

**Разработка, проверка и согласование  
технических заданий на всех  
уровнях разработки  
программно-аппаратных систем с  
использованием  
формальных исполняемых моделей  
на языке моделирования UML 2.0  
и среды разработки IBM Rational Rhapsody**

**Дмитрий Рыжов**

**Руководитель  
направления IBM Rational  
SWD Software**



# О компании SWD Software

## Миссия компании

Обеспечение разработчиков **сложных систем, встраиваемых устройств и приложений реального времени** надежной программно-аппаратной платформой и **эффективным инструментарием** для создания специализированных систем любой сложности

## • Общая информация

- создана в 1991 году
- офисы в Санкт-Петербурге, Москве, Польше
- Официальный дистрибьютор QNX
- Бизнес партнер IBM по Rational Software
- Продвижение Rhapsody с 2005 года

## • Основные направления деятельности

- Предпродажные консультации, Поставка, Поддержка, Обучение и Консультации по инструментам для разработки

# Содержание

- Разработка систем – сегодня
- От управления требованиями к разработке систем на основе моделей
- Разбор примера создания технического задания на разработку системы контроля доступа

# Разработка систем - Сегодня

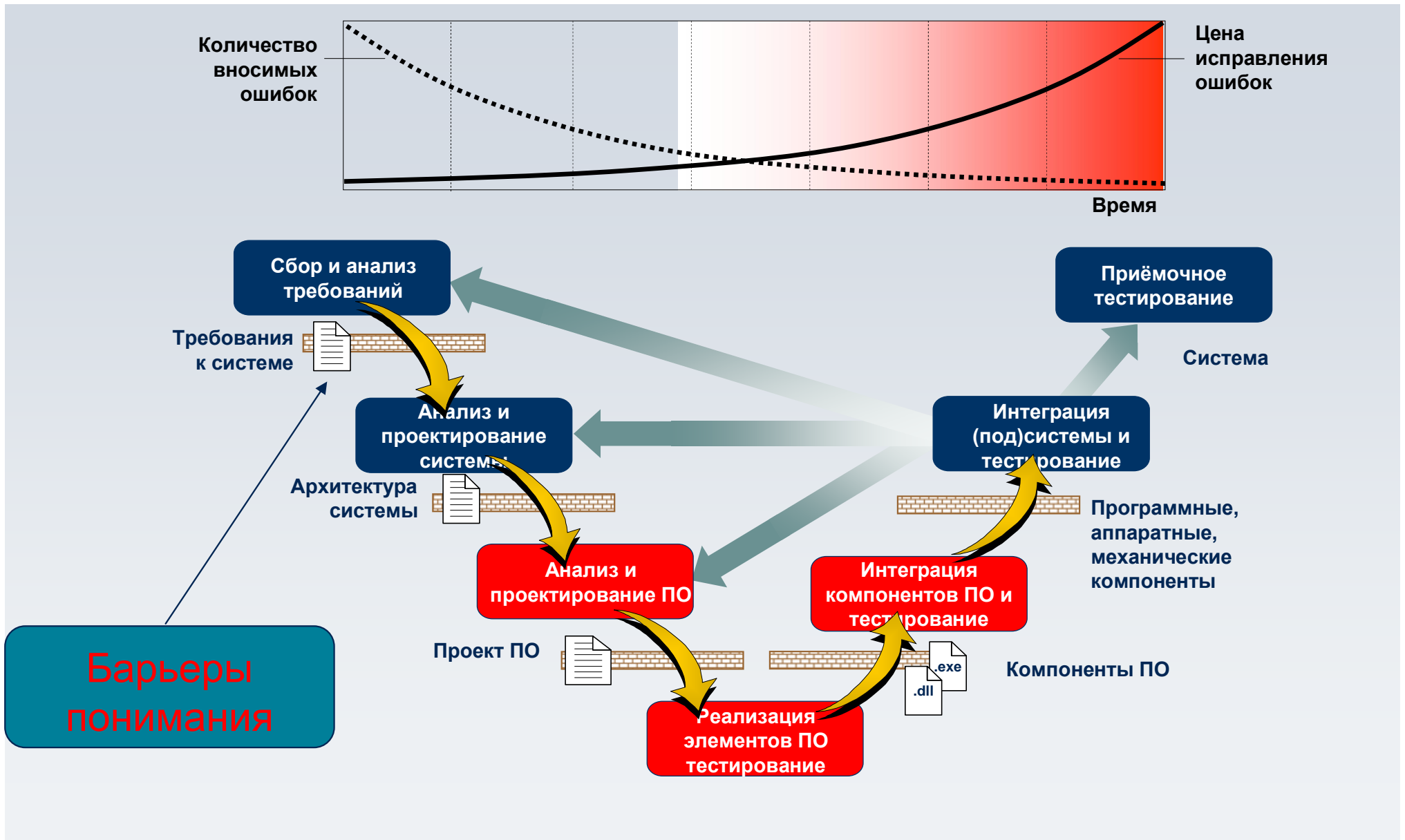
Подход на основе документации  
Так это должно работать...



- Технические задания
- Требования к интерфейсам
- Проекты системы, подсистем, ПО, аппаратуры
- Анализ возможных решений
- Планы тестирования

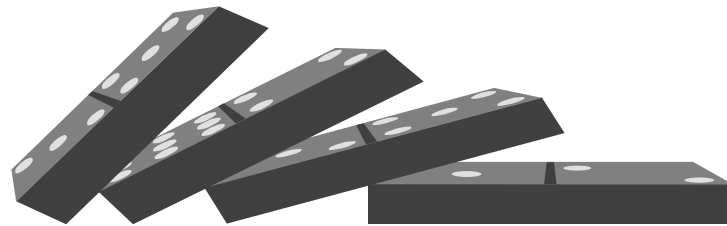


# Вот так это реально работает ...



# Что и где происходит не правильно?

- Требования плохо осознаются заинтересованными лицами или аналитиками?
- Требования плохо определены и содержат ошибки?
- Соответствие требованиям не проверено при тестировании?
- Разработчики пропустили требование?
- Разработчики не правильно поняли требование?
- Пользовательские требования изменились, но их влияние на реализацию и тестирование не было учтено?



# Требования - Что это такое?

- Любое утверждение, которое должно иметь место в разработанной системе
- Часто выражается в текстовом виде
- Формулируются в процессе разработки системы и является отправной точкой для разработки

# Как же все таки разработать качественные требования?

Мы теперь знаем

- Как хранить
- Как управлять требованиями
- Как отслеживать изменения в требованиях

Но пока не знаем

- Как разработать качественные требования

И ответом на этот вопрос является МОДЕЛИРОВАНИЕ !

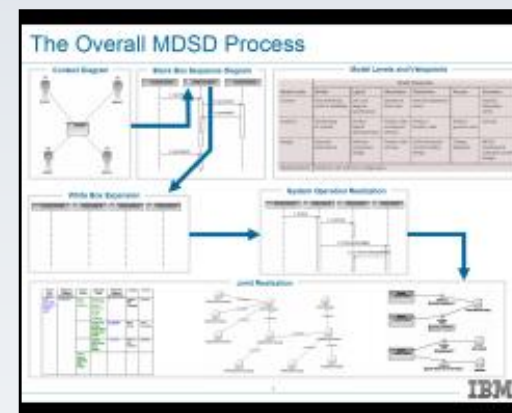
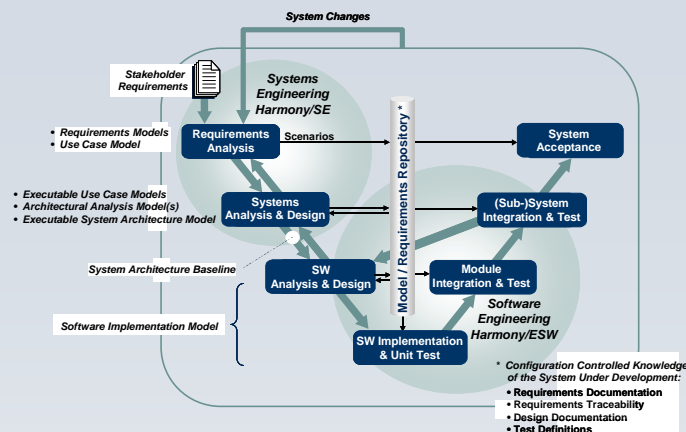


# Разработка систем на основе моделей

Визуальный подход для превращения требований в надежную систему

## Разработка на основе моделей:

- Структурный подход к разработке систем для управления сложностью путем абстрагирования
- Использует модели как основные артефакты для разработки систем
- Объединяет конструкторов, электронщиков и программистов
- Позволяет гарантировать, что система соответствует всем требованиям
- Улучшает коммуникации в команде системных инженеров



# Изображение содержит тысячу слов?

## 2 Функциональные требования

### 2.1 Включение двигателя

#### 2.1.1 Движение автомобиля

##### 2.1.1.1 Движение вперед

*Автомобиль должен позволять двигаться вперед со скоростью в диапазоне от 0 до 200 км/ч при движении по стандартной ровной дороге и скорости ветра 0 км/ч, развивая при этом мощность 180 л.с.*

##### 2.1.1.2 Движение назад

*Автомобиль должен позволять двигаться назад с максимальной скоростью 20 км/ч при движении по стандартной ровной дороге и скорости ветра 0 км/ч, развивая при этом мощность 180 л.с.*

### 2.1.2 Разгон автомобиля

*Автомобиль должен разгоняться с места до 100 км/ч за 10 сек. при движении по стандартной ровной дороге и скорости ветра 0 км/ч.*

*Автомобиль должен разгоняться со скорости 100 км/ч до 150 км/ч с ускорением 5 км/с при движении по стандартной ровной дороге и скорости ветра 0 км/ч.*

*Автомобиль должен разгоняться со скорости 150 км/ч до 200 км/ч, с ускорением 3 км/с при движении по стандартной ровной дороге и скорости ветра 0 км/ч.*

## 2.2 Управление автомобилем

### 2.2.1 Включение зажигания автомобиля

*В автомобиле должна быть предусмотрена возможность идентифицировать людей, имеющих право включать зажигание и управлять автомобилем.*

### 2.2.2 Управление скоростью

*Для управления скоростью в автомобиле должен использоваться ножной механизм. Скорость должна свободно регулироваться в диапазоне от нуля до максимальной. Скорость автомобиля должна контролироваться средствами автоматики.*

### 2.2.3 Торможение автомобиля

*При скорости 10 км/ч автомобиль должен останавливаться за 2 сек.*

*При скорости 30 км/ч автомобиль должен останавливаться за 6 сек.*

*При скорости 100 км/ч автомобиль должен останавливаться за 30 сек.*

*При скорости 200 км/ч автомобиль должен останавливаться за 45 сек.*

# Изображение содержит тысячу слов?

2 Функциональные требования

2.1 Включение двигателя

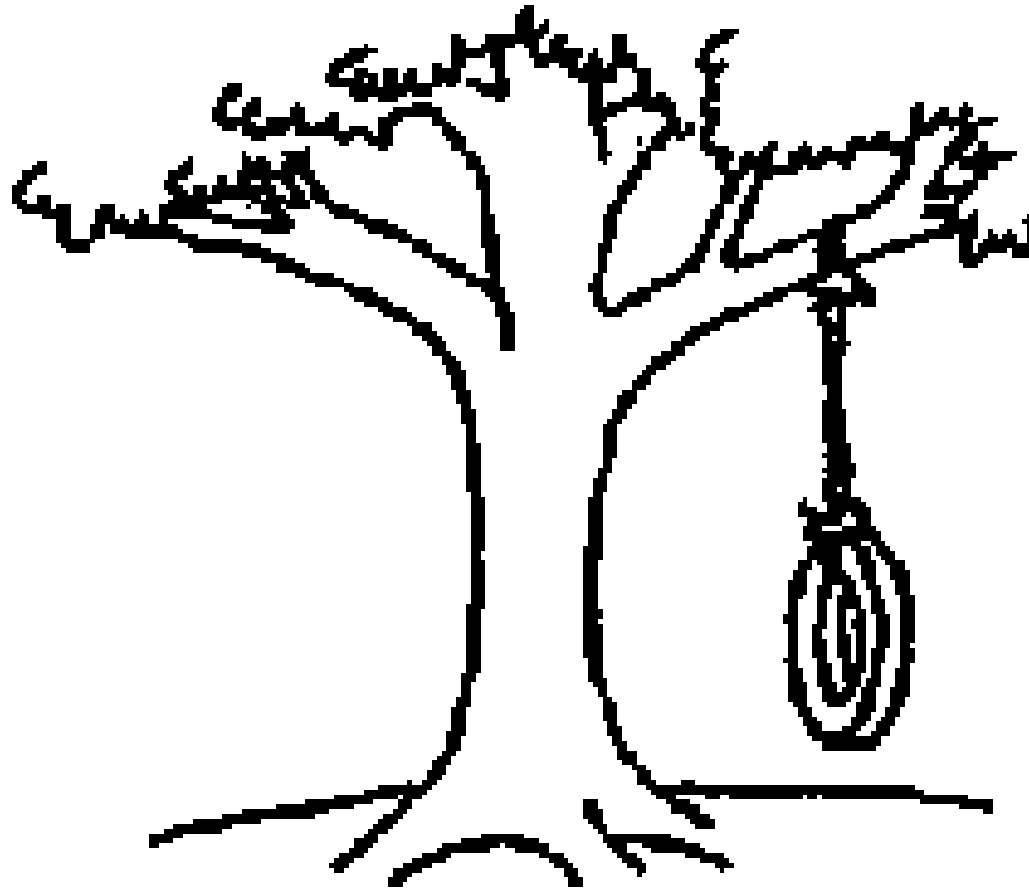


*При скорости 30 км/ч автомобиль должен останавливаться за 6 сек.*

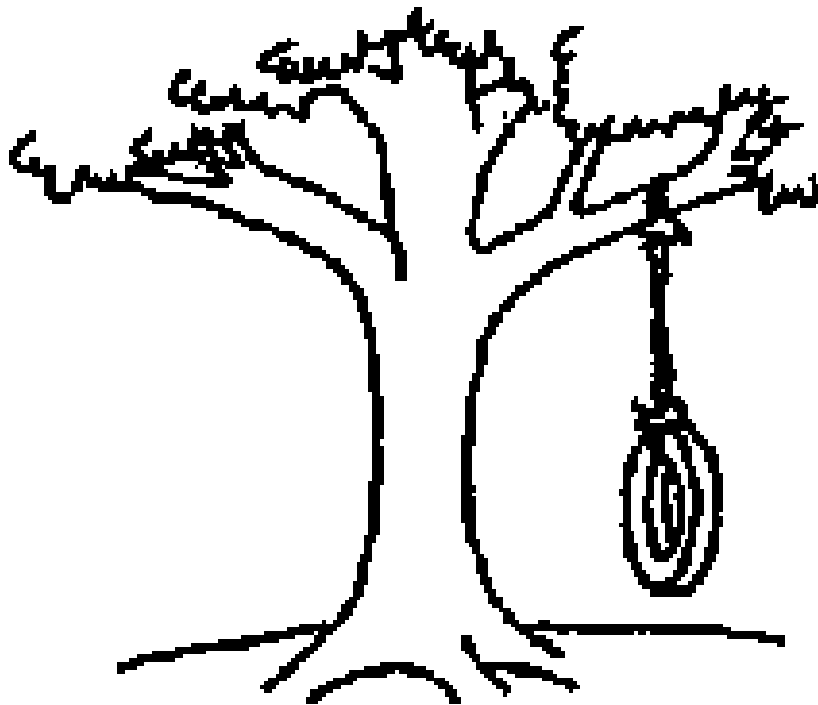
*При скорости 100 км/ч автомобиль должен останавливаться за 30 сек.*

*При скорости 200 км/ч автомобиль должен останавливаться за 45 сек.*

...или скрывает тысячу требований?



## ...или скрывает тысячу требований?



*Качели должны выдерживать вес до 80 кг*

*На качелях должны помещаться двое маленьких детей*

*Качели должны быть подвешены не ниже 0,5 м над землей*

*Качели не должны иметь возможности раскачиваться больше чем на 180 градусов*

*Упругость веревки для качелей должна равняться.....*

*Качели должны удовлетворять следующим нормам безопасности.....*

*Качели должны удовлетворять следующим дополнительным требованиям .....*

# Уровни и виды требований

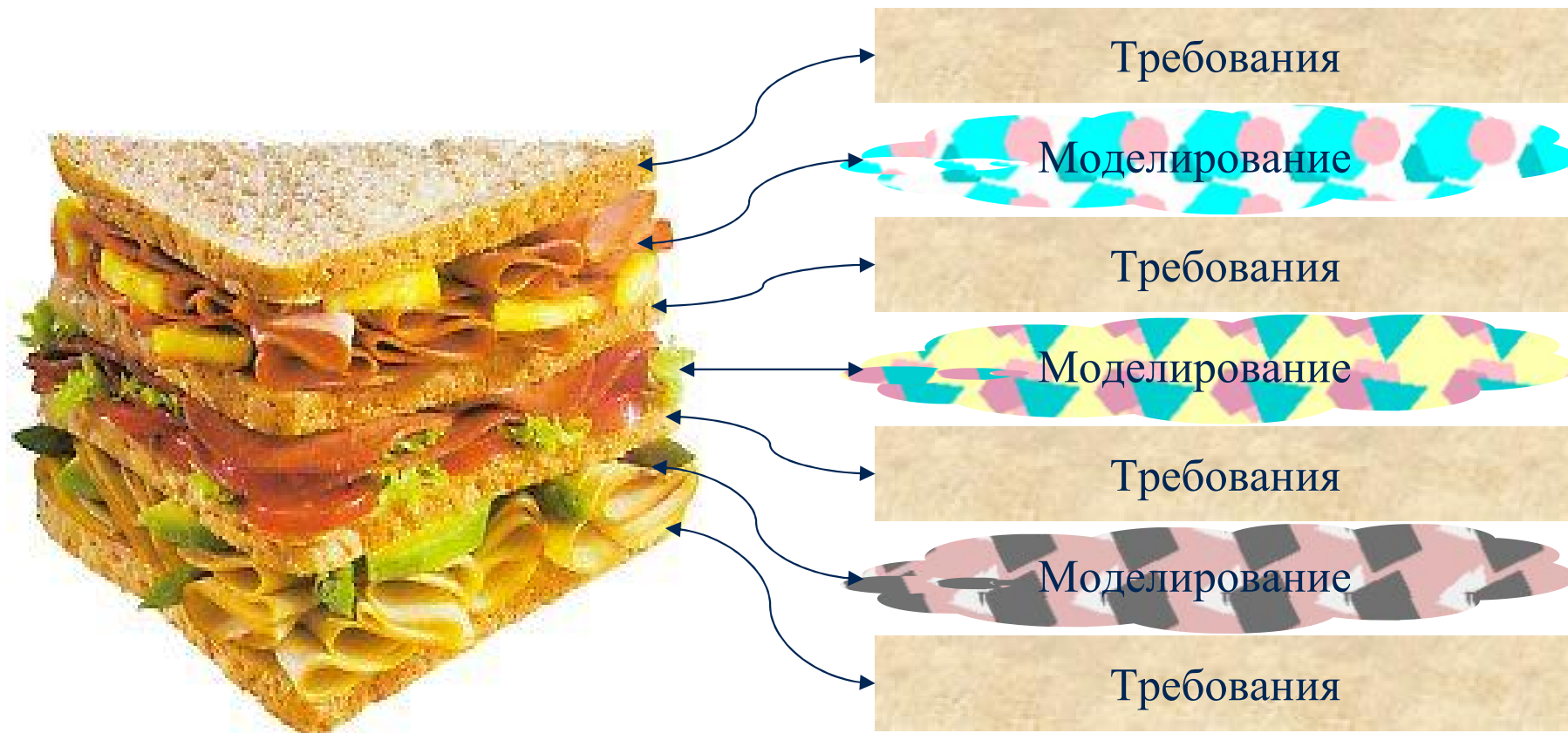
Требования сопровождают разработку на всех уровнях

- Требования пользователей
- Требования заказчика
- Требования к системе
- Требования к подсистемам
- Требования к программным/аппаратным/механическим компонентам

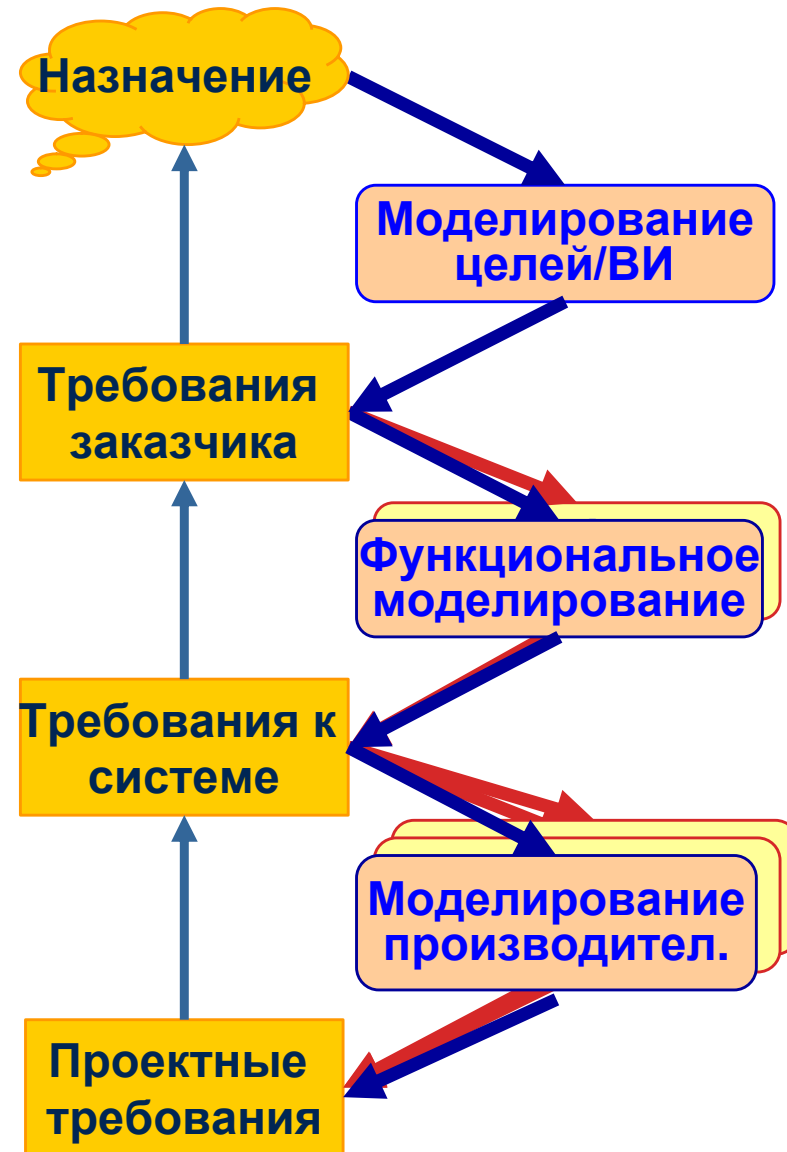
Требования бывают различного вида

- Функциональные требования
- Операционные требования
- Требования к качеству
- ...

# Слоенный пирог при разработке систем



# Моделирование соединяет уровни требований





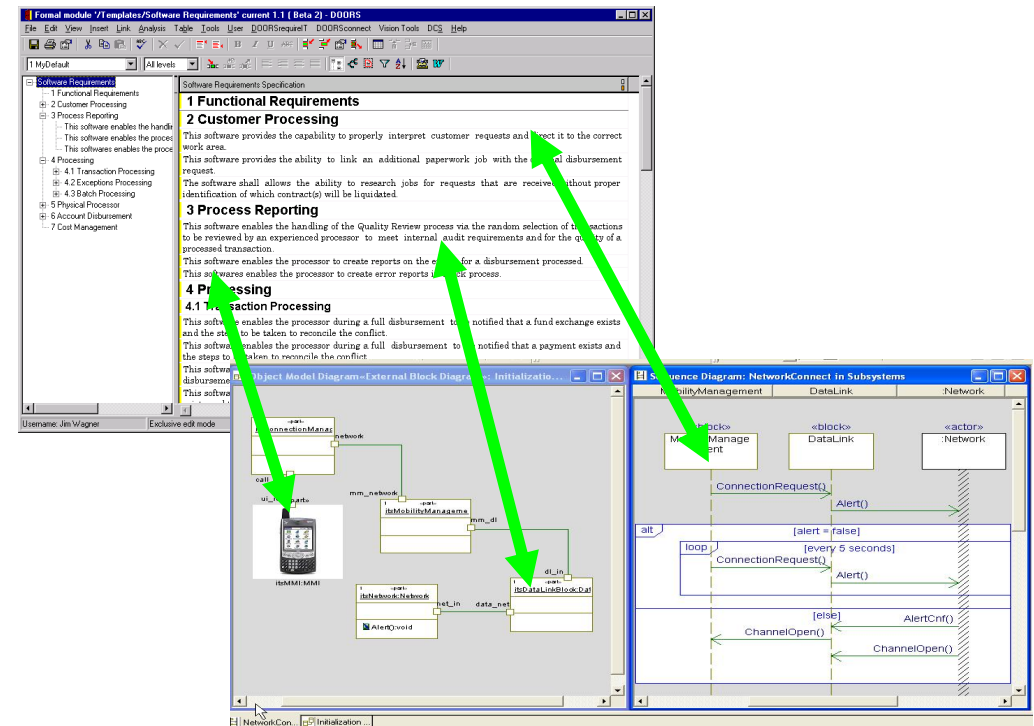
# Симуляция вместо подхода делаем-переделываем!

- Никто не создает готовый самолет до проверки, что он удовлетворяет требованиям по аэродинамике
- Зачем же мы часто ждем когда будет написано ПО и до того момента не проверяем, что проект удовлетворяет требованиям
- Поиск и исправление ошибок на этапах анализа требований и проектирования на порядки дешевле



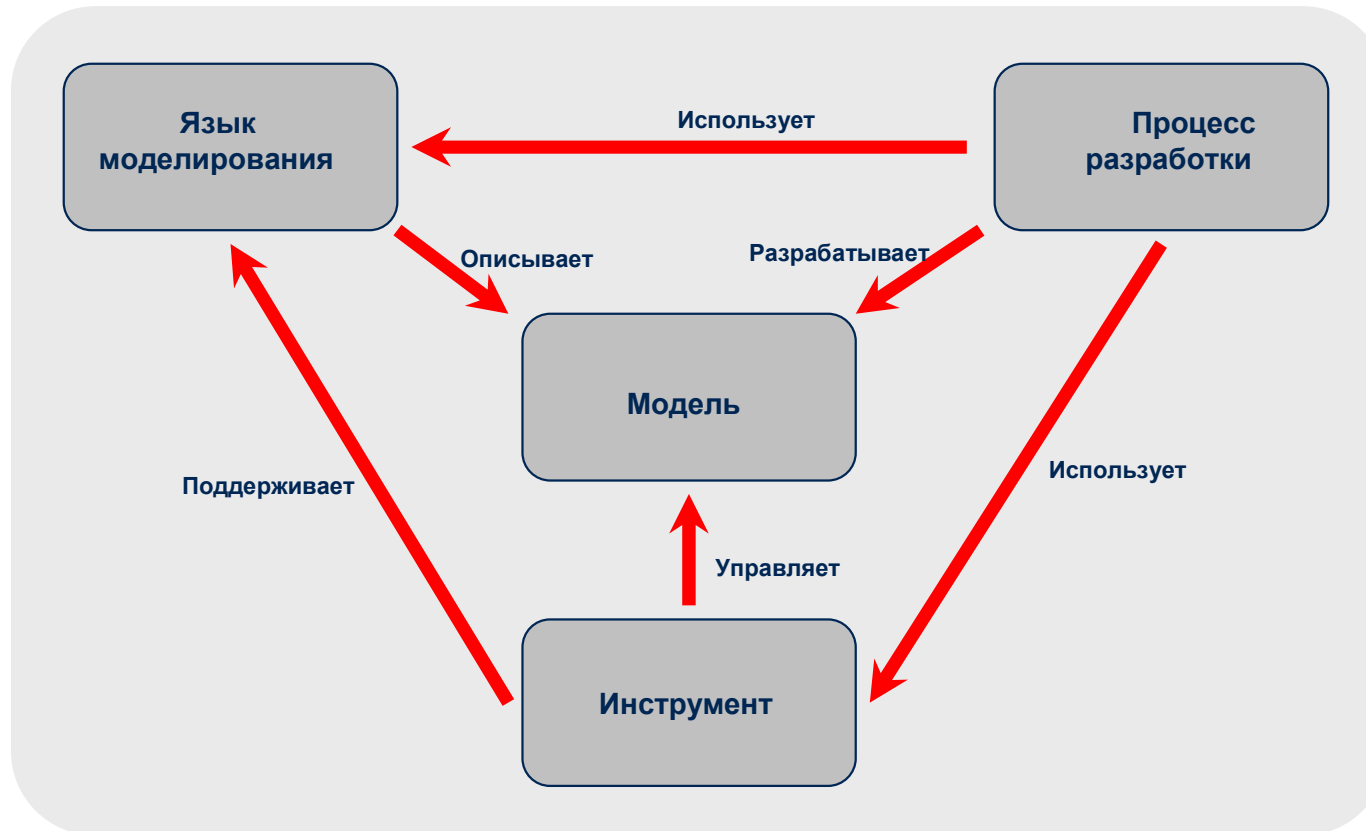
# Трассировка между требованиями и моделью

- Трассировка позволяет отследить соответствие модели требованиям
- Легко проверить:
  - Наличие требований, которым нет соответствия в модели
  - Наличие элементов модели, не имеющих привязки к требованиям
- Быстрый и полный анализ влияния
  - Оценка влияния изменений ДО их внесения
  - Гарантия, что одобренные изменения полностью реализованы



# Составляющие подхода к разработке на основе моделей

- Центральным артефактом является модель



**Язык** – определяет нотацию (визуальное представление) и семантику (значение)

**Инструмент** – среда используемая для создания и управления моделью

**Процесс** – определяет множество последовательностей работ, ролей и результирующих артефактов

\* Данные компоненты независимы, но очень сильно влияют друг на друга

# Диаграммы языка SysML

Диаграмма требований

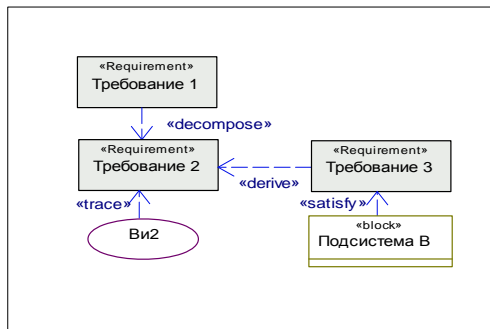


Диаграмма использования

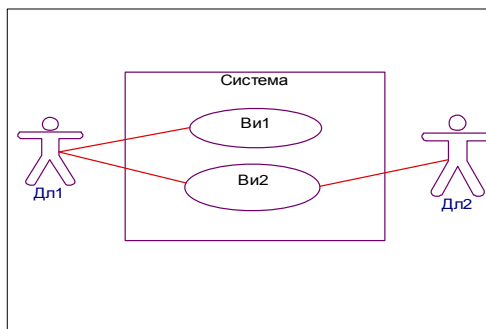


Диаграмма деятельности

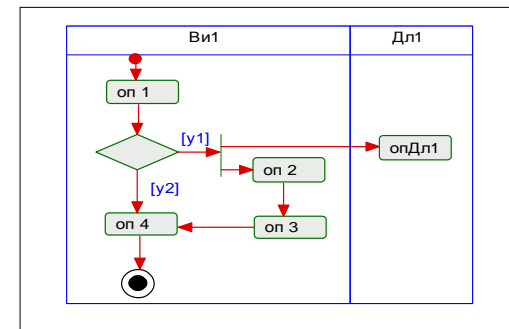


Диаграмма определения блоков

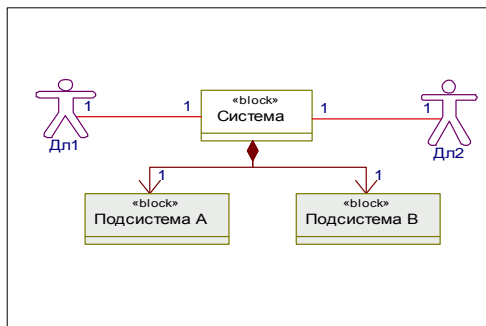


Диаграмма последовательности

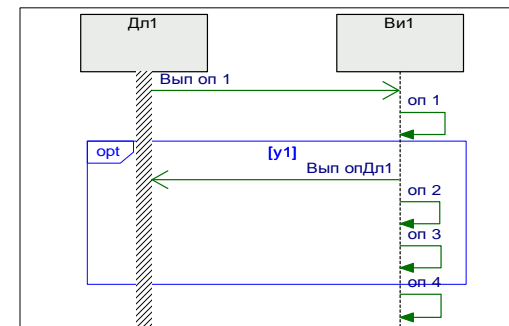


Диаграмма внутренней структуры

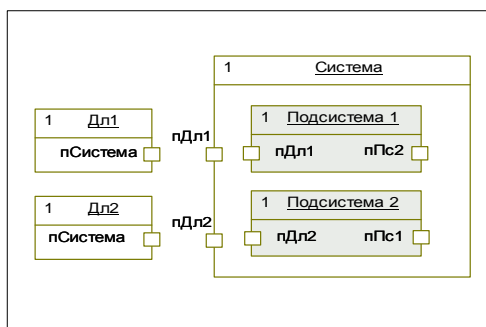
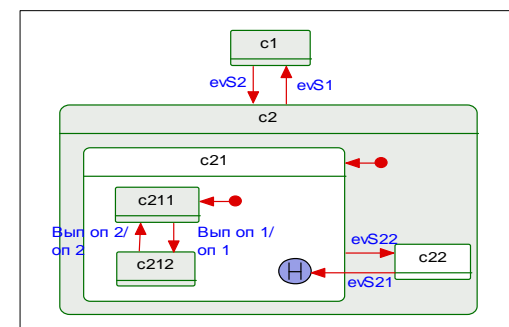


Диаграмма состояний

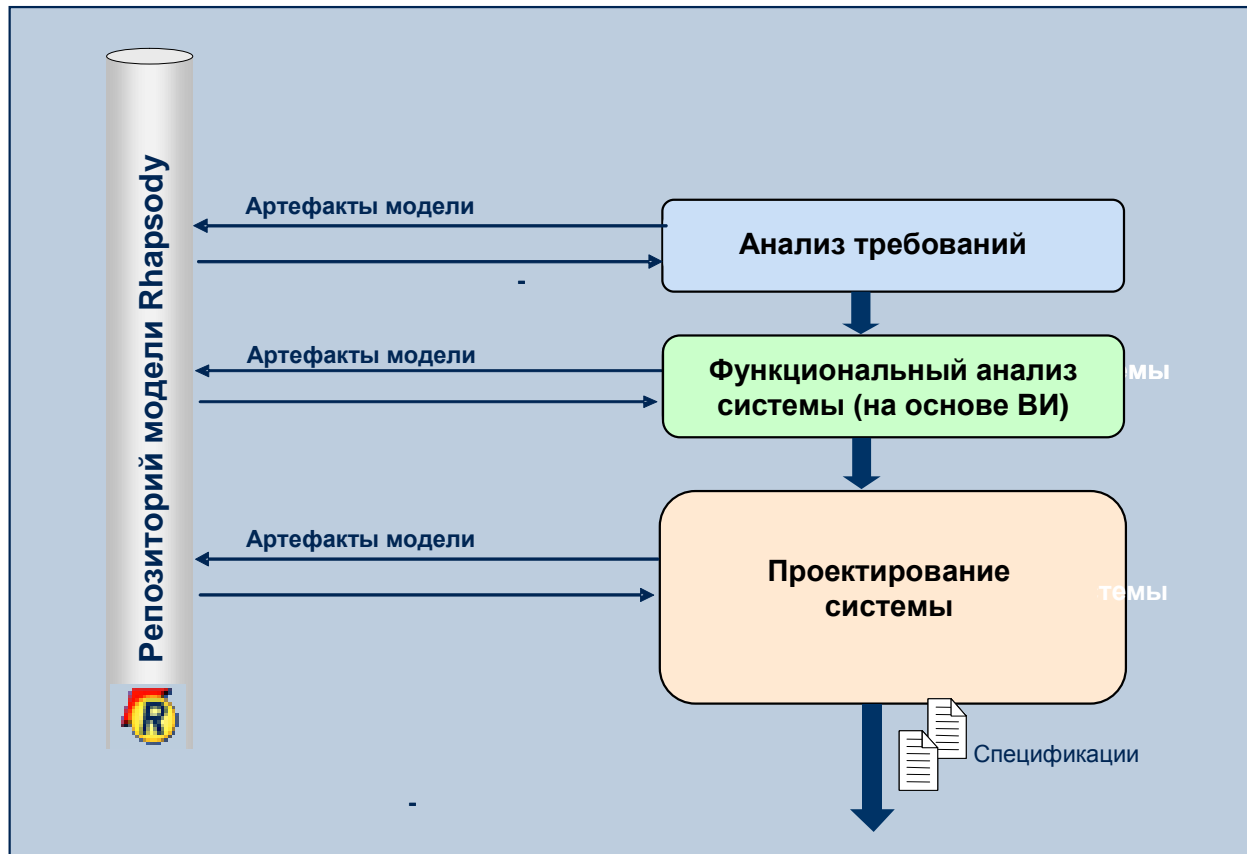


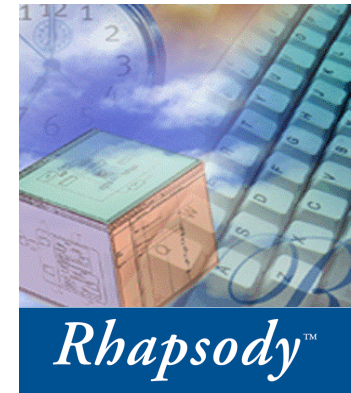
# Поддержка разработки систем на основе моделей в Rhapsody



# Процесс Harmony-SE

- Harmony-SE - процесс разработки систем на основе моделей, который детально описывает как использовать SysML и Rhapsody для выполнения анализа, проектирования, верификации и валидации систем.





# Система контроля доступа

# Уровни требований и моделей в DOORS

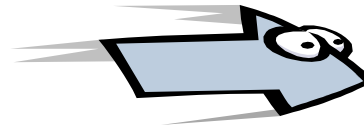
The screenshot shows the DOORS Database application window titled "DOORS Database: /SecuritySystemDemo - DOORS". The interface includes a menu bar (File, Edit, View, Favorites, Tools, Help), a toolbar with various icons, and a "Location" field set to "/SecuritySystemDemo".

The main area displays a tree view on the left with "DOORS Database" expanded to show "Demos" and "SecuritySystemDemo". The right pane shows a table of requirements and models:

Name	Type	Description
1. Требования заказчика	Formal	
2. Техническое задание на систему	Formal	
3. Модель архитектуры системы	Formal	SecuritySystem.rpy
4. Техническое задание на подсистему	Formal	
5. Модель архитектуры подсистемы	Formal	
6.1 Техническое задание на разработку ...	Formal	
6.2 Техническое задание на разработку ...	Formal	
6.3 Техническое задание на конструктор...	Formal	
7. Модель разработки ПО	Formal	

At the bottom of the window, the status bar shows "Username: User" and "User type: Database Manager".





---

# Анализ требований

---

# Анализ требований

## Анализ требований

- На этапе анализа требований производится анализ входов процесса
- На основе требований заказчика создаются системные требования, которые определяют:
  - Что система должна делать (*функциональные требования*) и
  - Как хорошо она должна это делать (*требования к качеству*)
- После того как требования хорошо осознаны они группируются по *Вариантам использования*

# Требования заказчика в DOORS

The screenshot displays the DOORS software interface. The title bar reads: "1. Требования заказчика" current 0.0 in /SecuritySystemDemo (Formal module) - DOORS. The menu bar includes: File, Edit, View, Insert, Link, Analysis, Table, Tools, Discussions, User, Help. The toolbar contains various icons for file operations and analysis. The left pane shows a tree view of requirements:

- 1. Требования заказчика
  - 1 Обзор системы
  - 2 Базовые требования
  - 3 Требования к идентификации
  - 4 Мониторинг персонала
  - 5 Требования к чрезвычайным ситуациям

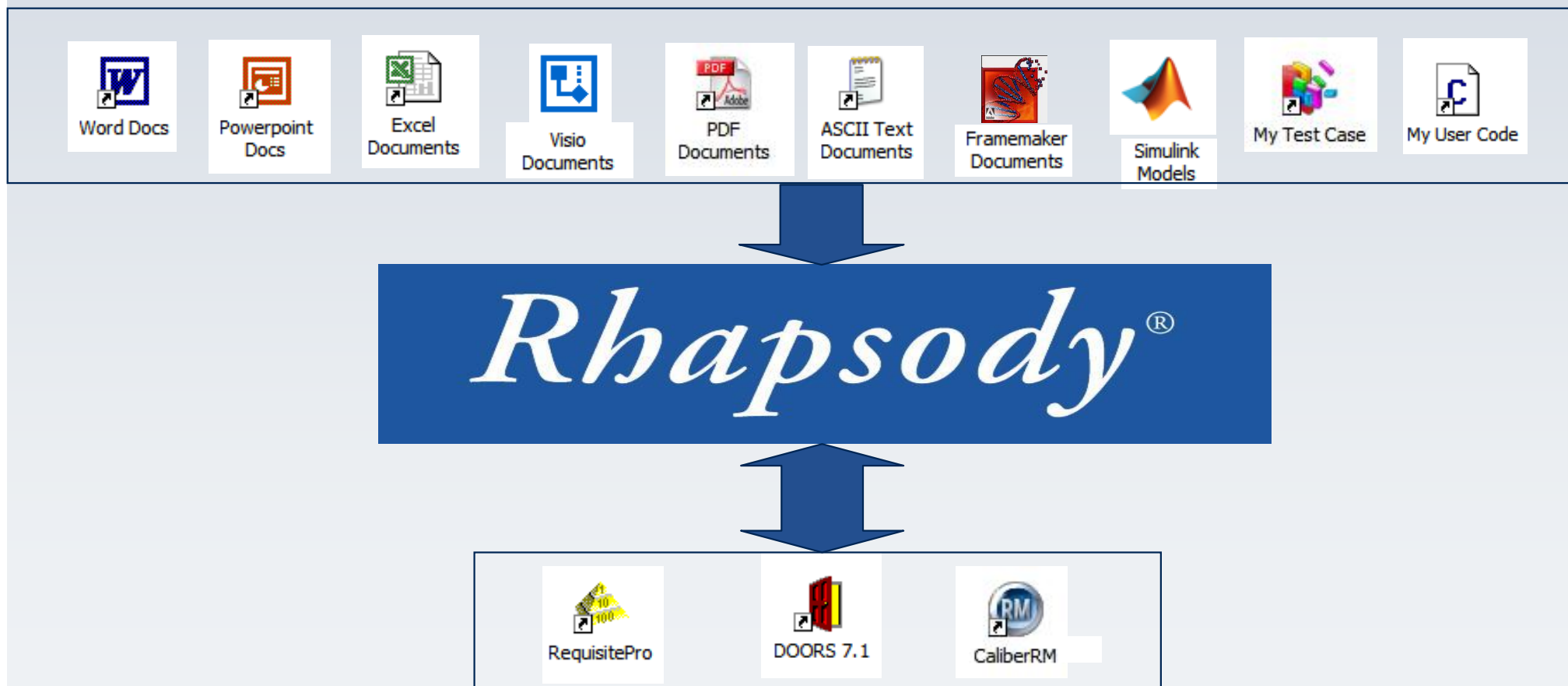
The main pane shows a table of requirements:

ID	Requirement
1	<b>1 Обзор системы</b>
2	<b>1.1 Кратко о системе</b> A security system is to be developed that controls entry and exit to a building through a single point of entry. Identification of personnel will be made by two independent checks. Each person will be photographed upon entry and their time in the building monitored.
3	<b>2 Базовые требования</b>
4	<b>2.1 Контроль доступа</b> Secure areas are to be protected by two independent security checks, one based upon an employee ID and one based upon biometric data. Access to secure areas will be unavailable until the users ID is confirmed. The time between the two independent security checks will not exceed a configurable period. The user is allowed three attempts at biometric and / or card identification before access is completely disabled. Any denied access attempt is to be recorded and sent to the administrator.
5	<b>3 Требования к идентификации</b>
6	<b>3.1 Карта доступа</b> Access will be denied to any user unless he has a valid Security Card. Security cards only contain the employee name and ID and will be renewed yearly. Out of date security cards cause a denial of access.
7	<b>3.2 Сканирование биометрических данных</b> Access will be denied to any user unless their biometric data is recognized. The biometric data is to be stored in the system database and not on the security card.
8	<b>3.3 Время и приоритет доступа</b> The system will only process one user at a time, giving them sufficient time to enter and exit the

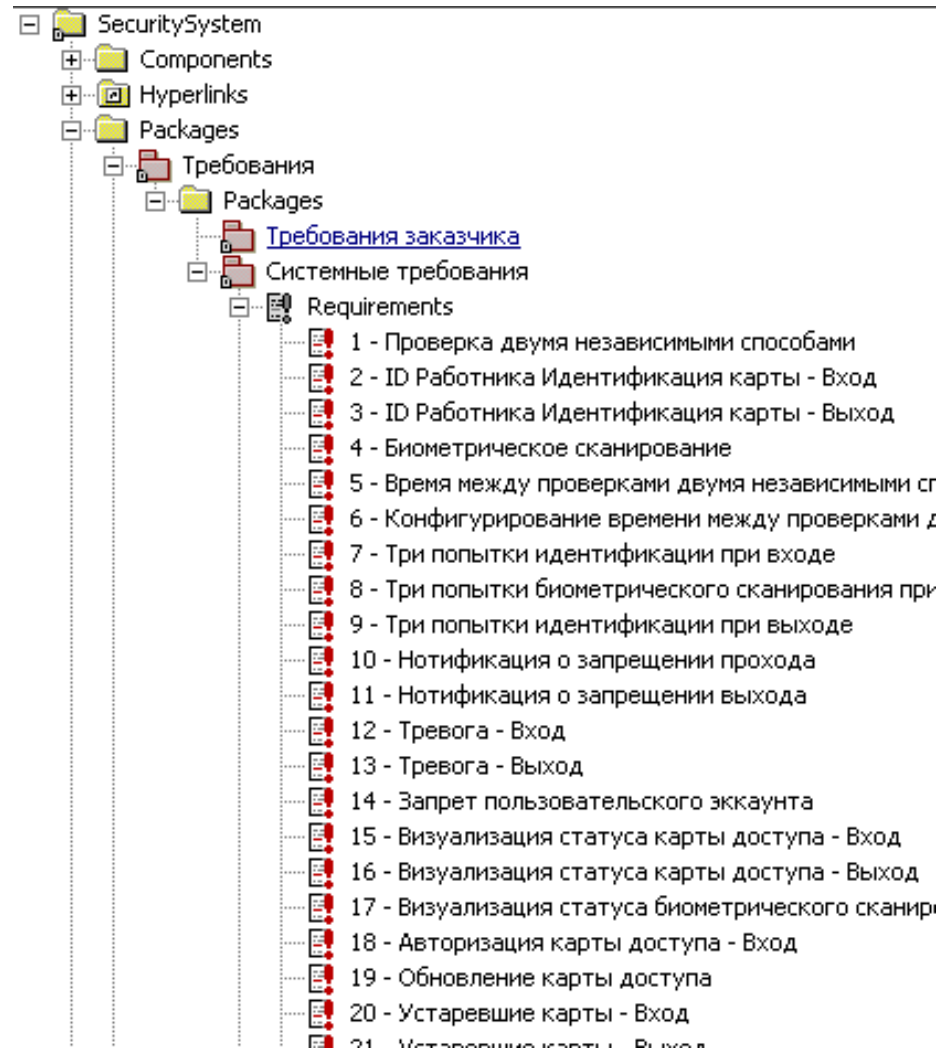
At the bottom of the window, the status bar shows: Username: User, Exclusive edit mode.

# Импорт требований в Rhapsody

- Импорт требований в Rhapsody из различных источников
- Создание и изменение требований в Rhapsody
- Синхронизация требований с системами управления требованиями

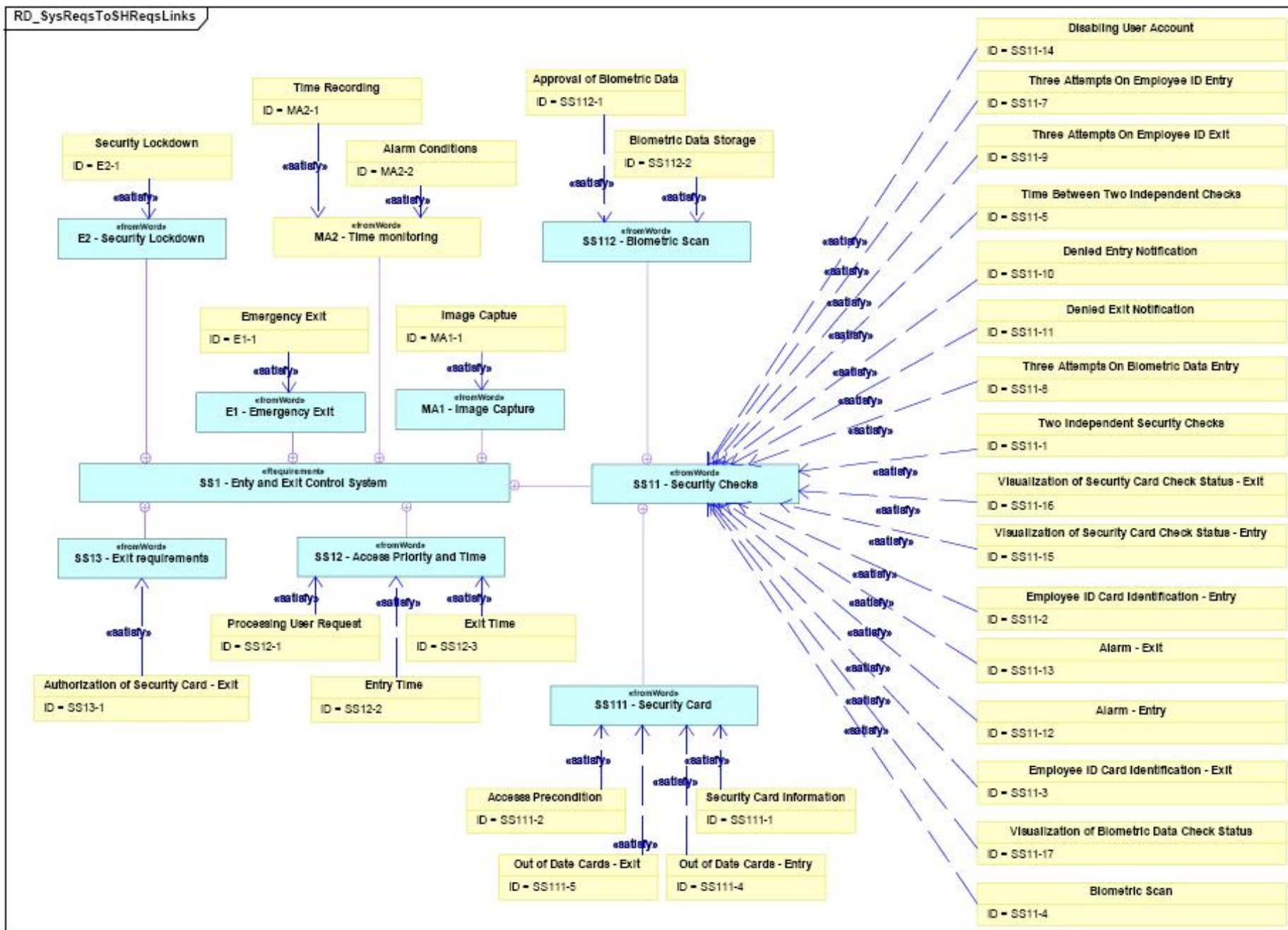


# Определение требований к системе в Rhapsody



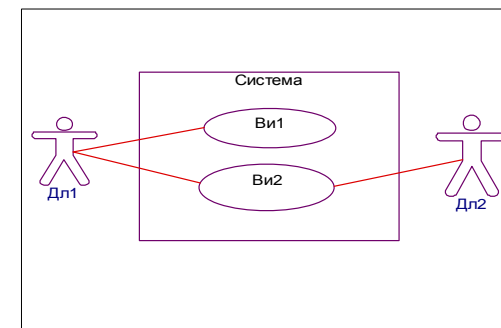


# Привязка требований к системе к требованиям заказчика



# Определение границ и окружения системы

- Где проходят границы системы, которую мы разрабатываем?
- Кто является пользователем системы?
- С какими внешними системами взаимодействует разрабатываемая система?

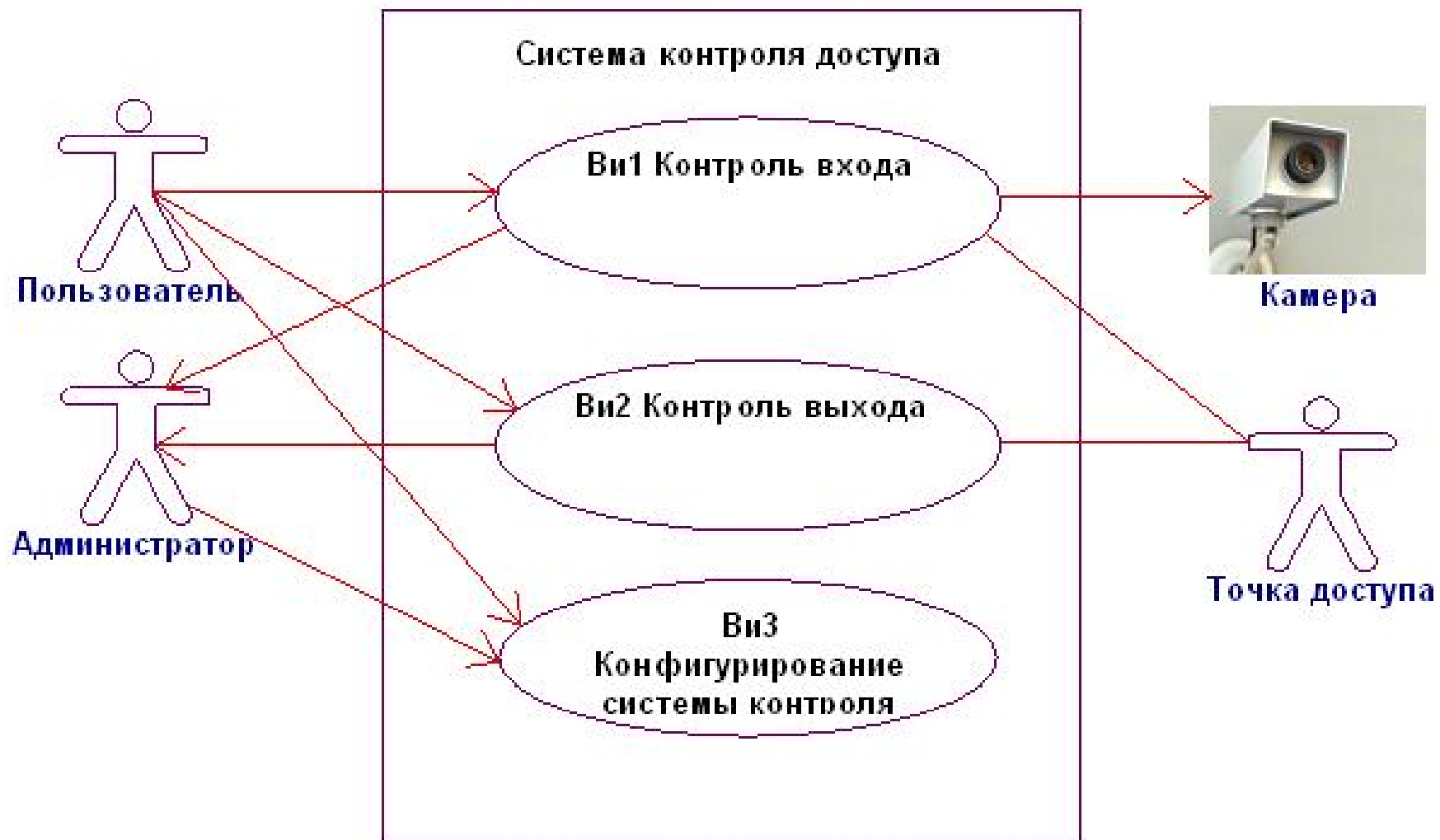


# Определение вариантов использования системы

- Главные возможности системы, имеющие законченную ценность для заказчика
- Позволяют определить цели и привязать разработку к целям
- Лучше реализовать 10% вариантов использования, чем 90% различных компонентов
- Основа для дальнейшего планирования проекта



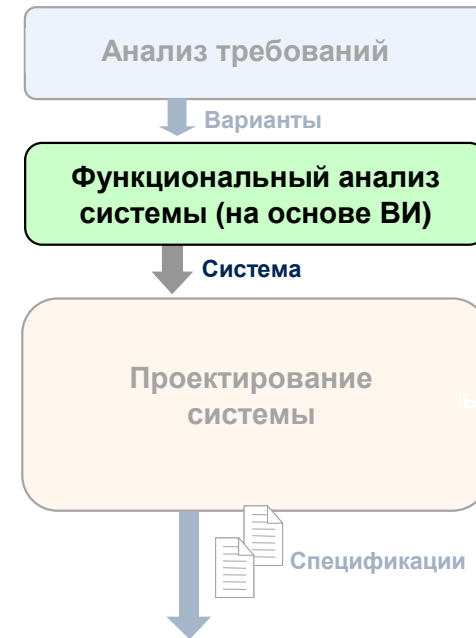
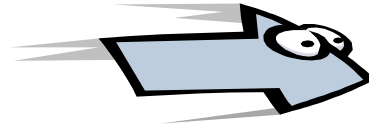
# Варианты использования системы контроля доступа



# Привязка требований к вариантам использования

- Варианты использования становятся главами “вывернутого” технического задания
- Позволяет сфокусироваться на требованиях для отдельного варианта использования системы
- Упрощается процедура проработки и согласования требований технического задания





---

# Функциональный анализ системы

---

# Функциональный анализ системы

Функциональный анализ  
системы (на основе ВИ)

- На этапе функционального анализа системы на основе функциональных требований производится определение согласованного набора функций (операций) системы
- Для каждого варианта использования создается модель и связанные с ним требования верифицируются и валидируются путем исполнения модели

# Детализация требований к варианту использования

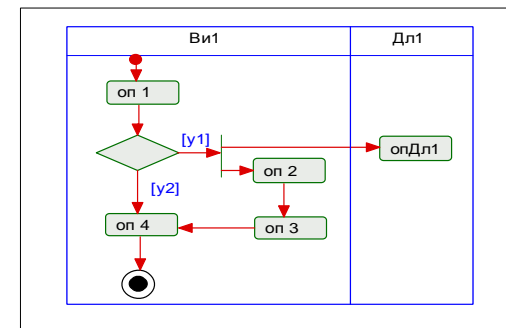
- Текстовое описание варианта использования
- Определение алгоритмов работы системы в ВИ
- Определение сценариев взаимодействия с системой в ВИ
- Определение поведения системы в ВИ

# Определение элементов модели для Ви1 Контроль входа



# Детализация требований на основе определения алгоритмов работы системы в ВИ

- Используются диаграммы деятельности UML
- Описывают последовательность действий, выполняемых системой в процессе работы
- Описывают несколько сценариев работы с системой
- Удобны для описания параллельных сценариев

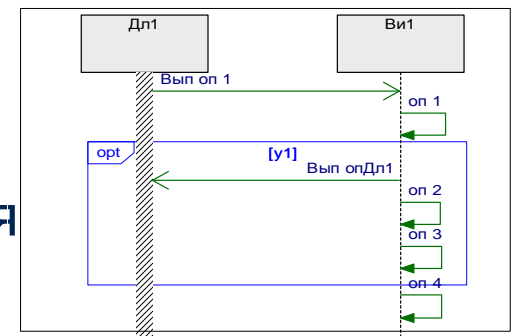




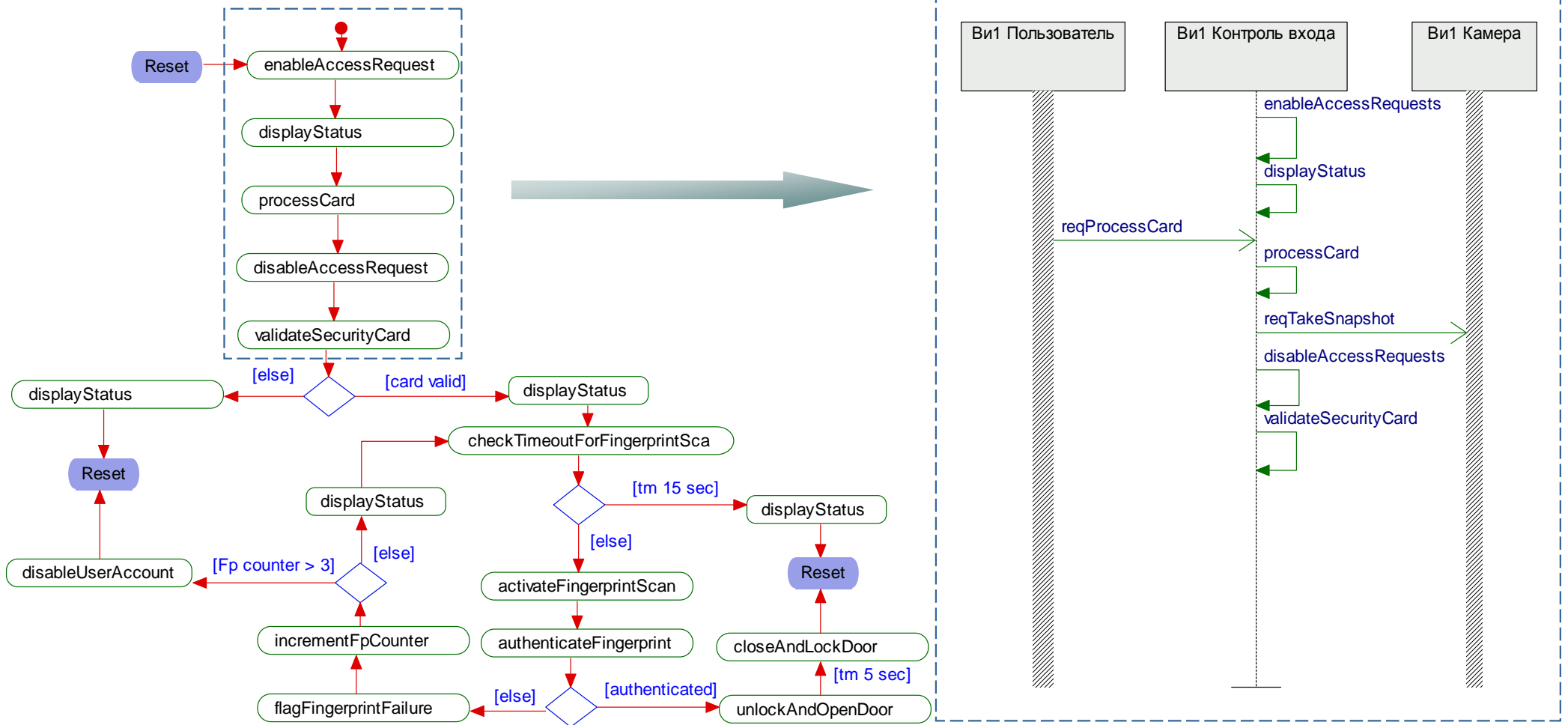


# Детализация требований на основе сценариев

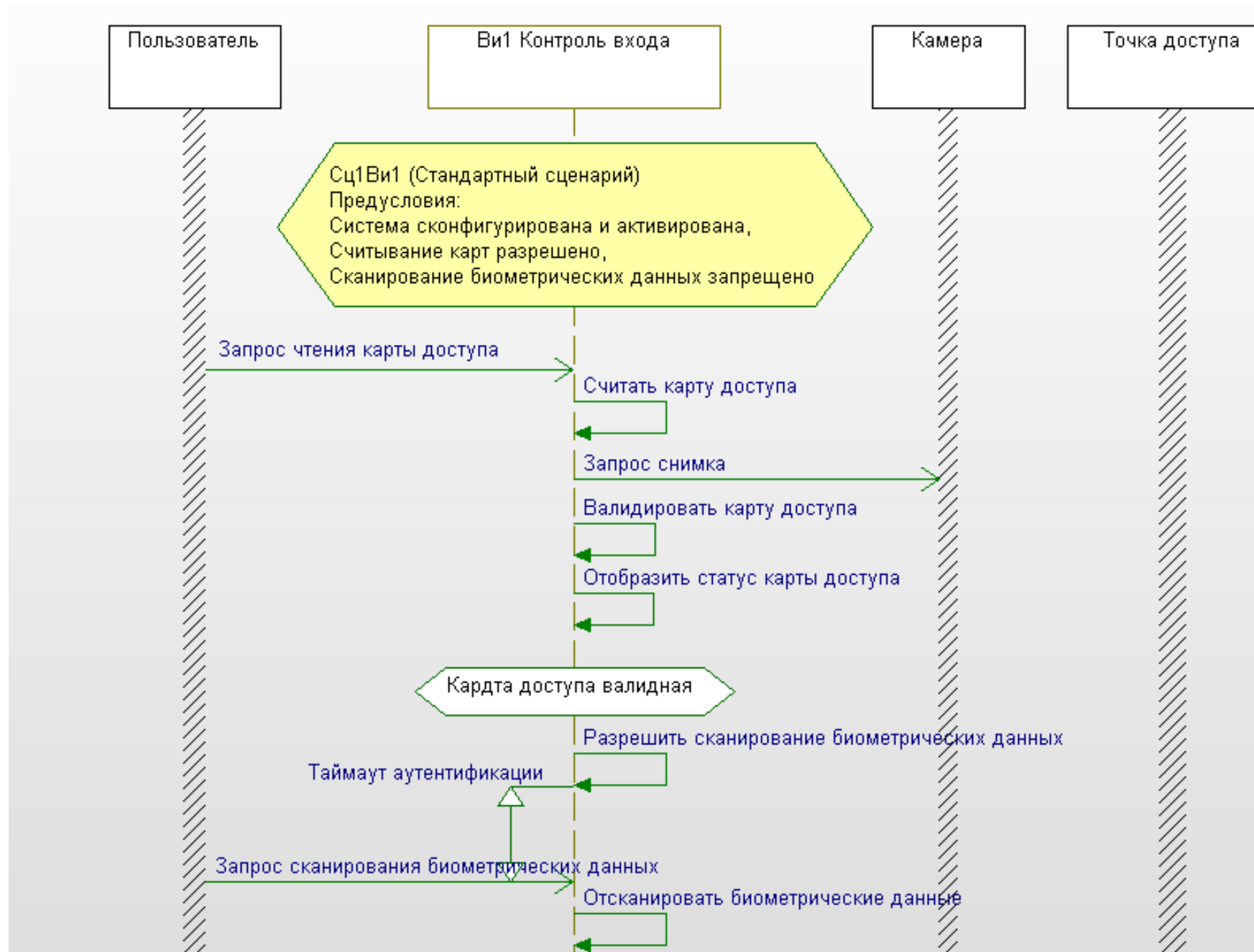
- Используются диаграммы последовательности UML
- Формальное описание взаимодействия между системой и ее окружением в рассматриваемом варианте использования
- Система выступает в виде черного ящика
- Определяют прикладные протоколы взаимодействия с системой
- Множество сценариев для каждого варианта использования
- Сценарии солнечного/дождливого дня
- Понятны нетехническим специалистам
- Позволяет отобразить различные ограничения



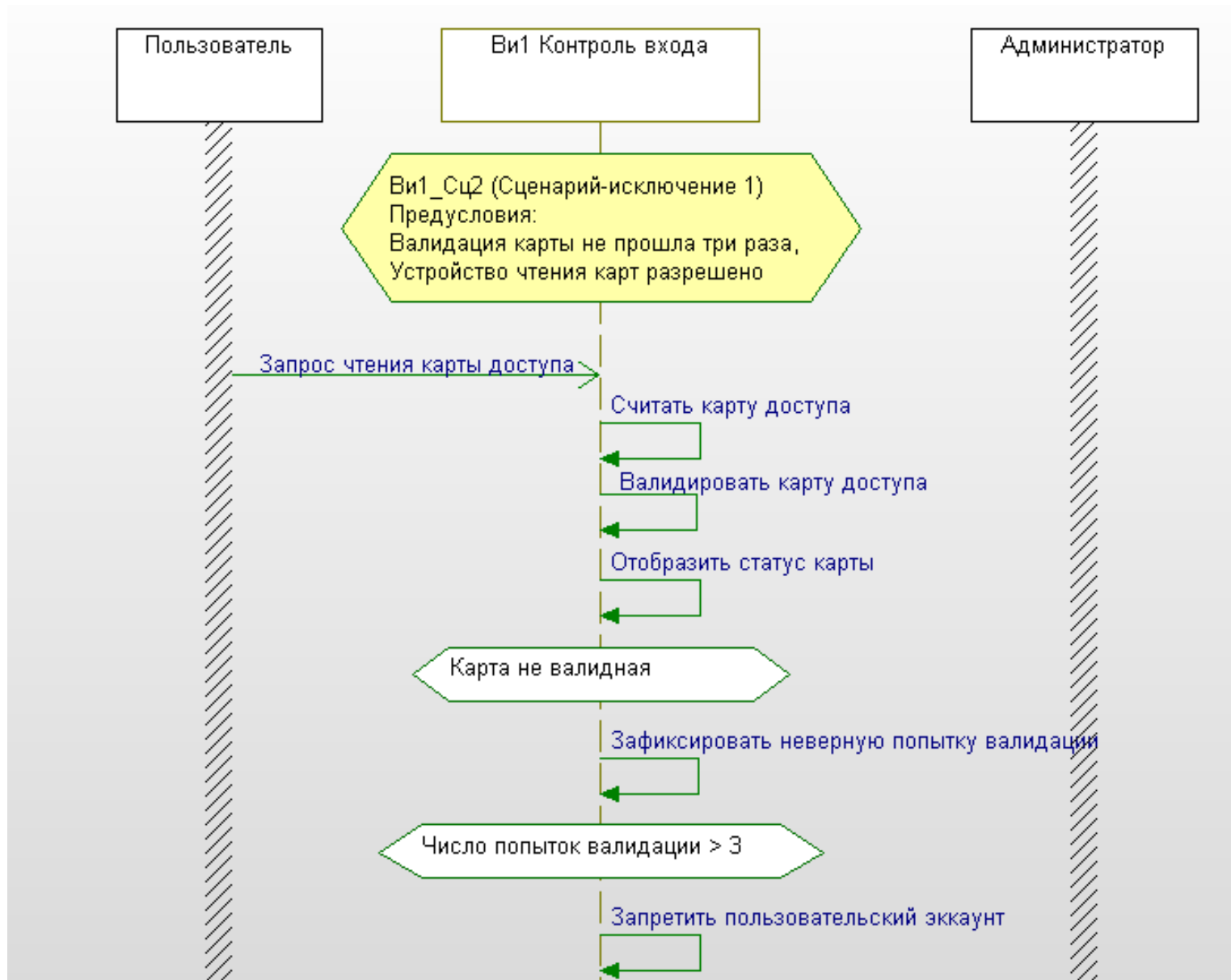
# Получение сценариев ВИ на основе диаграммы деятельности



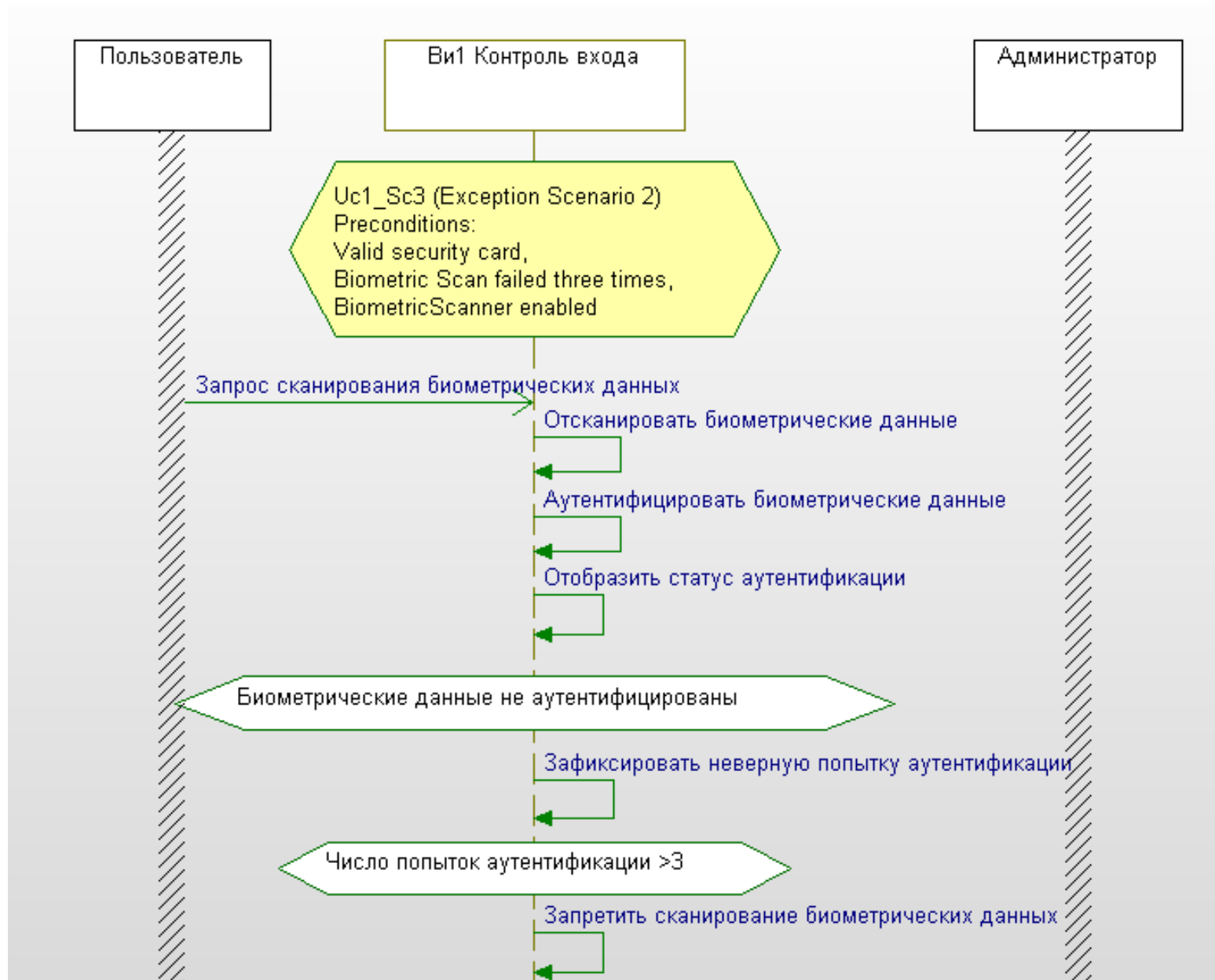
# Ви1 Сценарий 1 Вход разрешен



# Ви1 Сценарий 2 Неверная карта



# Vi1 Сценарий 3 Вход Неверные биометрические данные



# Определение операций и интерфейсов системы

- Сценарии являются эффективным способом определения операций и интерфейсов системы
- На данном этапе определяются ПРИКЛАДНЫЕ интерфейсы между системой и внешним окружением

# Атрибуты и операции системы в ВИ

## Контроль входа

1 Пользователь

*Operations*

- RequestEntry
- ScanFingerprint
- RequestExit

1 Администратор

*Operations*

- reqProcessAlert

1 «block»  
Ви1 Контроль входа

*Attributes*

- Число попыток валидации
- Число попыток аутентификации
- Таймаут аутентификации
- Время открытия двери
- Статус аутентификации
- Статус карты

*Operations*

- Валидировать карту доступа
- Запрос чтения карты доступа
- Запрос сканирования биометриче...
- Событие Дверь открыта
- Событие Дверь закрыта
- Зафиксировать неверную попытку...
- Запретить пользовательский эккаунт
- Считать карту доступа
- Отсканировать биометрические да...
- Аутентифицировать биометрическ...
- Сохранить данные о входе
- Зафиксировать неверную попытку...
- Разрешить сканирование биометр...
- Запретить сканирование биометри...
- Сохранить данные эккаунта
- Отобразить статус карты доступа
- Отобразить статус аутентификации
- Тревога

1 Камера

*Operations*

- Запрос снимка

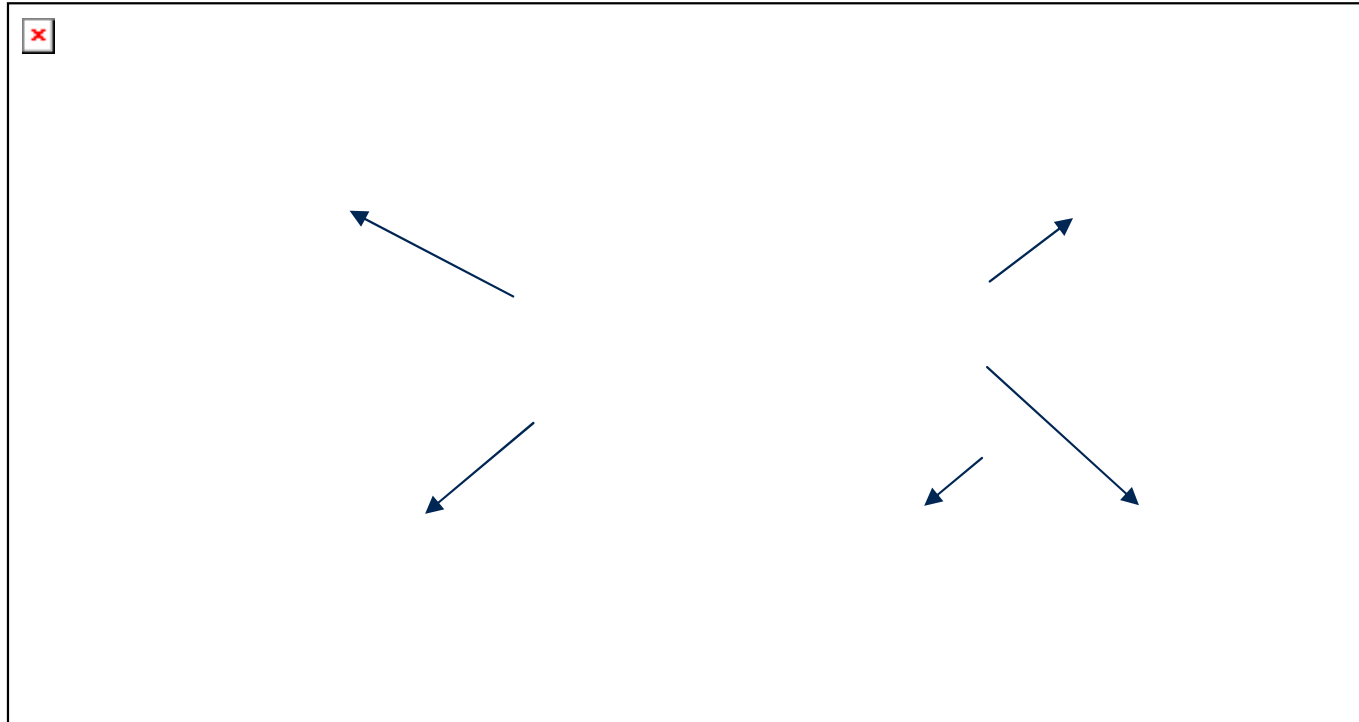
1 Точка доступа

*Operations*

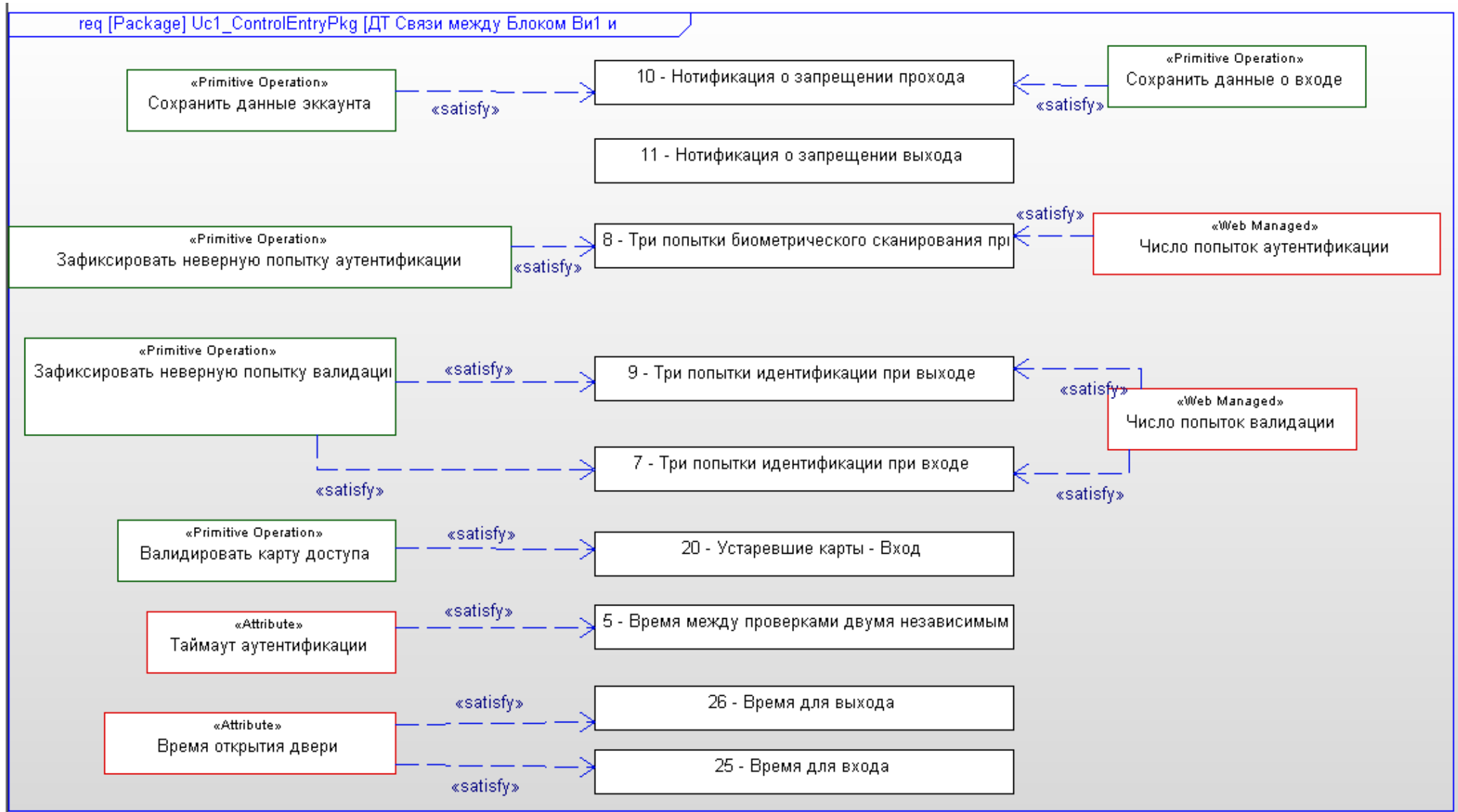
- Запрос открытия двери
- Запрос закрытия двери



# Интерфейсы системы в ВИ Контроль входа

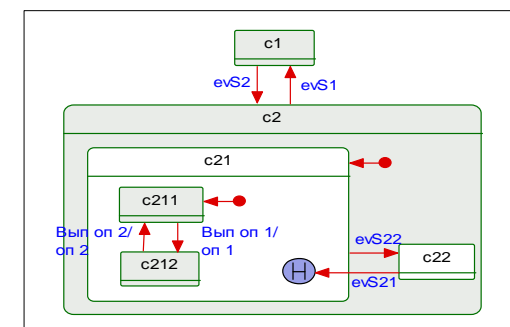


# Привязка атрибутов и операций к системным требованиям

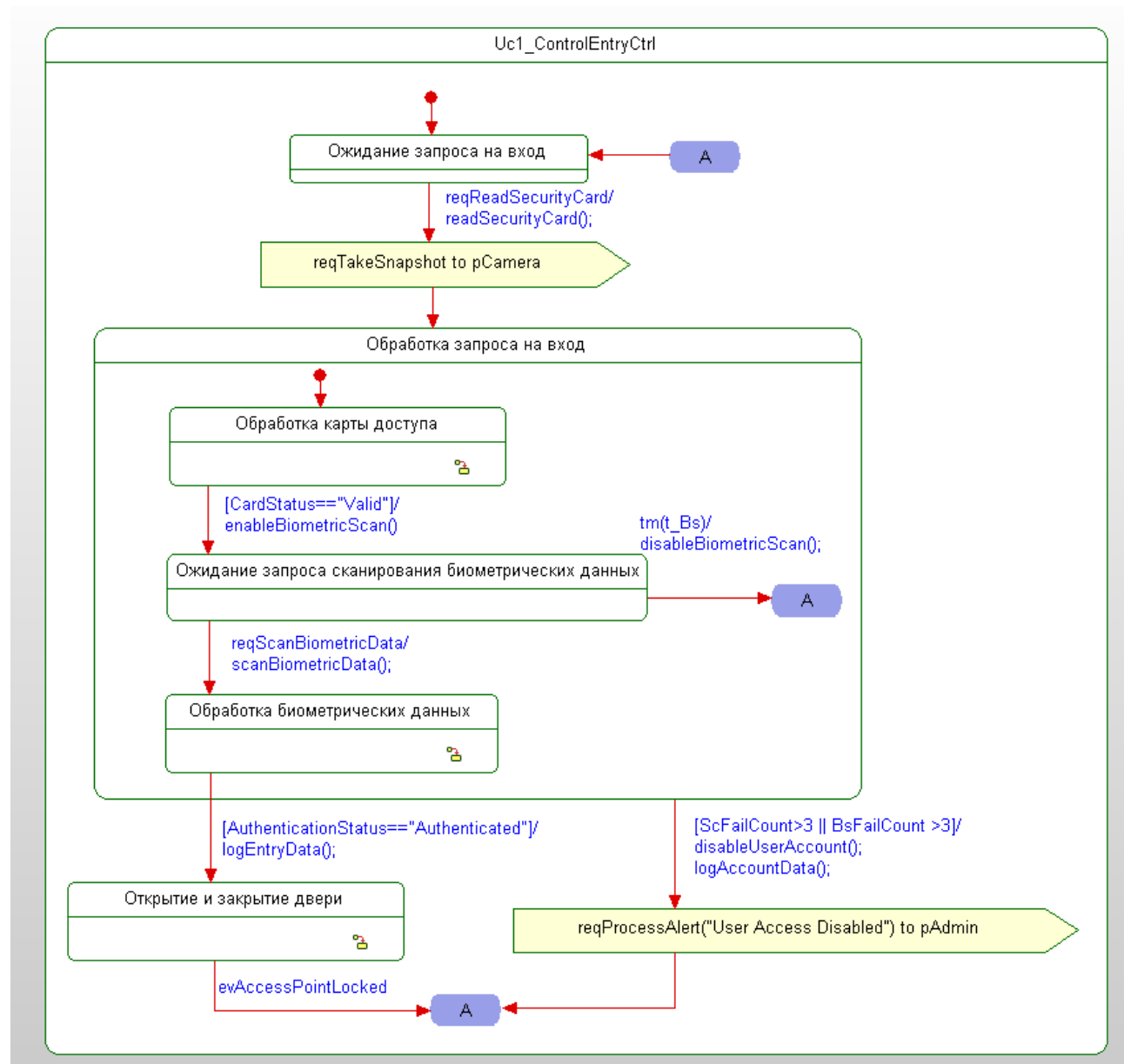


# Детализация требований путем определения поведения на основе конечных автоматов

- Полное формальное описание поведения системы в ВИ
- Определяют состояния/режимы работы системы
- Определяет условия перехода между состояниями/режимами
- Определяет действия выполняемые в состояниях и при переходах
- Является исполняемым описанием



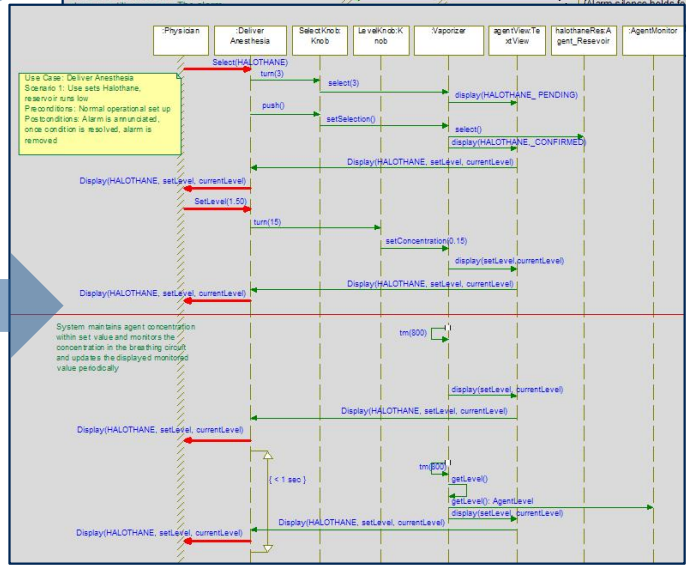
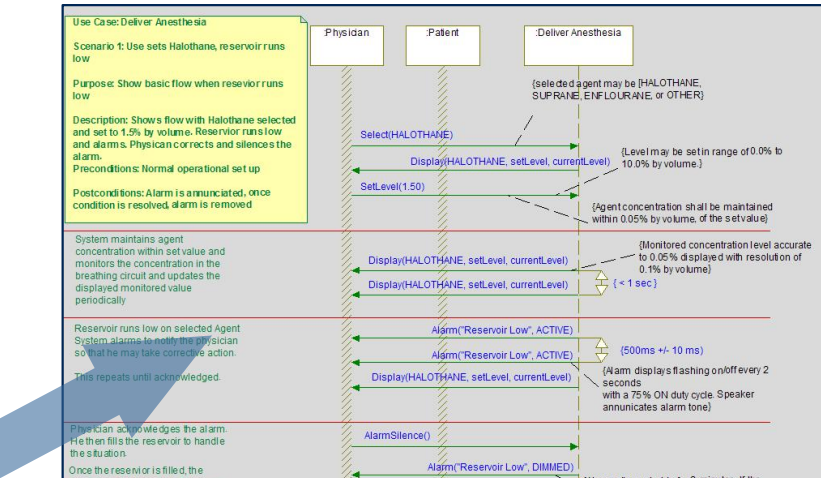
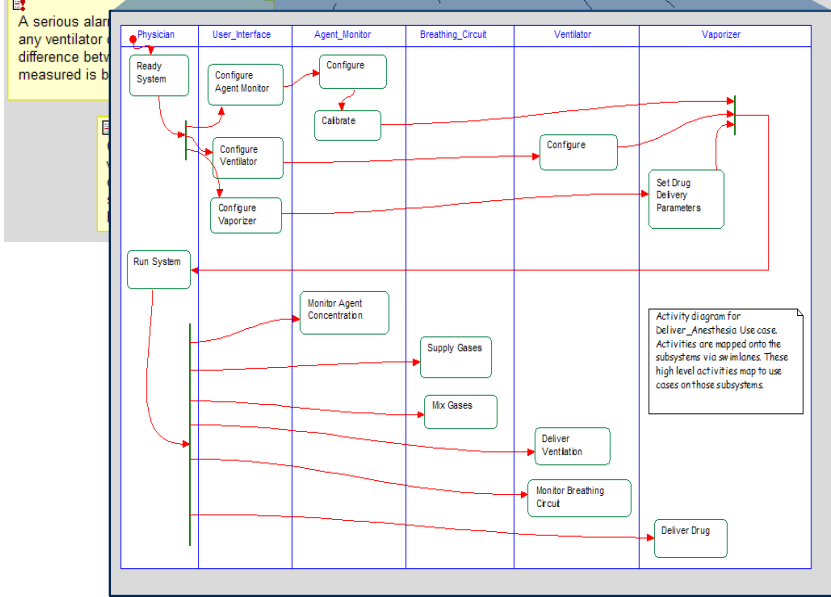
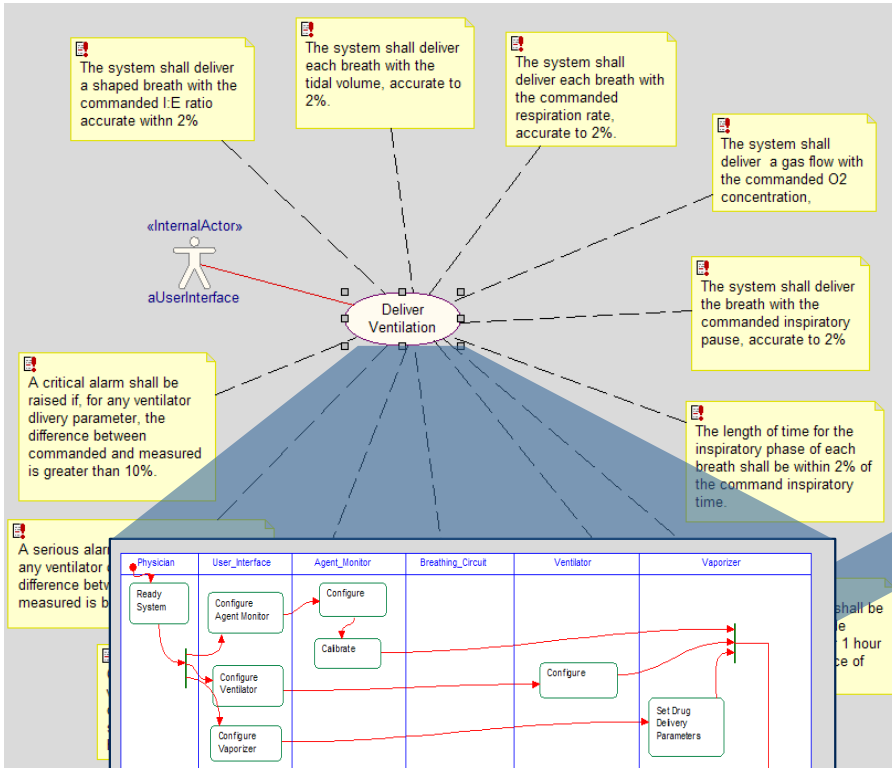
# Определение конечного автомата для описания поведения системы в Ви1



# Моделирование внешних систем

- На данном этапе для внешних систем могут также создаваться модели
- Полученные модели могут использоваться при симуляции

# Визуализация и детализация требований

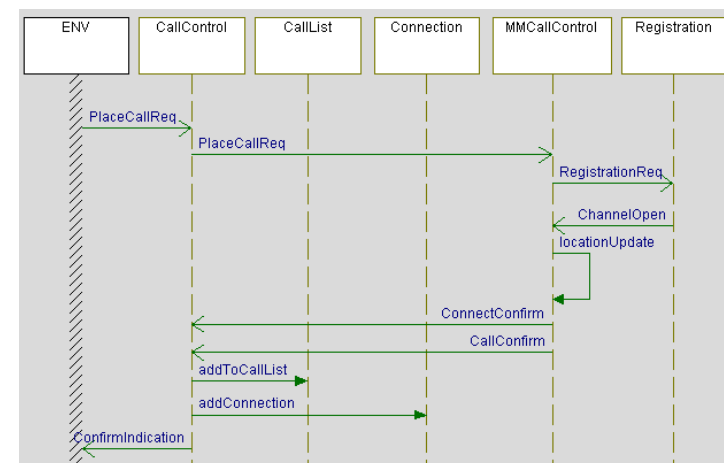


# Симуляция модели для оценки соответствия требованиям

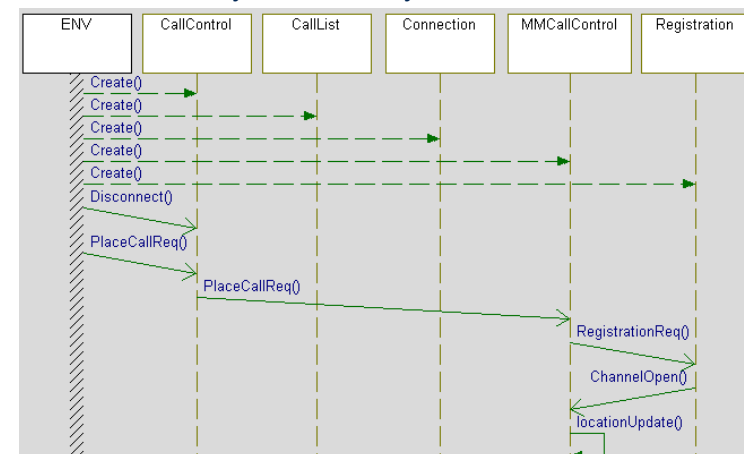
- Используйте формальные сценарии для оценки проекта на соответствие требованиям
- Симуляция модели для проверки правильности модели
  - Устранение ошибок
  - Уменьшение дорогих переработок
  - Увеличение качества
- Прототипирование пользовательских интерфейсов
  - Очень помогает для обзора проекта и обмена информацией



Ожидаемые сценарии

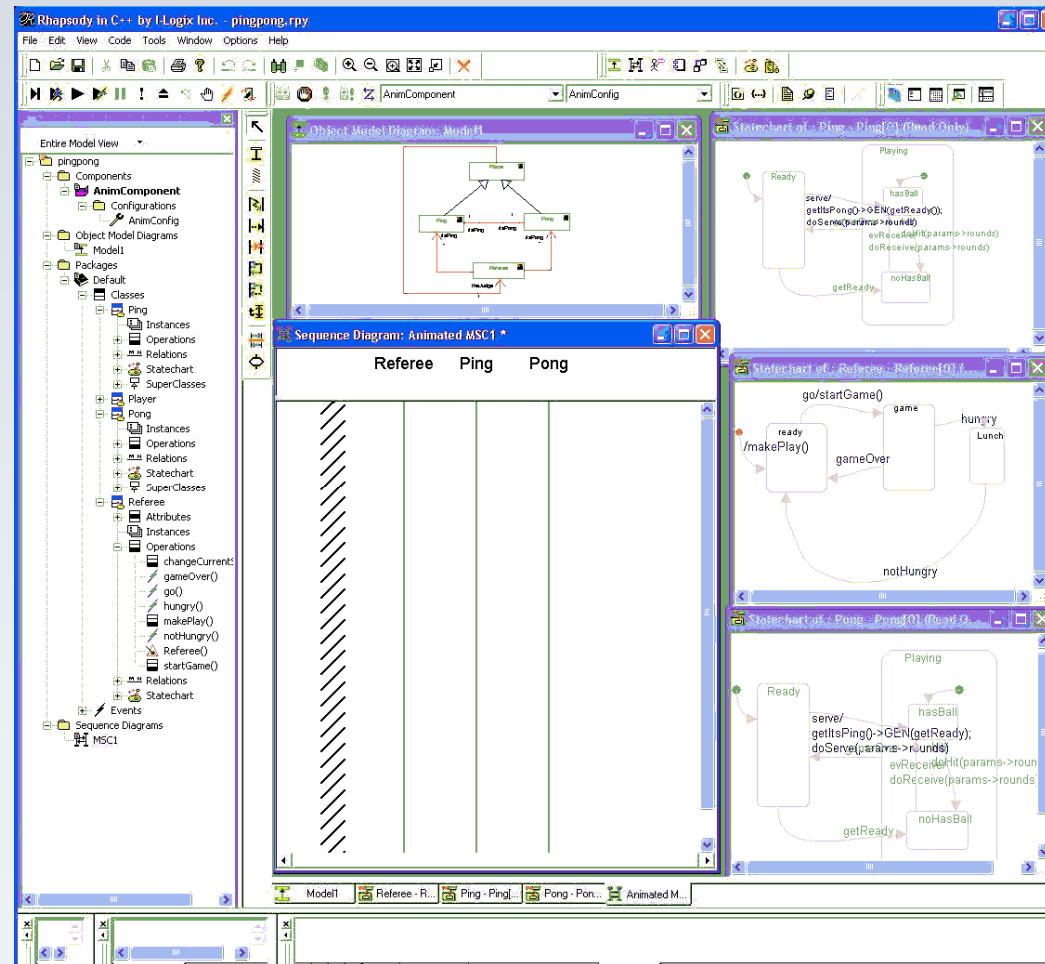


Результаты симуляции



# Симуляция, Исполнение & Анимация

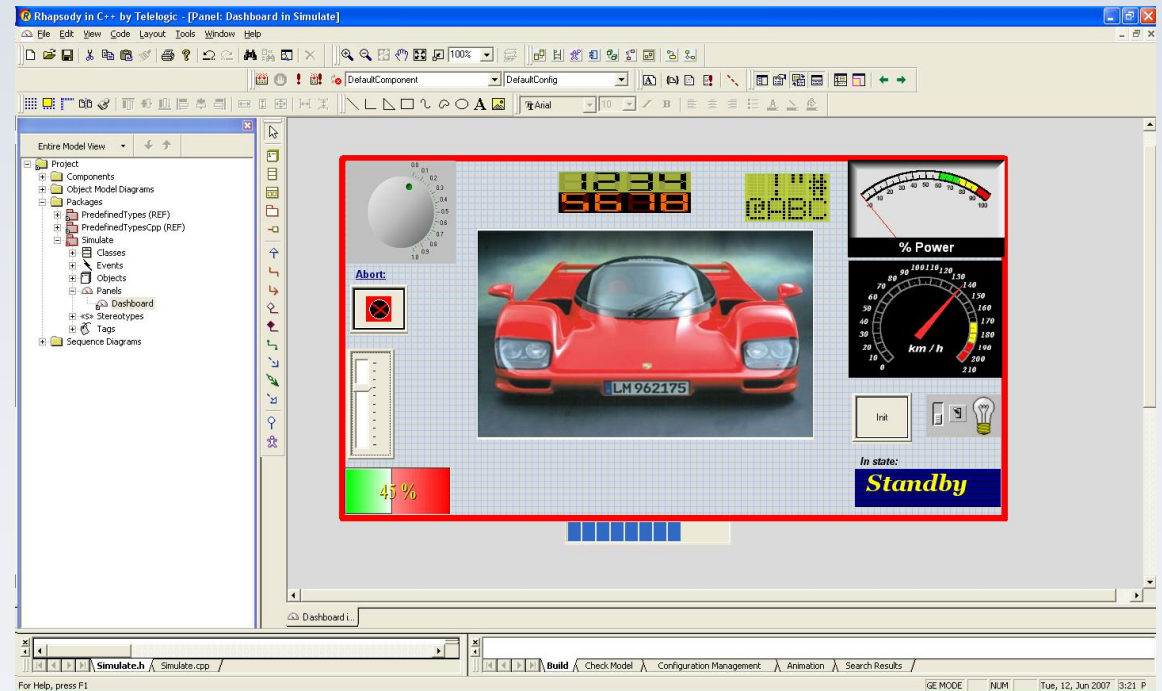
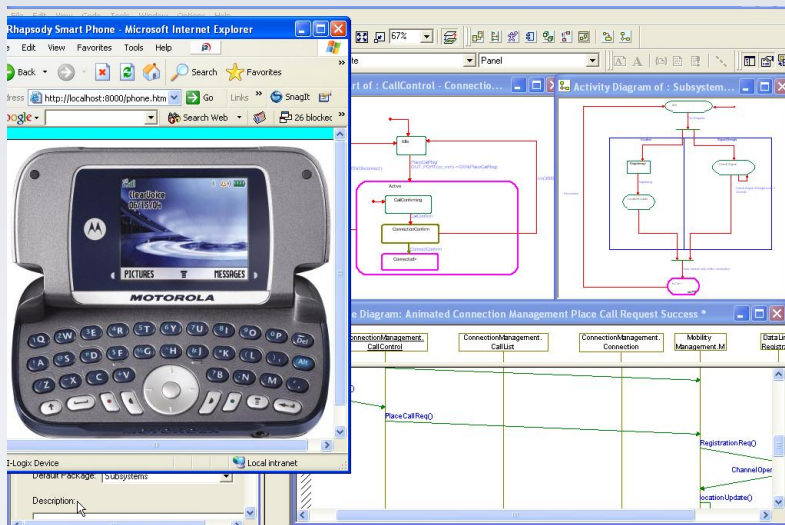
- Симуляция для проверки правильности модели
  - Лучший способ избежать ошибок и как следствие снизить их стоимость





# Прототипирование графических интерфейсов

- Создание прототипов пользовательских интерфейсов для эффективного взаимодействия с заказчиком и внутри команды
- Изменяйте, отслеживайте и анализируйте значения данных в процессе симуляции чтобы убедиться в правильности поведения системы уже на ранних этапах работ



# Генерация документации

2. Package : **Package : CORBAMobilePhonePkg**

Package specific to the MobilePhone containing a proxy for communicating to the Headset via CORBA.

**Object Model Diagram : CORBA Headset CORBA Proxy Overview**

Diagram showing the CORBA Headset Proxy

```
classDiagram
    class ICORBAMobilePhone {
        <<CORBAInterface>>
    }
    class ICORBAHeadset {
        <<CORBAInterface>>
    }
    class HeadsetCORBAProxy {
        blue
    }
    ICORBAMobilePhone <|-- HeadsetCORBAProxy
    HeadsetCORBAProxy --> IHeadsetMobilePhone
    HeadsetCORBAProxy --> IMobilePhoneHeadset
    HeadsetCORBAProxy ..> ICORBAHeadset : itsICORBAHeadset
```

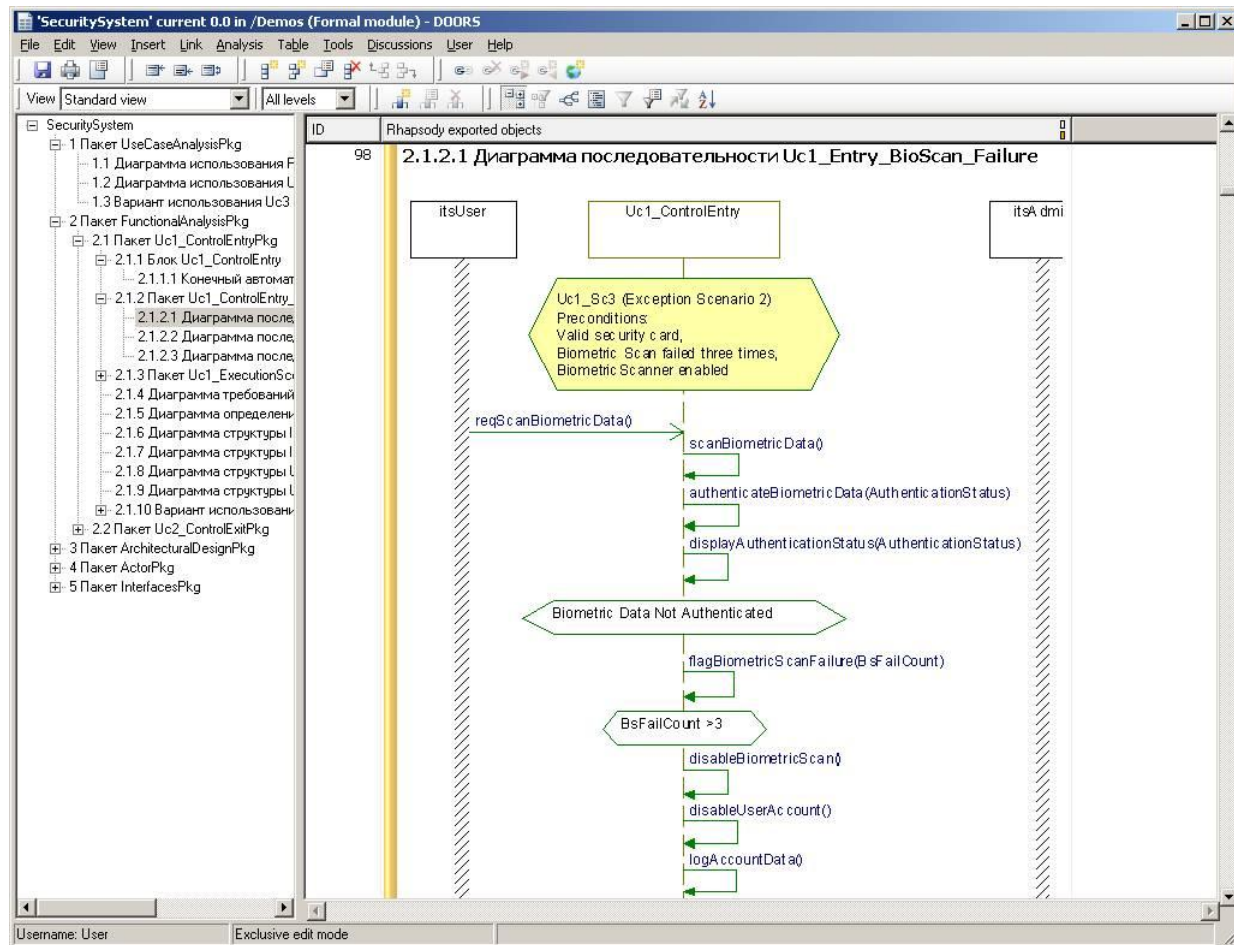
Document generated using "RhapsodyPackageOrientedProjectReport.tpl" Template

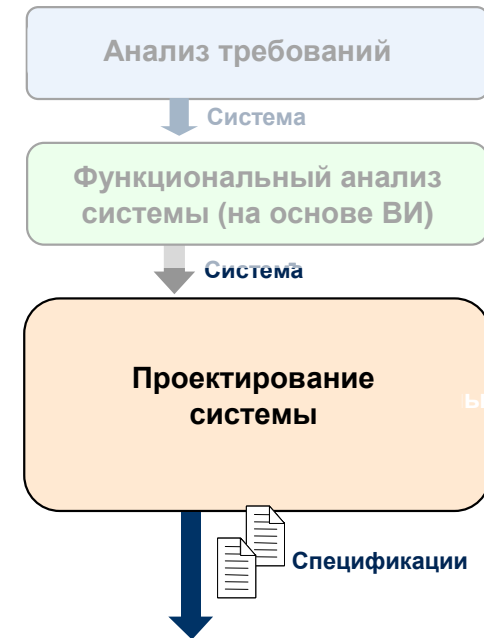
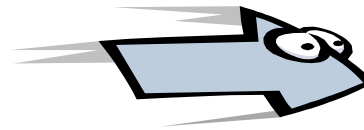
# Настраиваемая автоматическая генерация документации

- Генерация документации на основе шаблонов документов
  - Большой выбор стандартных шаблонов
  - Визуальная разработка новых шаблонов
- Генерация документации в форматах HTML, PowerPoint, Word, Rich Text Format
  - Совместима с большинством текстовыми редакторами через RTF
  - Создает ссылки в документе для быстрой навигации
- Разработка шаблонов для генерации документации в соответствии со стандартами

# Экспорт модели Rhapsody в DOORS

- Элементы и связи между требованиями и моделью могут быть экспортированы в DOORS
- Позволяет осуществлять трассировку между требованиями и элементами модели средствами DOORS





---

# Проектирование системы

---

# Разработка систем на основе моделей (*Harmony-SE*)

## Проектирование системы

Проектирование  
системы

- На этапе проектирования системы производится
  - Декомпозиции системы на подсистемы
  - Привязка операций уровня системы к подсистемам
  - Определение портов и интерфейсов для подсистем
  - Определение поведения подсистем на основе состояний
- Полученная модель верифицируется путем исполнения



# Разработка на основе моделей способствует получению предсказуемых результатов

“Разработка на основе моделей используется для более точного анализа требований, определения проектных спецификаций, проверки концепций системы на основе симуляции, и автоматической генерации кода для целевой платформы.”

“Наиболее важным аспектом применения является близость конечных результатов к ожидаемым в начале.”



## Улучшения в проектах

Срок до начала поставок	19%
Число закрытых проектов	49%
Число затянувшихся проектов	46%
Задержка в затянувшихся проектах	22%

Embedded Market Forecasters, Gaining a Competitive Design Advantage in the New Telecom/Datacom Marketplace, Dr. Jerry Krasner, June 2008

# Устранение барьеров

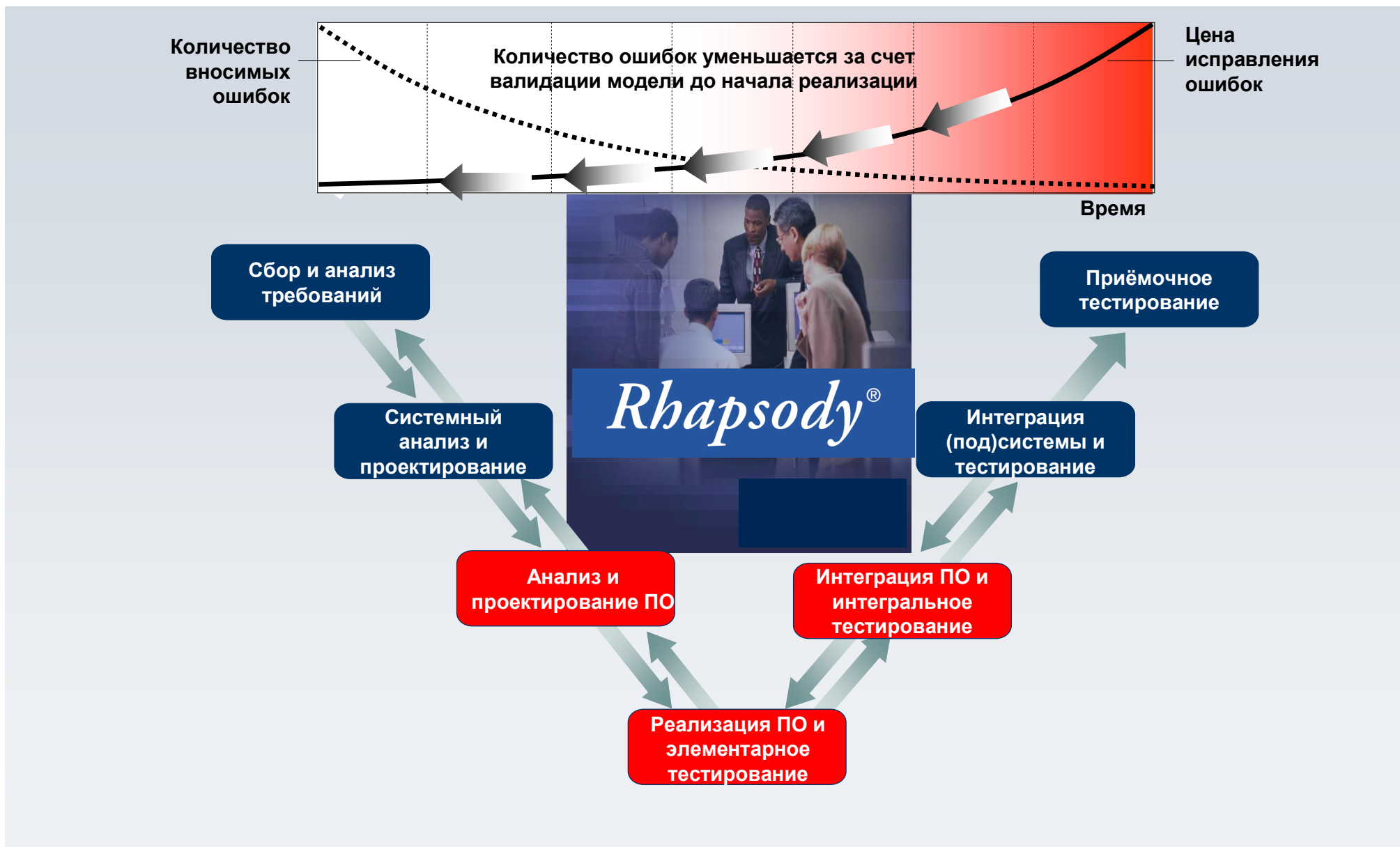




# Ранние обратные связи



# Уменьшение стоимости разработки



# IBM Rational Rhapsody

## Среда разработки на основе визуального моделирования



для разработки  
сложных  
систем



# Спасибо за внимание!

Дмитрий Рыжов

e-mail: [d.ryzhov@swd.ru](mailto:d.ryzhov@swd.ru)

<http://www.swd.ru/>



196135, г. Санкт-Петербург,  
пр. Юрия Гагарина 23  
тел.: (812) 309-2936

115553, г. Москва,  
пр. Андропова 22/30  
тел.: (495) 651-6136