

Моделеориентированное выявление требований

Ян Александер

www.scenarioplus.org.uk

RuSEC

Москва, сентябрь 2010

Почему?

Это просто –
нужно только
записать свои
требования

На самом деле, чтобы
сделать их **точными и**
проверяемыми
придется затратить
немного усилий

Мы не беспокоимся о
требованиях, мы **просто**
фиксируем сценарии
использования

Может быть, это только
кажется простым, но
на самом деле не
просто

Мы не считаем сценарии
использования
достаточными. Нам люди
сообщают решения, а мы
восстанавливаем их
фактические цели.

Моделеориентированное выявление требований

- Сценарии
- Контекст
- Цели
- Заинтересованные лица
- Развилки
- Обоснование
- Метамодель
- Инструменты
- Размышления

Небольшой пример



- **Торговый автомат**
 - как на картинке ...
 - но *не делая заранее предположений* о его устройстве или интерфейсе пользователя...

Почему не просто сценарии*?

Давайте попробуем «**Купить лимонад**»:

- Путешественник хочет купить напиток на вокзале.
- Путешественник видит торговый автомат.
- ТА показывает перечень продаваемых напитков.
- Покупатель нажимает кнопку (№5), выбирая лимонад.
- ТА показывает стоимость (€20) лимонада.
- Покупатель бросает в прорезь монеты на указанную сумму.
- ТА проверяет и считает монеты.
- ТА выдает товар - банку лимонада.

Итак, ясны ли требования ?

* Сценарии использования в универсальном языке моделирования UML – это способ организации набора Сценариев с соответствующей им Целью, триггером, субъектами действия (актерами) и гарантиями

Может быть, просто описание потребностей?

«Купить лимонад»:

- Путешественник хочет купить напиток на вокзале.
- Путешественник видит торговый автомат.
- ТА показывает перечень продаваемых напитков.
- Покупатель нажимает кнопку (№5), выбирая лимонад.
- ТА показывает стоимость (€20) лимонада.
- Покупатель бросает в прорезь монеты на указанную сумму.
- ТА проверяет и считает монеты.
- ТА выдает товар - банку лимонада.



Почему не показать ассортимент непосредственно через стекло?

Почему не бросать монеты сначала, показывая текущую сумму?

Почему бы не принимать банковские карты?

Почему бы не работать по предоплаченным картам? *

Почему бы не воспринимать голосовые команды?

Почему бы не позволить покупателю воспользоваться бесконтактной картой и открывать любую дверь, чтобы взять/заплатить за товар?

....и т.д.

* Таким, как лондонская транспортная карта "Oyster"

Что дает сценарий?

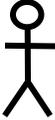
«Купить лимонад»:

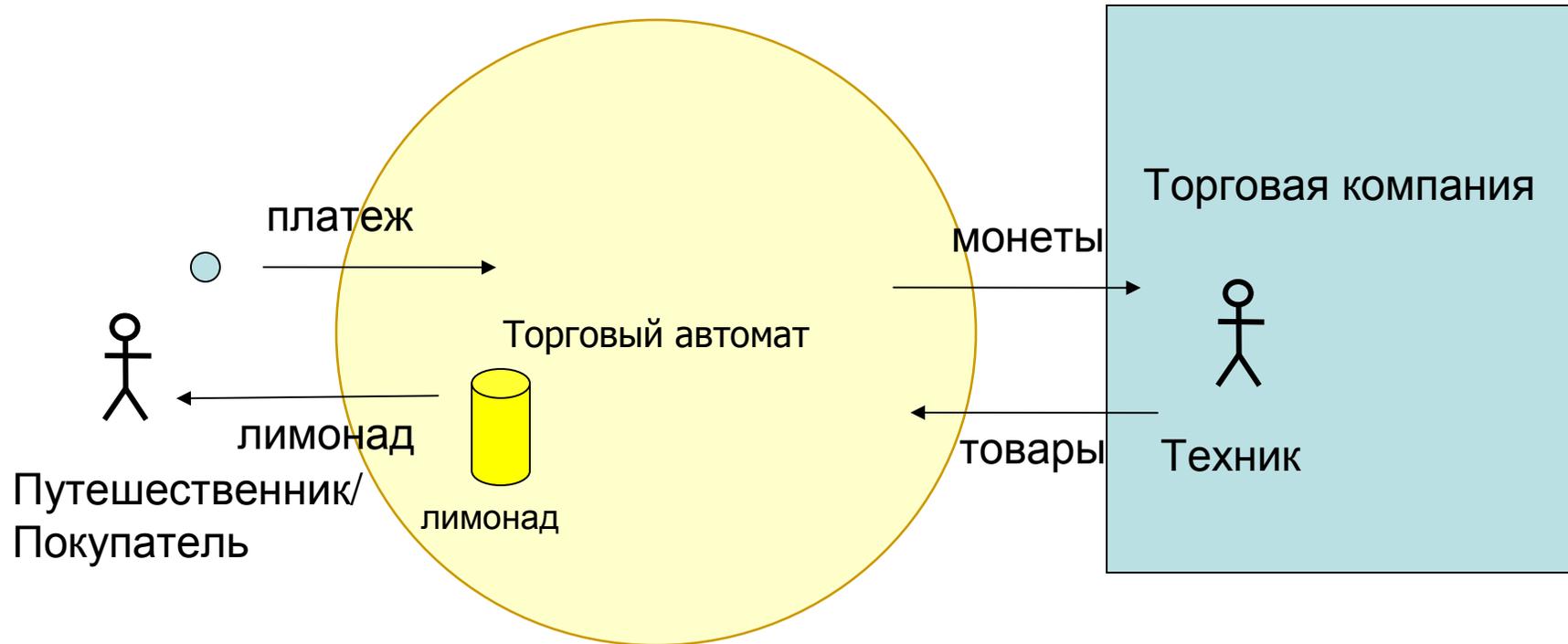
- Путешественник хочет купить напиток на вокзале.
- Путешественник видит торговый автомат.
- ТА показывает перечень продаваемых напитков.
- Покупатель нажимает кнопку (№5), выбирая лимонад.
- ТА показывает стоимость (€20) лимонада.
- Покупатель бросает в прорезь монеты на указанную сумму.
- ТА проверяет и считает монеты.
- ТА выдает товар - банку лимонада.

- Сценарий *достоверно описывает* **РЕЗУЛЬТАТ** спецификации / процесса принятия конструкторских решений
- В ответ на *[незаданные?]* **нужды** (путешественник, торговец)
- ∴ Сценарии применяются на **ПОЗДНЕЙ СТАДИИ** определения требований.

Что предшествует сценарию?

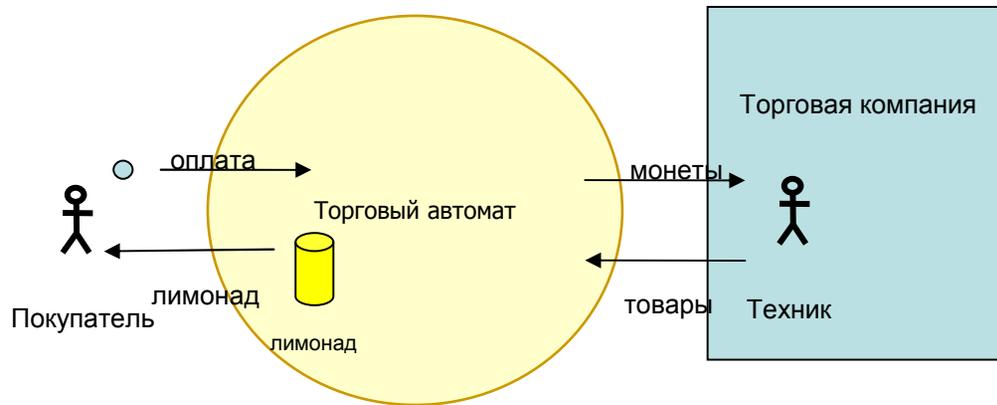
- Контекстная модель?

Владелец  (Большая машина)



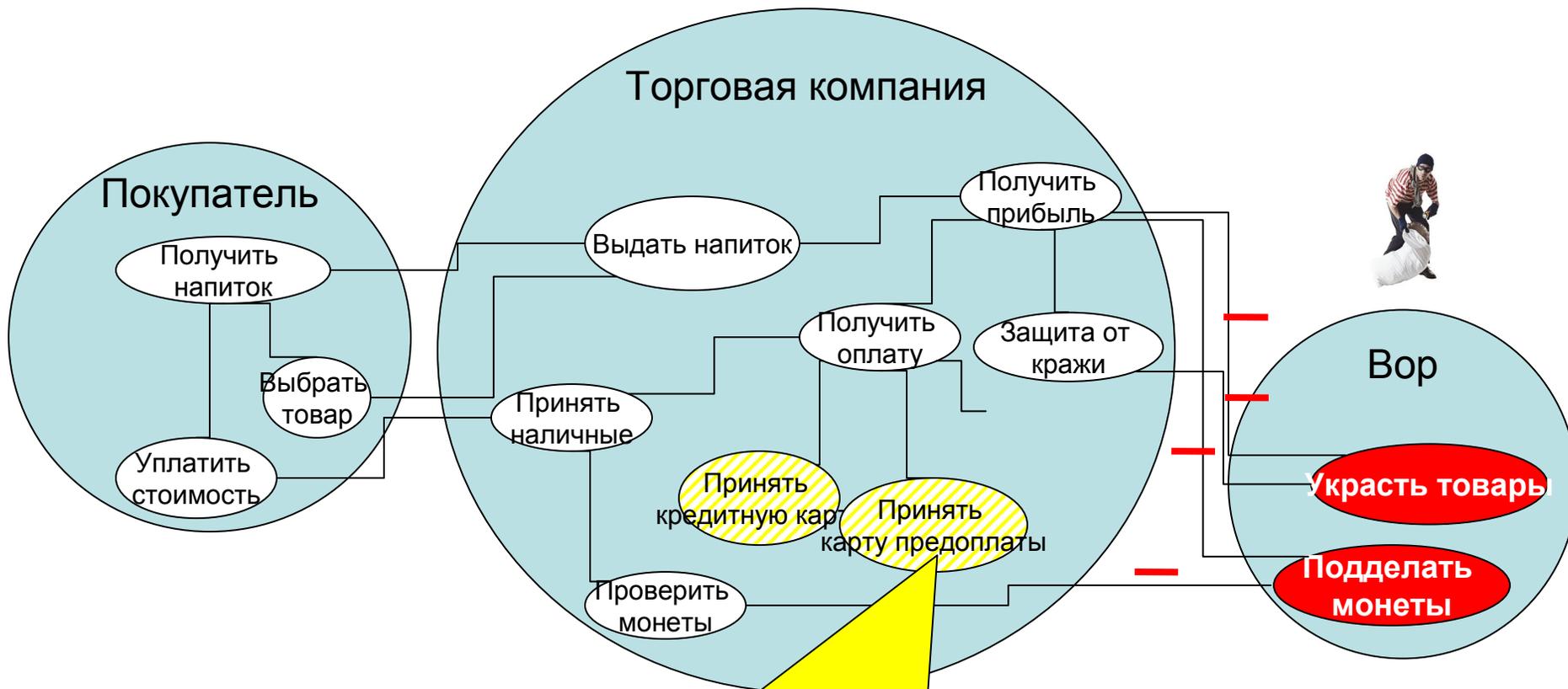
Оптовый продавец

Что дает контекстная модель?



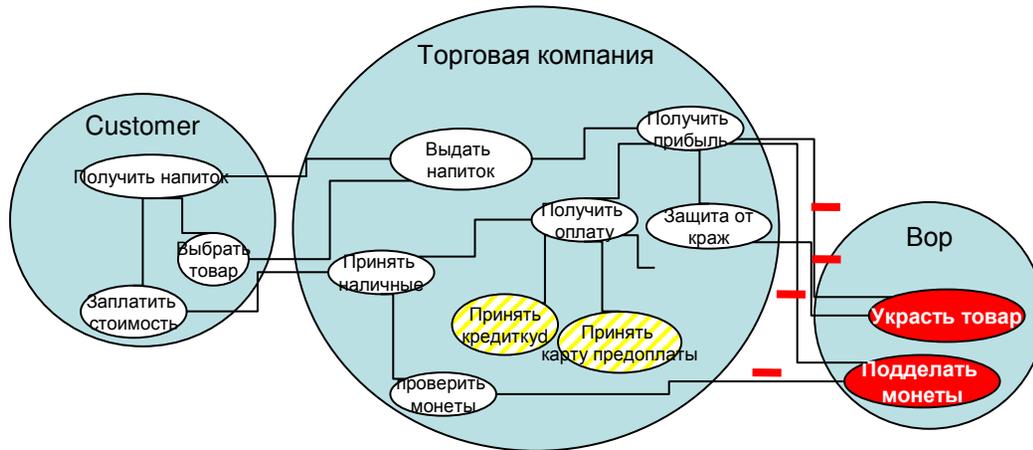
- Весьма полезна для отражения *будущей ситуации*
- Все еще отражает *множество уже принятых решений*
 - Например, торговля автоматизирована, принимаются монеты
- По крайней мере, не предписывает «сценарий» в виде последовательности действий
 - может быть реализован вариант *оплата после просмотра* или *наоборот*

Цели стоят за решениями?



Разбиение на мелкие (низко-уровневые) цели сразу навязывает конструктивные решения, т.е мы переходим от настоящих целей заинтересованных сторон к вариантам решений

О чем говорят цели?

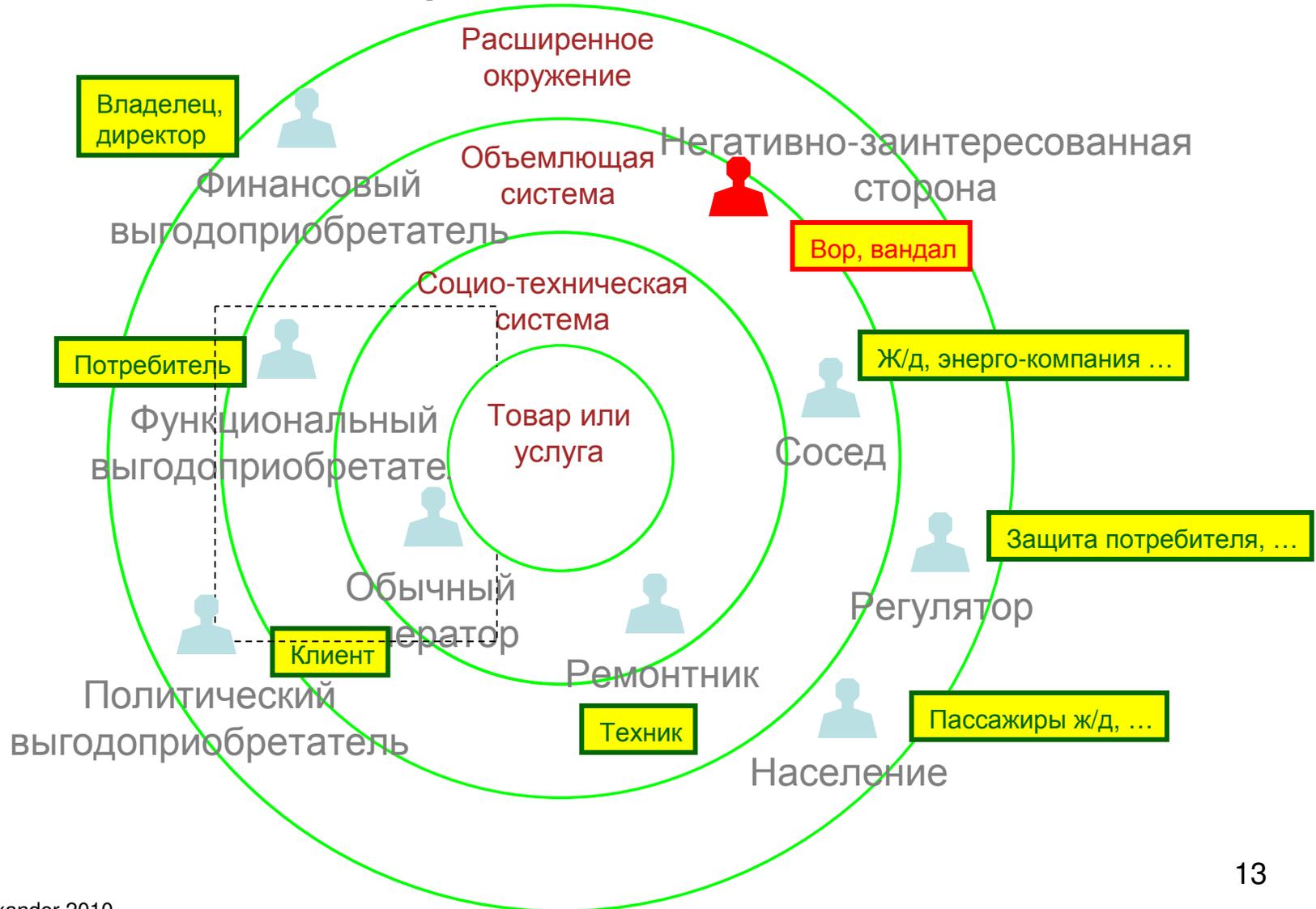


- Хорошо помогают понять, кто чего хочет
- Могут объяснять *угрозы/препятствия* (напр., воровство)
- Могут показывать *корректирующие меры*
- Могут показывать *альтернативы* (возможные требования, варианты конструктивных решений)
- ... но сначала необходим *Анализ заинтересованных сторон!*

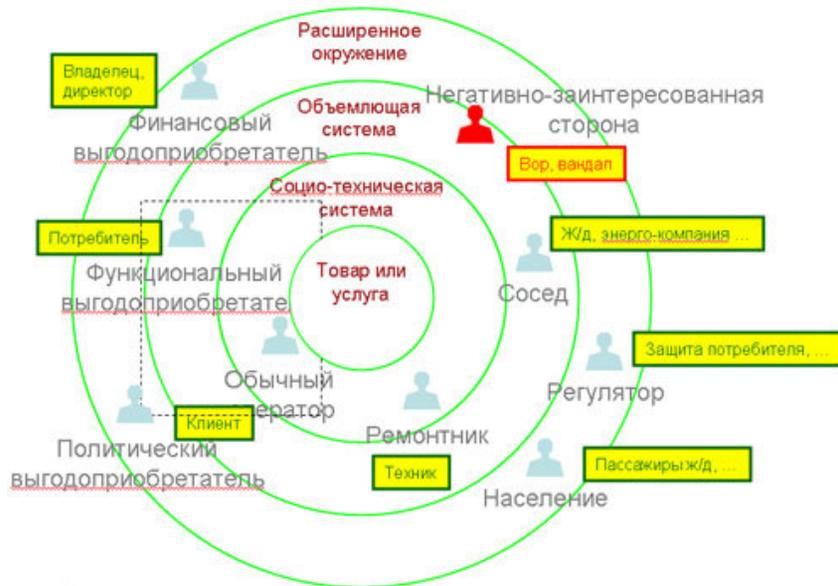
Места в «луковичной» модели заинтересованных сторон



Роли в «луковичной» модели заинтересованных лиц



Значит, заинтересованные стороны в первую очередь?



- Большое улучшение
 - выявлены некоторые роли, *не рассматривавшиеся ранее*
 - будут выявлены многочисленные пропущенные требования

- Но даже тут - *предположения* о конструкции
 - это предмет, нуждающийся в обслуживании, возможна умышленная порча
 - (мы уверены, что это не может быть просто человек с лотком?)

Разработка вариантов, развилки

Вариант	+	-
Выбрать, потом оплатить (1 предмет)	Проще в изготовлении и легче при эксплуатации	Продает только по 1 предмету Кодовые номера могут приводить к ошибкам
Оплатить, затем выбрать	Стимулирует покупку нескольких предметов	Медленнее в использовании? Сложнее, дороже производство
Давать сдачу	Более честно и гибко по отношению к клиенту	Сложнее, дороже; больше поживы для вора

Развилки

На основе *Целей*

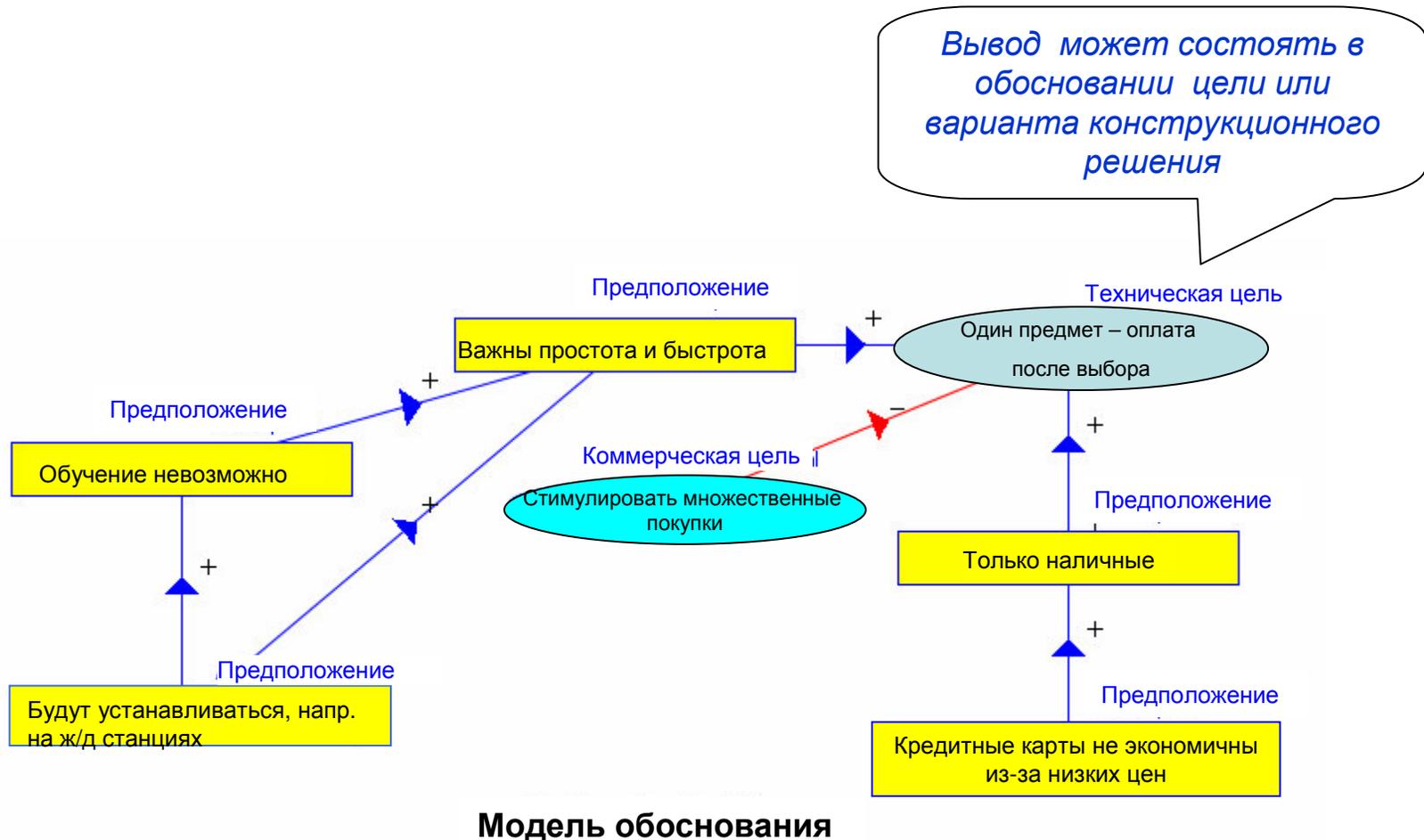
Критерии						
Варианты		C_1	C_2	C_3	C_4	C_5 ...
	O_1	Преимущества, Цены				
	O_2					
	O_3					
	...					

Для
конструкции

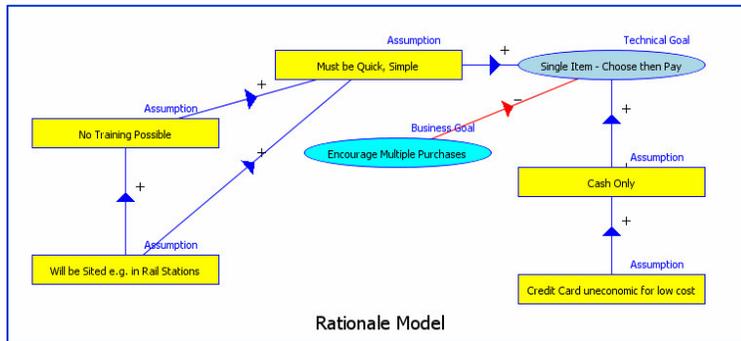
Видимо, это тоже
*желаемые свойства и
результаты*
(мета-требования)?

Всегда *многомерное
пространство*
Можно использовать многомерные
инструменты, такие, как *Principal
Components Analysis (PCA)*

Результат развилки: Обоснование



Обоснуем обоснование



- Демонстрирует надлежащую процедуру оценки
- Снижает риск
- Предотвращает повторную работу по восстановлению соображений, лежащих в основе требований и конструктивных решений
- Даёт возможность повторного использования для создания вариантов изделия или семейства продуктов
- Объясняет ключевые решения по проекту новым участникам

Из чего состоит анализ развилок?

- На входе:
 - Цели
 - Приоритеты заинтересованных сторон (*еще одна модель...*)
- На выходе:
 - Выбранный вариант конструкционного решения (решений), архитектура
 - Измерения (*да, снова моделирование*)
 - Обоснования выбора вариантов, требований
 - Требования
 - Приоритеты проекта

Мы даже не знаем, какие цели превратятся в требования, пока мы не закончили анализ развилки

Цели, варианты, требования

- Цели:
 - то, чего хотят заинтересованные стороны; кандидаты в требования
 - не всегда реалистичны, доступны по цене и поддаются проверке
 - могут противоречить другим целям
 - могут оказаться возможными только для некоторых вариантов
- Варианты:
 - альтернативные подходы к решениям на любом уровне
 - *функции программного обеспечения, конкурирующие за время/бюджет разработки*
 - *свойства аппаратного обеспечения, конкурирующие за энергоснабжение, массу, и т.д.*
 - *альтернативные алгоритмы достижения данной цели*
 - *конкурирующие архитектуры системы*
- Критерии прохождения развилки на основе целей:
 - размерности, в которых оцениваются и сравниваются варианты
- Требования, после прохождения развилки:
 - согласованные, измеримые, верифицируемые; компромиссные/приоритизированные цели

Цели, варианты, требования

- Цели:
 - (путешественник): *получить напиток, быстро, удобно, недорого*
 - (компания): *делать деньги (посредством торговли)*
- Варианты (предполагая торговый автомат*):
 - только наличные, монеты или банкноты, со сдачей или без
 - банковские карты
 - предоплаченные карты
- Критерии для развилки, на основе целей:
 - прибыль/издержки эксплуатации
 - прибыль/ издержки сооружения
 - привлекательность для клиента
 - риск кражи, мошенничества (минимизация)
- Требования, после развилки:
 - ТА должен принимать монеты
 - ТА должен давать сдачу монетами

Могут ли это быть так же цели? Да, когда цели декомпозируются, они естественно углубляются в особенности решений

*Цели могут быть реальными и важными, но **бесполезными** как критерии. Если варианты только напоить путника и обеспечить платежи, эти критерии не помогут найти победителя!*

** на уровне системы, если не предполагать ТА, тогда варианты включают:
ТА на платформе, киоск на платформе, уличный торговец (с лотком/тележкой), тележка в поезде, вагон-ресторан, ...*

Предполагаемый процесс Моделирования ориентированного выявления требований (MBRD)

0. Выкиньте учебники по инженерии требований «с нуля»

1. Отмоделировать **цели** модернизации

2. Определить **контекст** существующих и новых систем

3. Определить интерфейсы, которые **нельзя менять**

4. Исследовать **варианты** допустимого изменения рамок

5. Выявить **заинтересованные стороны**, особенно там, где вносятся изменения

6. Выявить **цели, противоречия, исходные приоритеты** заинтересованных сторон

7. **Сравнить** альтернативные решения с учетом целей

8. Документировать обоснования принятых решений

9. Установить **выходные приоритеты** проекта



На существующей базе,
не классическое УТ



повторить цикл

Матрица MBRD

<p><i>Элементы требований</i></p> <p>Дано</p> <p>1. Видение</p> <p><i>Контекст выявления</i></p>	1. Заинтересованные лица	3. Цели	4. Контекст	5. Сценарии	6. Свойства и ограничения	7. Обоснование	8. Определения	9. Измерения	10. Приоритеты
A. От отдельных лиц	<p>Элементы, подлежащие выявлению</p>								
B. От групп									
C. От объектов									
D. Развилки									

В ЭТИХ КОНТЕКСТАХ

«Сценарии от групп»
напр., 1-дневный семинар по сценариям использования

Практики в MBRD



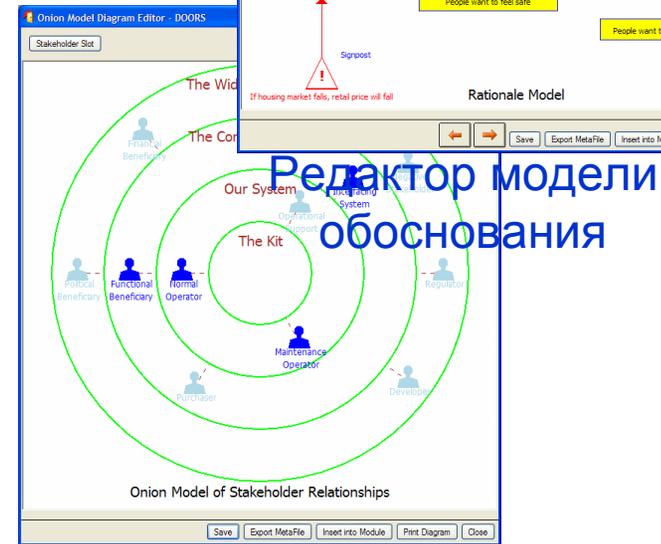
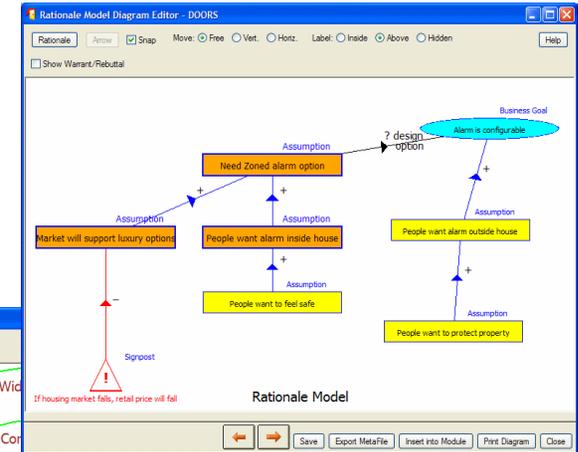
Валидация требований на основе модели

- Убедиться в том, что для всех объектов модели:
 - Цель принадлежит какой-то **Заинтересованной стороне**
 - Операционное заинтересованное лицо *играет роль* в **Сценарии**
 - Цели *приоритизирована* определенным **Приоритетом**
 - **Высокоприоритетные цели** *используются как критерии* при выборе **Развилок**
 - **Конфликты между целями** *устраняются* в процессе прохождения **Развилок**
 - **Препятствия/Угрозы** *смягчаются* **Целями**
 - **Цель удовлетворяется** **Требованием**
 - **Требование** *делается проверяемым* **Измерением**
 - **Развилка** *объясняется* **Обоснованием**
 - **<Термин>*** в **Требованиях** *определяется* в **Словаре**

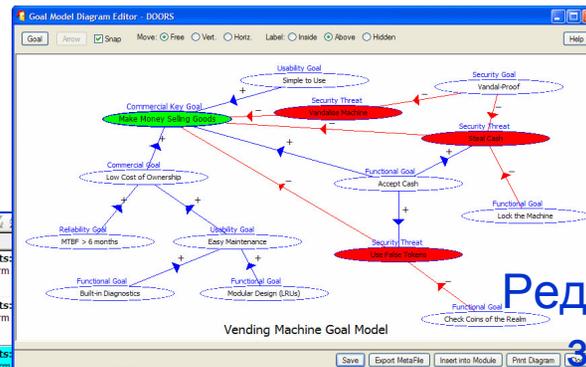
* <Термин> может быть любым **Состоянием**, **Целью**, **Операционной ролью**, **Измерением**

Поддержка инструментами

- Существующий бесплатный набор надстроек для DOORS
 - Полная трассируемость до отдельных объектов в диаграммах
 - Полная системно-инженерная история, базы
- Флюмастер и доска часто идеальны для выявления требований
- Любой графический редактор, если не нужна трассировка



Редактор модели обоснования



Редактор модели целей

ID	Definitions of Terms Used in Project	Places where dictionary terms are used
D-6	5 intrusion Apparently unauthorised entry into the Protected Household (external event)	/ Burglar Alarm/Requirements: Object Text: The household alarm
D-5	6 detect Action by Household Alarm to identify a significant external event e.g. Intrusion, Tampering	/ Burglar Alarm/Requirements: Object Text: The household alarm
D-4	7 timeout period Period of TBD 30 seconds to allow householders to enter or exit	/ Burglar Alarm/Requirements: Object BR-55: Object Text: The household alarm should notify an / Burglar Alarm/Requirements: Object BR-6: Object Text: The householder shall be able to detect
D-3	8 deactivate Action (by user) to enable the Household Alarm	/ Burglar Alarm/Requirements: Object BR-6: Object Text: The householder shall be able to detect

Редактор «луковичной» модели заинтересованных сторон

Составитель и редактор связей словарей

Почему MBRD?

- Обеспечивает максимальный учет
 - раинтересованных сторон, целей, конфликтов, препятствий, сценариев, обоснований,
- Обеспечивает понимание всех сторон *задачи*
- Позволяет увидеть как они связываются, чтобы образовать целое



Модель требований, Модель системы?



Не является ли MBRD частью обычной моделиеориентированной системной инженерии?

- **ДА, НО...**

- Только при правильном выполнении
 - *вы уже моделируете заинтересованные стороны, цели, и обоснования?*
- Многие подходы давно знакомы
 - *например, контекст, сценарии эксплуатации*
- Часто плохо трассированы
 - *особенно если цели и решения, до которых должна проводиться трассировка, не задокументированы*
- Заинтересованные стороны и их цели часто недооцениваются
 - *целые группы упущенных требований*
- MBRD также необходима в *софтверной инженерии*

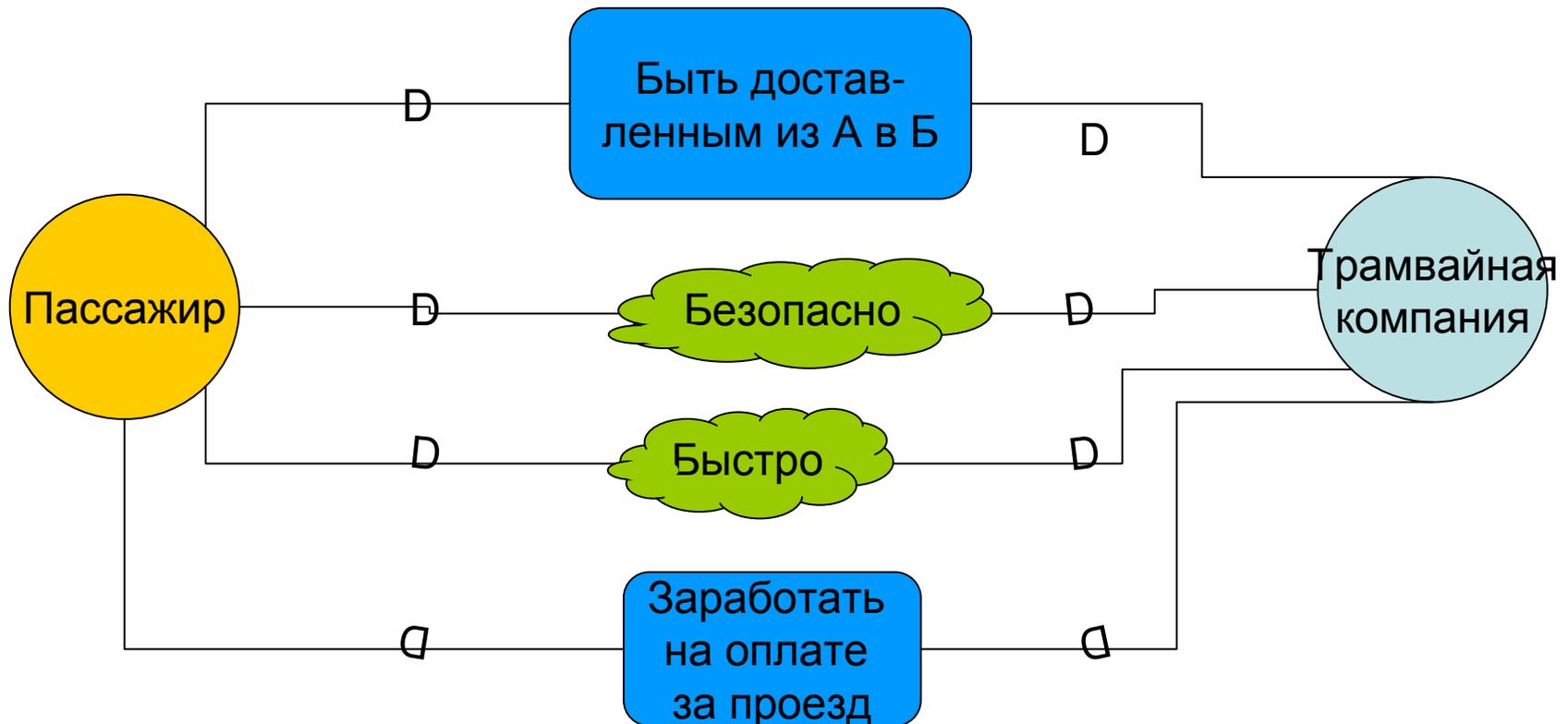
А что насчет целеориентированной инженерии требований (GORE)?

- **Целеориентированная ИТ** представляет собой подход, в котором делается акцент на Цели как ключ к требованиям

	<i>Requirement Elements</i>									
<i>Discovery Contexts</i>	2. Stakeholders	3. Goals	4. Context	5. Scenarios	6. Qualities and Constraints	7. Rationale	8. Definitions	9. Measurements	10. Priorities	
	A. From Individuals									
B. From Groups										
C. From Things										
D. Trade-Offs										

- 2 основных школы в исследованиях GORE:
 - **i*** из Торонто (Eric Yu)
 - **KAOS** из Лувена/Лёвена (Axel van Lamsweerde)
- Обеим школам ~20 лет

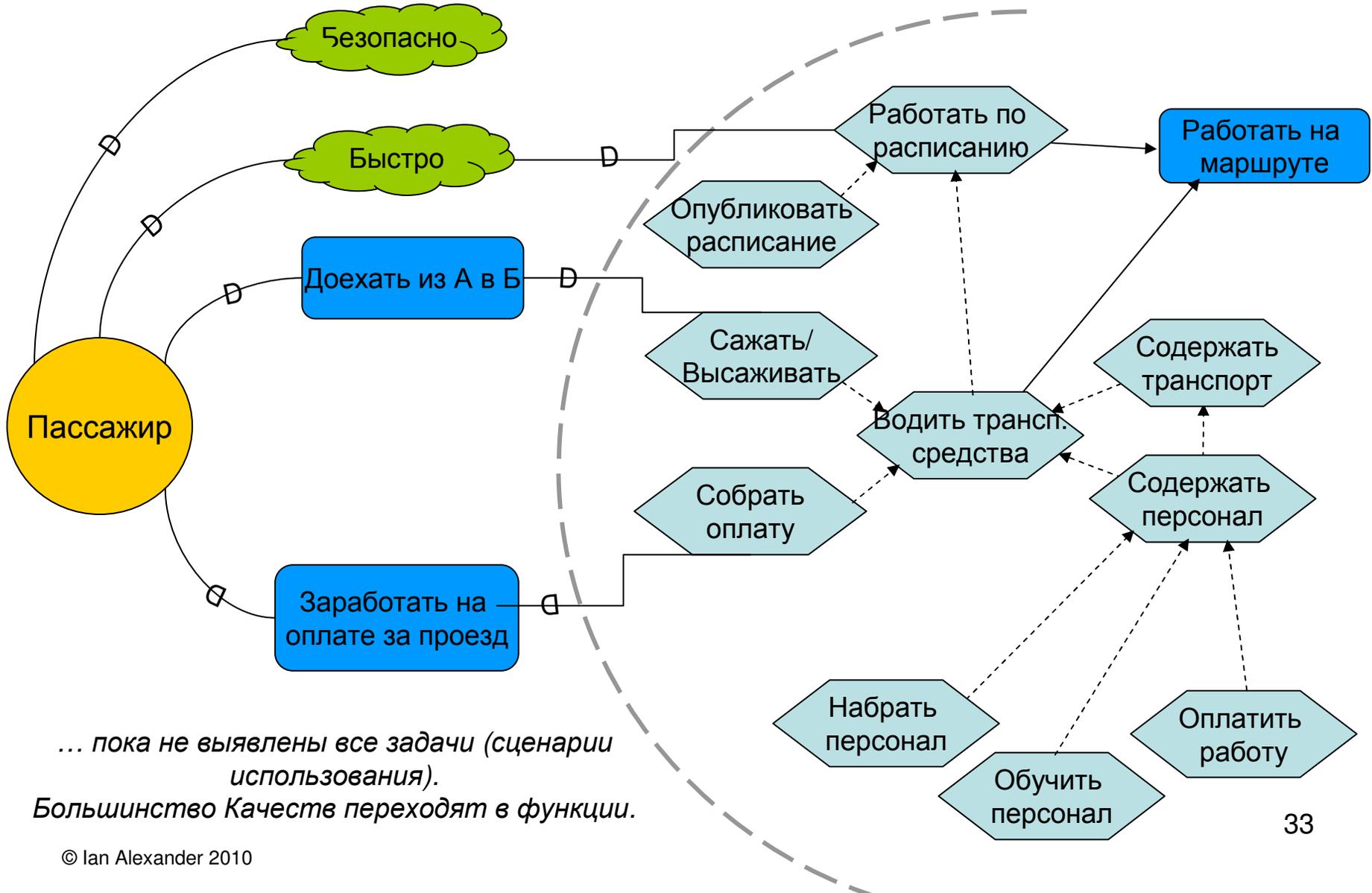
i* Стратегическая зависимость



Примечание: Цветовая кодировка, если используется, зависит от использованного ПО.

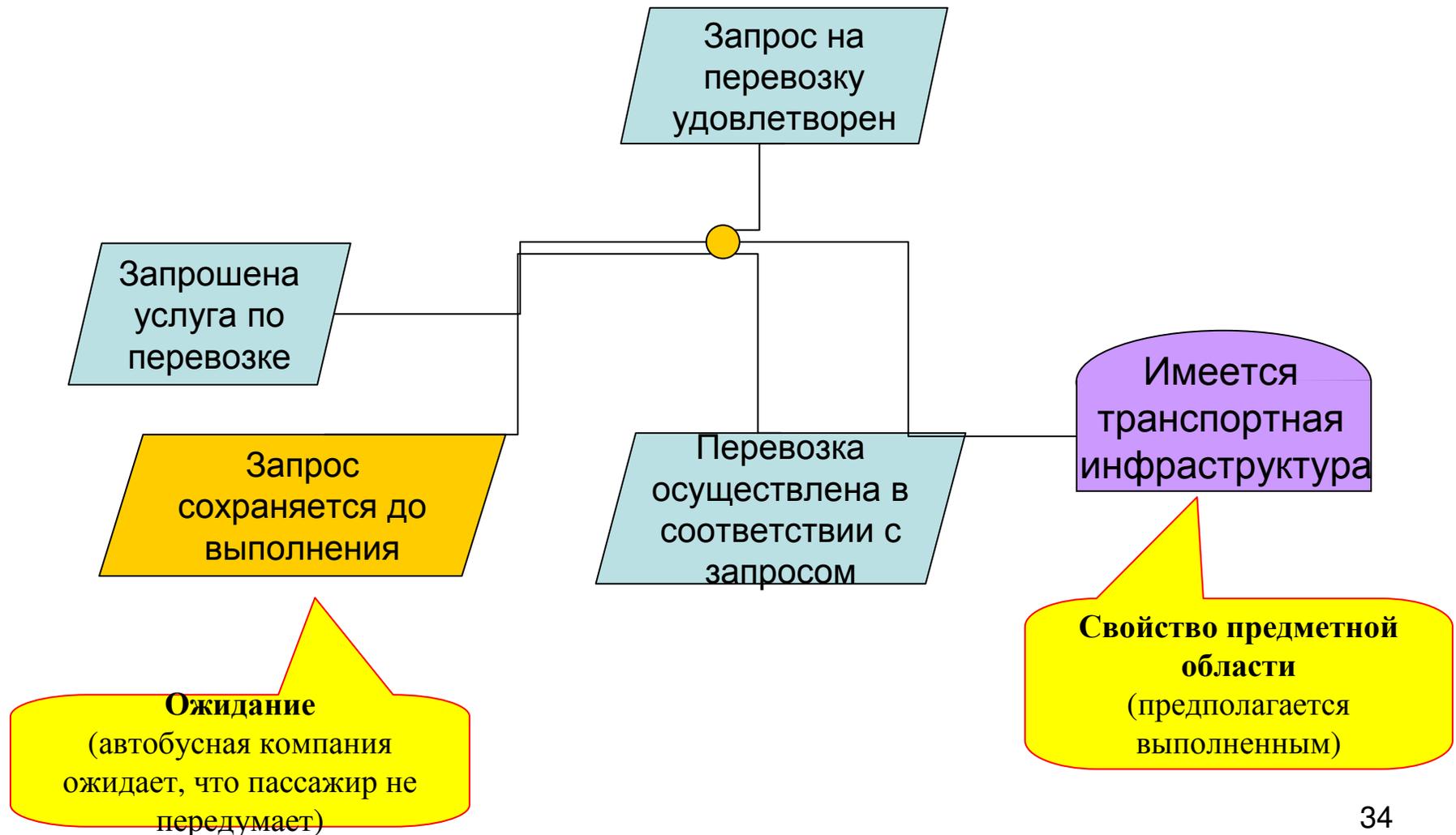
i* -диаграммы обычно печатаются черно-белыми

i* Стратегическое обоснование



... пока не выявлены все задачи (сценарии использования).
Большинство Качеств переходят в функции.

Типовая схема KAOS «Запрос на предоставление услуги»



Как дела с Целями в промышленности?

- В промышленности явное моделирование целей – редкость
- Больше внимание к целям уменьшит добавление «бантиков и рюшечек»
- Не очевидно, что i^* или KAOS найдут широкое применение
- Овал из Сценариев применения в языке UML – просто отправная точка
 - функциональная цель («сделать что-либо»)
 - реализуется набором сценариев
 - так почему бы не злоупотребить обозначением и для нефункциональных целей?!

Requirement Elements	Discovery Contexts									
	2. Заинтересованные стороны	3. Цели	4. Контекст	5. Сценарии	6. Качества и ограничения	7. Обоснование	8. Определения	9. Измерения	10. Приоритеты	
A. От отдельных лиц										
B. От групп										
C. От предметов										
D. Компромиссы										



Конкурирующие подходы?

Подход (элемент)	Как он описывает потребности	Школы, в которых он предпочитается	Недостатки исключительного использования
Анализ заинтересованных сторон	Выявляет политические, экономические, социальные, культурные давления на конструктивное решение	<i>Методология разработки ПО</i>	Неточен, не проверяем; непригоден для контрактов
Моделирование целей	Говорит, чего хотят заинтересованные стороны	<i>Метод KAOS;</i> язык моделирования целей i *	Плохо пригоден для определения динамики (сценариев), риск углубиться в конструкцию
Анализ событий	Выявляет события на интерфейсах и говорит, как реагировать на них	<i>Методы, ориентированные на события</i>	Контекст предполагается четко заданным; игнорируются тонкие системные вопросы, конфликты целей
Анализ сценариев	Показывает, как конструкция обеспечит результаты при воздействиях человека	<i>Сценарии применения в стиле Кокберна; Гибкие методы (истории пользователя)</i>	Излишний акцент на поведение продукта
Повторное использование – стандарты и шаблоны	Определяет общие потребные качества, ограничения, и процедуры их достижения/выполнения	Стандартизация, регулирование, обеспечение качества	Не учитываются функциональные и инновационные аспекты
Моделирование обоснования	Объясняет, почему нужна такая конструкция, или что нужно, чтобы сделать ее безопасной	<i>Compendium;</i> <i>GSN, CAE</i>	Недостаток учета динамики (сценарии; риск отставания выбранного решения)
Моделирование данных	Определяет правила, обеспечивающие нормальную деятельность	<i>Моделирование классов – язык моделирования UML</i>	Недостаток сквозного описания (сценариев), контекста, назначения
Измерение	Точно показывает, какие результаты должна обеспечить конструкция	Традиционная; требования типа «система должна»...	Формулировка требований должна нести всю нагрузку
Приоритеты	Показывает, какие свойства конструкции наиболее нужны	<i>Бизнес-сценарии, Анализ выгод и издержек, Инженерия стоимости</i>	Предполагает, что требования независимы и сравниваются по цене

Дополняющие подходы

<i>Элементы требований</i>	Заинтересованные стороны	Цели	Контекст, интерфейсы, охват	Сценарии	Качества и ограничения	Обоснование	Определения	Измерения	Приоритеты
<i>Контексты выявления</i>									
От отдельных лиц									
От групп									
От объектов									
Развилки									

В вашем проекте могут понадобиться любые из них, или все

Спасибо за внимание

