown by two parallel lines, or by an ellipse. They show collections (aggregates) of data that the system must remember for a period of time. When the systems designers and programmers finish building the system, the stores will typically exist as files or databases.	In Design	Low	Version 1.1	24/06/02	Tea

Formal module - Demo Project / Requirements' current 0.2 (beta) - DOORS

File Edit View Insert Link Analysis Table Tools User Forum Elisra Hebrew Kitchen Manageware Reports Spelling Word Help

Project Plan [All levels] [Icons: Undo, Copy, Paste, Print, Find, etc.]

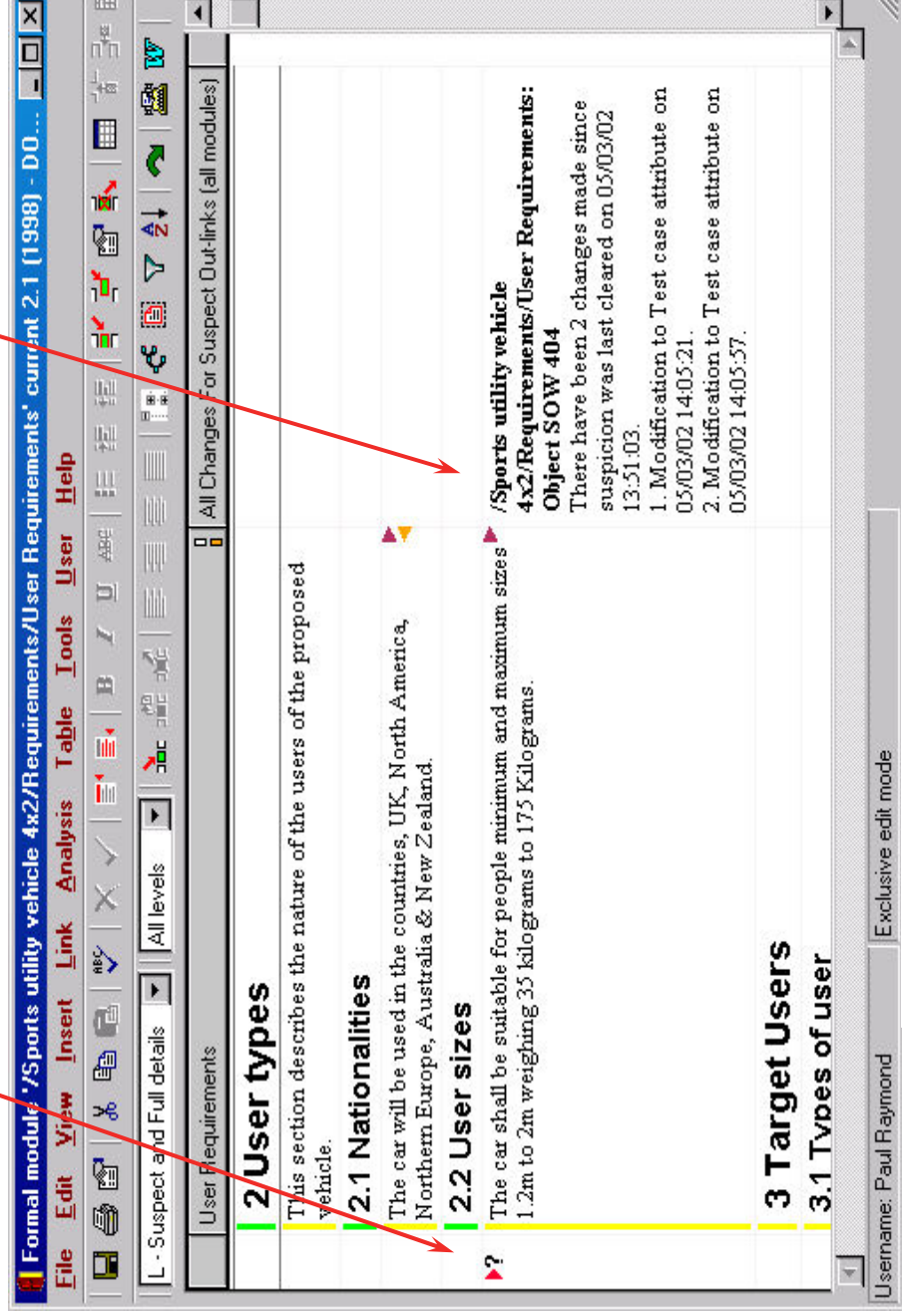
Status	Priority	Release	Release Date	Team	Responsible	Initiator	Cost	Budget	Cost Budget	Approved
In Design	Low	Version 1	24/06/02	Team A	David	QA			C B	True
In Design	Med	Basis		Team A	David	QA	4500	8300	C B	True
Done	High	Version 2	14/06/02	Team C	David	R&D	5676	9900	C B	False
In Design	High	Basis	24/06/02	Team C	David	QA	4555	9000	C B	True
In Design	Low	Basis	24/06/02	Team A	David	QA	2344	1000	C B	False
In Design	Med	Version 1.1	24/06/01	Team C	David	QA			C B	True
In Design	Low	Basis	24/06/02	Team C	David	QA			C B	True
In Design	Low	Basis	24/06/02	Team C	David	QA			C B	True
In Design	Med	Version 2	14/06/02	Team C	David	QA			C B	True
In Design	High	Basis	24/06/02	Team C	David	QA			C B	True
In Design	Low	Basis	24/06/02	Team C	David	QA			C B	True
In Design	Med	Basis	24/06/02	Team C	David	QA			C B	True
In Design	Low	Version 1.1	24/06/02	Team C	David	QA			C B	True

Подозрительные связи

Подозрительные связи (Suspect links) представлены в документе:

либо как индикаторы

либо как полное описание

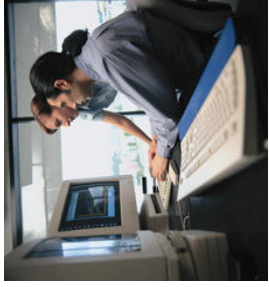


Цель последовательности никогда не прервется



Трассировка поддерживает целостность набора документов

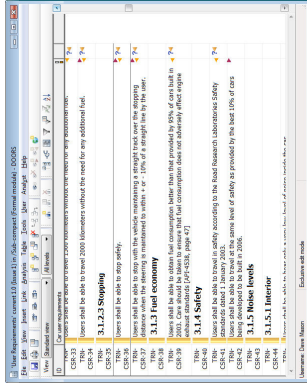
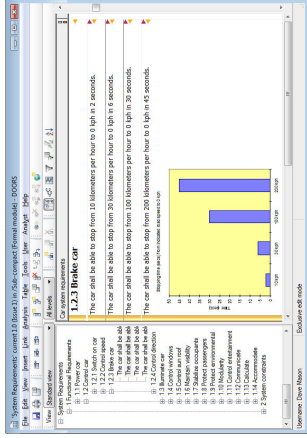
Если документы связаны ...



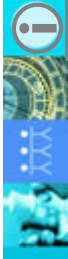
то изменение,
сделанное в одном
документе...



... отражается в
виде флага в
другом документе

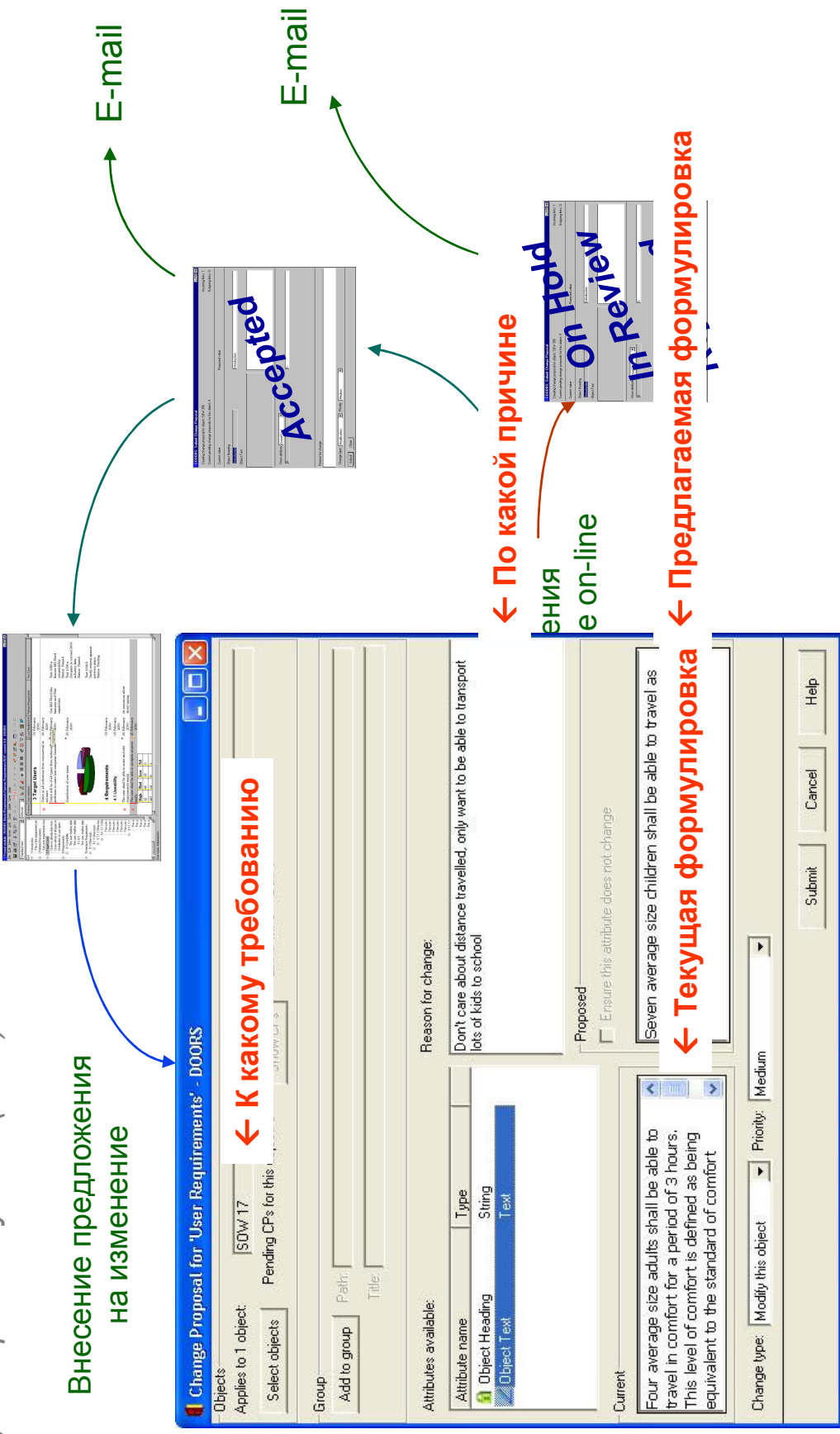


Может быть организована нотификация исполнителей по e-mail





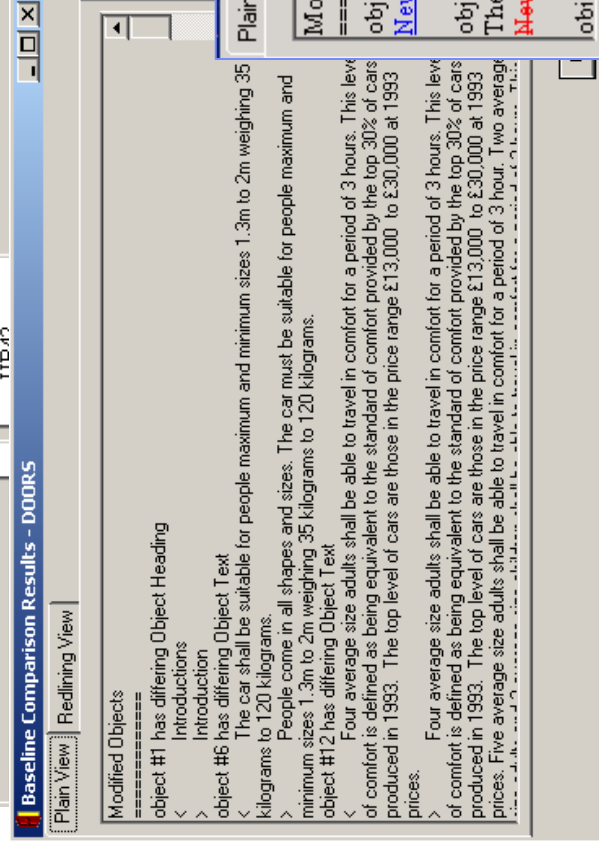
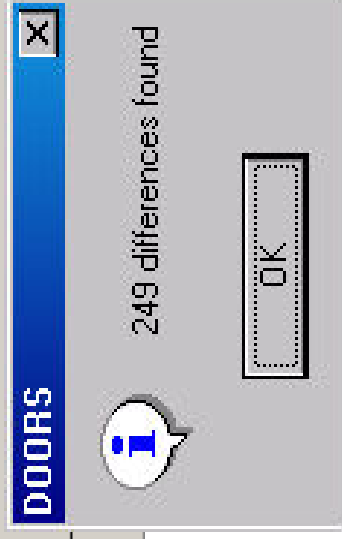
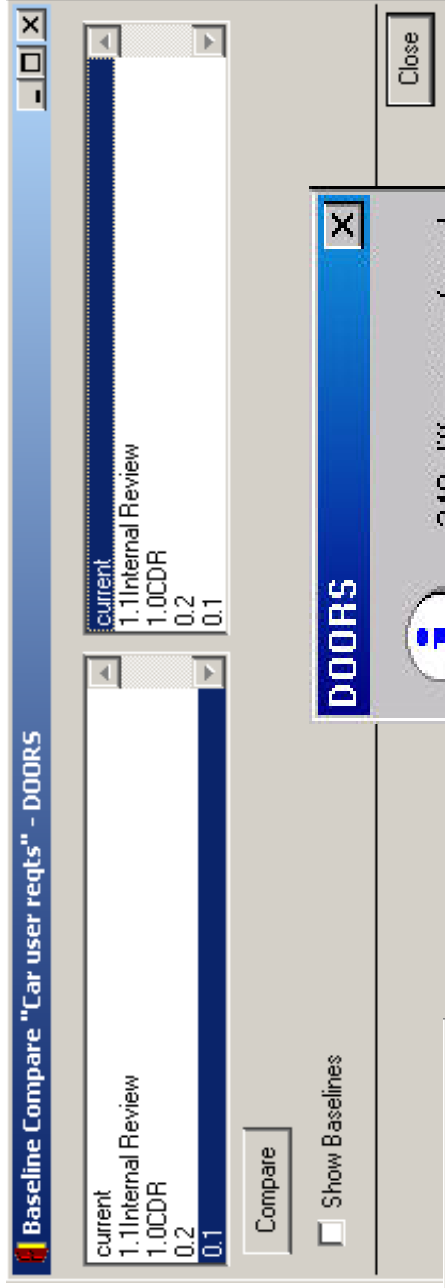
СИСТЕМА ВНЕСЕНИЯ ИЗМЕНЕНИЙ: Change Proposal System (CPS)



Пусть DOORS вместо вас контролирует процесс внесения изменений



Поддержка Baselines модулей, проектов





DWA как альтернатива DOORS Desktop

Telelogic DOORS Web Access - Windows Internet Explorer

http://1212.209.58.153:8080/dwa/dwa.html#

File Edit View Favorites Tools Help

Telelogic DOORS Web Access

User: Administrator, Package: Review Logout

Goto URL Layout Package Help

Stakeholder Requirements View: 04 - Linked System

+/-	Object Identifier	Car user requirements	System Requirements
	TRN-CSR-34	need for any additional fuel.	The user shall be able to accurately calculate fuel required for journey.
	TRN-CSR-35	3.1.2.3 Stopping	TRN-MR-28 Sec. #: 3.1.3.1.0-2 The driver shall be able stop car safely.
	TRN-CSR-36	Users shall be able to stop safely.	TRN-MR-28 Sec. #: 3.1.3.1.0-2 The driver shall be able stop car safely.
	TRN-CSR-37	Users shall be able to stop with the vehicle maintaining a straight track over the stopping distance when the steering is maintained to within + or - 10% of a straight line by the user.	TRN-MR-28 Sec. #: 3.1.3.1.0-2 The driver shall be able stop car safely.

System Requirements View: Standard view

+/-	Id	Control direction	Test Plan
	TRN-SR-24	1.2.4 Control direction	4.2 Planned tests
	TRN-SR-25	1.2.4.1 Straight line	4.2.1 (Item(s) to be tested)
	TRN-SR-26	The car shall have a mechanism to enable it to be moved or backwards.	4.2.1.1 (Project-unique identifier o
	TRN-SR-	1.2.4.1.1 Direction mechanism	5 Test schedules

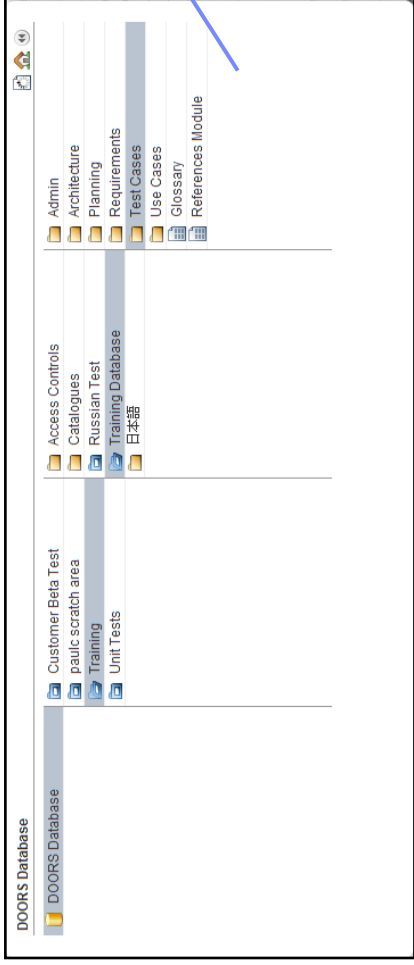
Attributes Discussions

Attribute	Value
User	
Object Heading	
Object Text	The car shall be able to stop from 100 kilometers per hour to 0 kph in 30 seconds.
Object Short Text	
Cost	10.000000
Industrial area	
Object Type	Requirement
Priority	Not Set
Progress	
Status	Draft
System_Old_InLinks	50285051
System_Old_OutLinks	3638
System_OldID	1022
System	
Absolute Number	22
Object Number	1.2.3.0-3
Last Modified By	Bill Young
Last Modified On	01 November 2007
Created By	Bill Young
Created On	06 September 2007

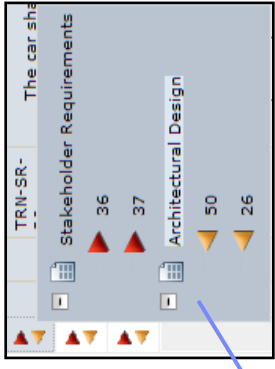
Copyright © 2008 Telelogic AB. Version 1.0.0.0 (Build 184)



DOORS Web Access – Все для работы



**Полный
доступ к базе**



Удобные ссылки

Attributes		Discussions	
User			
Object Heading	The speed control shall be infinitely variable from zero to maximum speed.		
Object Text			
Object Short Text			
Cost	10.000000		
Industrial area	Requirement		
Object Type	Not Set		
Priority			
Progress			
Status	Draft		
System_Old_InLinks			
System_Old_OutLinks			
System_OldID	1017		
System			
Absolute Number	17		

Полная панель атрибутов



Поиск

Attributes		Discussions		New	
Sort By					
	Paul Clunie	19-Jun-2008	01:01:AM		
	[3] test for Liz				
	Mark Higgins	07-Jul-2008			
	[2] this is a test discussion summary				
	Administrator	03:20:PM			
	Agreed this discussion is relevant for the way that DWA will display the data to the user. If you have any questions then please contact the DWA administrator.				
	Mark Higgins	10-Jun-2008	11:51:AM		
	and this is a test comment				
	Morgan Brown	10-Jun-2008	11:50:AM		
	[3] Demonstration for Customer				

**Доступ к
ДИСКУССИЯМ**



Расширение возможностей Doors

DXL : Doors eXtension Language

DOORS: Tender Assessment by Richard M. Marshall

Requirement: TBS View Feature: Apple9 View

Operating temperatures: between 5C and 32C. Will operate in temperatures ranging from 6C to 32C.

Value: 6 Bidder: Apple Cost:

Importance: Mandatory

Requirement is: partially met by B2 %

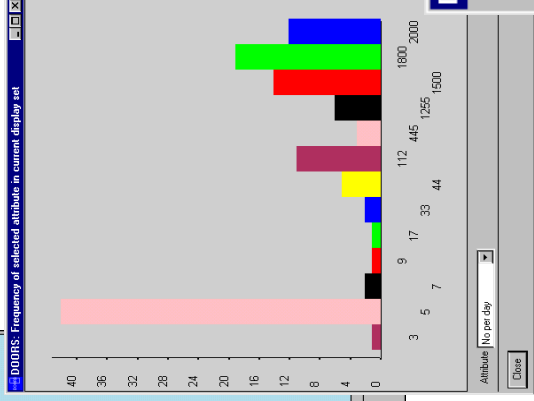
Comments: Unlikely to cause problems - willik stand up to being kept overnight in a car in winter?

Accept Next Bidders Save Close

DOORS: Tender Evaluation Results

	Apple	Protel	Tiger	Transbas
Offer quality	52%	22%	100%	90%
Understanding	85%	80%	95%	85%
Mandatory not met	4	8	1	None
Offer value	49%	40%	84%	85%
Offer cost	£2470	£1925	£10355	£1975
Overall	53%	35%	82%	90%
Acceptance	Re-assess	Reject	Overwrite	Accept

Notebook computer invitation to tender
Target price £2500
Assessors: Chris Anders Richard M. Marshall



Анализ

Специальные
окна
пользователя

Статистика

DOORS

No test specified for this requirement

OK

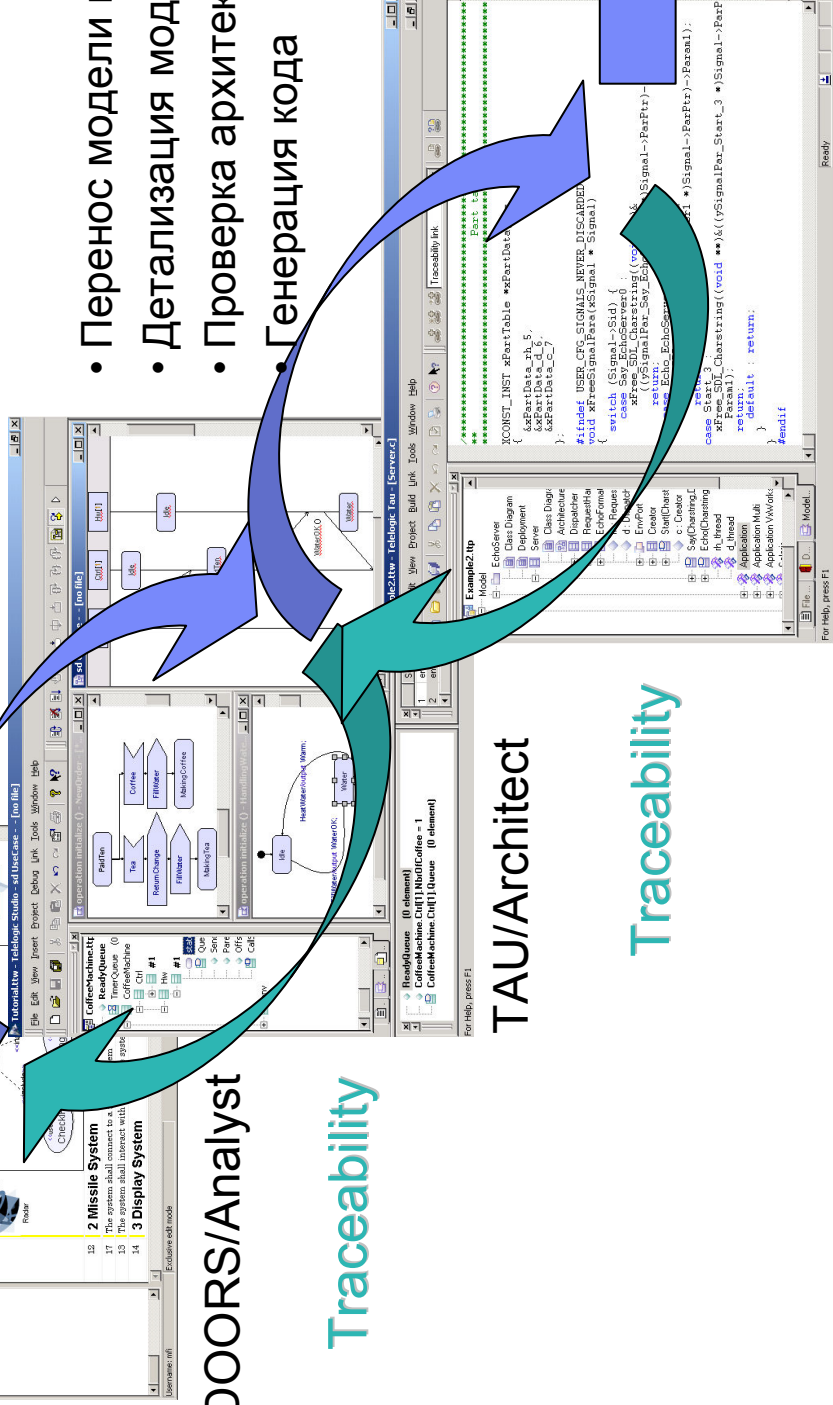
Подсказка





Экспорт модели в Rhapsody, Tau, Rose

- Перенос модели в TAU/Architect
- Детализация модели
- Проверка архитектуры и функционала



- Перенос модели в TAU/Developer
- Детализация модели
- Проверка архитектуры и функционала
- Генерация кода

TAU/Architect
Traceability



Application

TAU/Developer



Разработка на основе требований

Предпосылки перехода к Model Driven Architecture

DOORS:
Управление требованиями и трассировка

The screenshot displays the IBM Rational DOORS tool interface. The top pane shows a list of requirements, with 'SR-145' circled in red. The bottom pane shows a UML class diagram for the 'GPSControl' package, with a red arrow pointing from the circled requirement ID to the 'GPSControl' class. To the right, another UML class diagram for the 'CarInterfaces' package is visible, with a blue arrow pointing from the 'GPSControl' class diagram to it.

Rhapsody & Tau:
Визуализация требований и модели

«Привяжите» элементы модели к вашим требованиям



Сравнение результатов тестирования

Formal module /Sports utility vehicle 4x2/Test and verification/Test Cases/ current 0.0 - DOORS

File Edit View Insert Link Analysis Table Tools User Test Tracking Help

Test Run Compare Must Test After All levels

Test Number	Must Test After	Expected Test Result	Actual Test Result 1/2	Test Engineer 1/2	Test Status 1/2
320	356	Check display for location indicator.	W6 was displayed. But W5 displayed also. ==== W6 was displayed only	PR ==== EM	Fail ==== Pass
357	356	Time the response of malfunction notification by using system code 185 to induce a serious failure.	Display time was 4 seconds. ==== Display time was 4 seconds.	PR ==== EM	Pass ==== Pass
389		Ensure malfunction notification by using system code 202 to induce a failure.	Malfunction was displayed. ==== Malfunction was displayed.	PR ==== EM	Pass ==== Pass
322	389	Check display for location indicator.	W4 was displayed. ==== W4 was displayed.	PR ==== EM	Pass ==== Pass
390	389	Users shall be able to be aware of any malfunction that affects the ability of the equipment to meet the statutory regulations within 1 second of that malfunction occurring	Display time was 4 seconds. ==== Display time was 3 seconds.	PR ==== EM	Fail ==== Fail

Username: Paul Raymond | Exclusive edit mode

Тестирование требований (статус реализации)

The screenshot shows the Rational software interface for testing requirements. The main window displays a list of requirements and their associated test results. A red circle highlights the test results for requirement 4.1.8, which are: Test Number: 390, Test Status 2: Fail, Test Number: 389, and Test Status 2: Pass.

Requirement	Test Results
4.1.8 Equipment malfunction Users shall be able to be aware of equipment malfunction within 10 second of the malfunction occurring.	Test Number: 350 Test Status 2: Pass Test Number: 349 Test Status 2: Pass
Users shall be able to be aware of any equipment malfunction that affects safety within 5 second of the malfunction occurring.	Test Number: 356 Test Status 2: Pass Test Number: 357 Test Status 2: Pass
Users shall be able to be aware of any malfunction that affects the ability of the equipment to meet the statutory regulations within 1 second of that malfunction occurring	Test Number: 390 Test Status 2: Fail Test Number: 389 Test Status 2: Pass
Users shall be able to have maximum safety protection against any malfunction of equipment.	Test Number: 318 Test Status 2: Pass Test Number: 320 Test Status 2: Pass Test Number: 322 Test Status 2: Pass

Фильтр:
результаты
всех тестов
по каждому
требованию



Сравнение результатов тестовых запусков

Compare Test Runs - DOORS

This tool will compare a single enumerated attribute between any two test runs. It displays the numbers of test cases that have changed in a particular way. Selected transitions are totaled, and you can filter the test case on the selection.

Data source
Enumerated attribute to compare: Test Status
Test runs to compare: Old [1] New [2] Collect Data

Filter details
 Convert view to show comparison View name to convert: test.2 Filter Tests
Number of test cases selected for filtering: 3 (75% of 4)

List view Graphic view
Graph of transitions (Select by clicking on circles)

Test Status (Test run 1)	Test Status (Test run 2)
Exempt	Exempt
Undetermined	Undetermined
Fail	Fail
Pass	Pass

The graphic view shows a 4x4 grid of transitions. The top row is labeled 'Test Status (Test run 1)' and the right column is labeled 'Test Status (Test run 2)'. The grid cells are colored based on the status: blue for Exempt, yellow for Undetermined, red for Fail, and green for Pass. Two circles are overlaid on the grid, labeled '1' and '2', indicating selected test cases.

Close Help

- **Наглядность**

- **ПОНЯТНОСТЬ**

- **Доступность**

Основные преимущества DOORS:

- Полная информация по проектам – в любое время, в любом месте
- В работе всегда самая последняя редакция любого документа
- Возможность контролировать реализацию каждого отдельного требования и всего проекта в целом
- Эффективная работа в коллективе (*в т.ч. и с заказчиками*):
 - ▶ Работа с единой базой данных
 - ▶ Контроль доступа к информации
 - ▶ Контроль за исполнением на любом этапе (особенно на самых ранних)
- Простота внедрения:
 - ▶ с текстом умеет работать все
 - ▶ остальному – научатся
- Значительное повышение качества разработок. Проект реализуется:
 - ▶ В нужные сроки
 - ▶ В рамках бюджета
 - ▶ С уверенностью, что каждый пункт задания выполнен
- Значительная экономия времени, средств и ресурсов

Дополнительная информация

- Презентации по DOORS и DWA:
 - ▶ http://public.dhe.ibm.com/software/dw/demos/rdoors/doors_final1.html?S_CMP=rnav
 - ▶ http://download.boulder.ibm.com/ibmdl/pub/software/dw/demos/rtelelogicdoorsweb/1438_Telelogic_DOORS_WEBACCESS.htm
- Демонстрационные ролики по DOORS и DWA:
 - ▶ <http://www-01.ibm.com/software/awdtools/doors/>
 - ▶ http://www.ibm.com/developerworks/offers/p/demos/summary/r-telelogicdoors.html?S_TACT=105AGX28&S_CMP=TWDDW
 - ▶ <http://www-01.ibm.com/software/awdtools/doors/webaccess/>
- Возможность в режиме on-line тестировать DWA:
 - ▶ <http://www.ibm.com/developerworks/downloads/r/doorswebaccess/index.html>
- Дополнительная информация о DOORS и DWA (англ.) может быть найдена здесь:
 - ▶ <http://www-01.ibm.com/software/awdtools/doors/>
 - ▶ <http://www-01.ibm.com/software/awdtools/doors/webaccess/>
- Инструкция пользователя - Rational DOORS Requirements Framework Add-On User's Guide (англ.):
 - ▶ http://publib.boulder.ibm.com/infocenter/rsdp/v1r0m0/index.jsp?topic=/com.ibm.help.download.doors.doc/topics/doors_version9_2.html
- Русскоязычная версия руководства пользователя DOORS (6 МБ) .
 - ▶ <http://download.boulder.ibm.com/ibmdl/pub/software/dw/ru/telelogic/doors-manual.pdf>
- Инструментальные средства (включая DOORS) для поддержания жизненного цикла разработки приложений:
 - ▶ <http://download.boulder.ibm.com/ibmdl/pub/software/dw/ru/telelogic/aim-development.pdf>
- Брошюра, поясняющая как формировать хорошие требования:
 - ▶ <http://download.boulder.ibm.com/ibmdl/pub/software/dw/ru/download/tenstepstobetterm.pdf>
- Решение IBM Rational для системного и программного инжиниринга
 - ▶ Текст - http://public.dhe.ibm.com/software/dw/ru/download/IBM_Rational_Workbench_09.11.2010.pdf
 - ▶ Видео - <http://ibmtdemo.edgesuite.net/software/rational/demos/RationalSystemsWB/RationalSystemsWB.html>



Анатолий Волохов

anatoly.volkhov@ru.ibm.com

© Copyright IBM Corporation 2008. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.