

Дисциплина

СПЕЦИФИКАЦИЯ, ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ

Лекция 4

СПЕЦИФИКАЦИЯ И УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ.

Преподаватель

к.т.н. Романенко Сергей Александрович

Требования – основа любого проекта

Требования определяют те потребности «заинтересованных сторон» (stakeholders) – заказчиков, пользователей (потребителей), разработчиков, которые являются для них необходимыми, а также тот функционал и свойства, которыми система должна впоследствии обладать, чтобы удовлетворить эти потребности.

Главная проблема: недостаток информации от пользователя или заказчика о функциях, неполные, некорректные требования, а также многочисленные изменения требований и спецификаций приводят к хаотическим, неупорядоченным действиям в процессе разработки программного обеспечения и неукладу проекта в целом.

Программная система это:

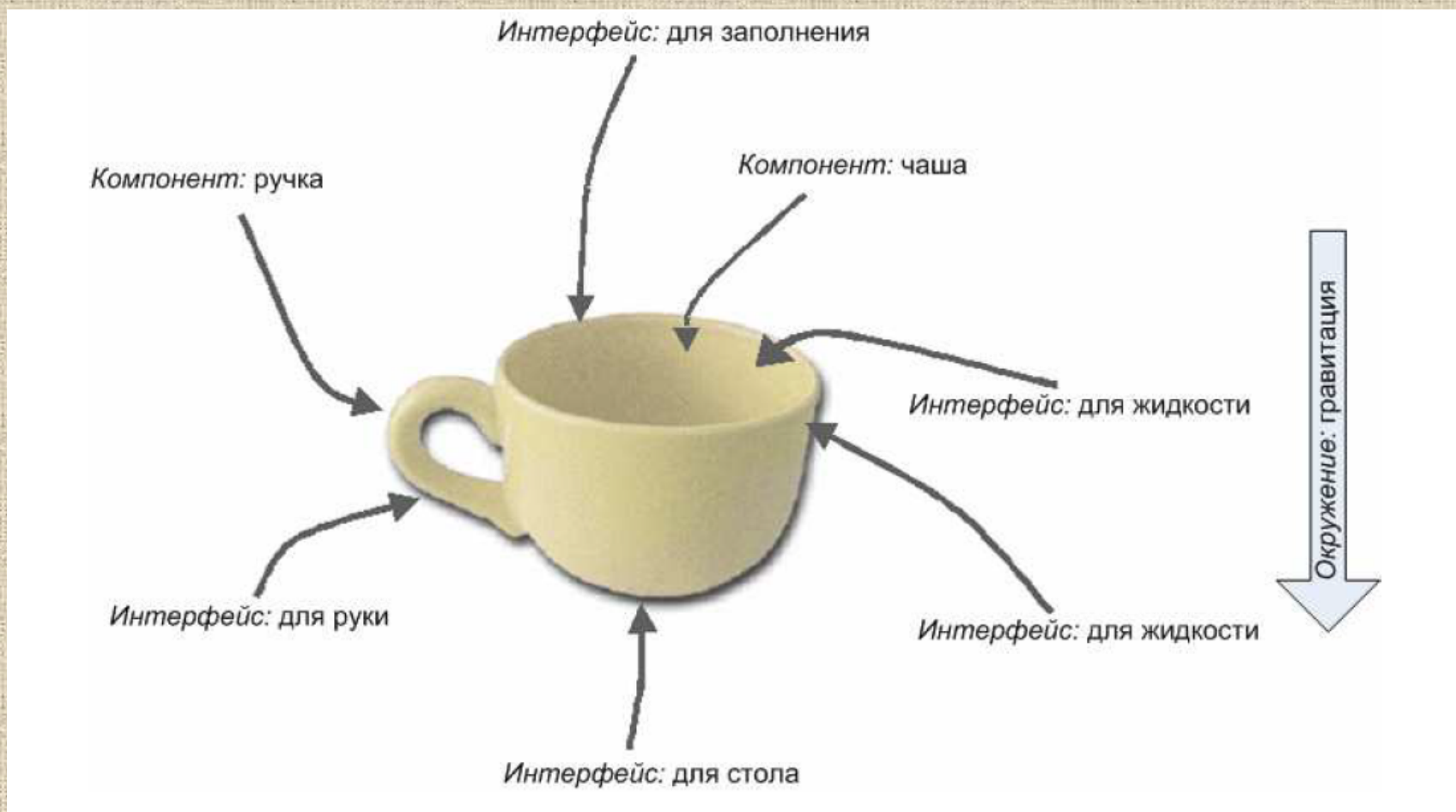
- ✓ ***набор компонентов*** – механизмов (аппаратных средств), программного обеспечения и людей, -
- ✓ которые ***согласованно взаимодействуют***
- ✓ для достижения некоторого ***заданного результата***, сформулированного в виде ***требований***

Пример. Пользовательские требования

*Система должна позволить человеку
перенести жидкость*

Требования в контексте системного проектирования

Пример простой системы



Способность системы удовлетворить сформулированную цель и назначение зависит от:

- ✓ свойств, которые появились в результате взаимодействия ее компонентов;
- ✓ соответствующих интерфейсов с внешними компонентами;
- ✓ ее корректного включения в окружение (среду);
- ✓ присутствия соответствующих внешних условий.

*В процессе разработки требований необходимо принимать во внимание **природу системы**. Нельзя рассматривать функциональные требования и системные свойства изолированно, необходим комплексный подход: **учет условий, которые привносит внешнее окружение**, накладываемые кем-то или чем-то ограничения, а также интерфейсы с окружающими системами.*

Требования и качество

Качество есть соответствие системы целям и требованиям — это обеспечение того, что удовлетворяет потребителя и в тоже время гарантирует, что нужды всех заинтересованных сторон учтены.

Уровни требований

Пользовательские требования
(stakeholder requirements)

удовлетворяются

Системными требованиями

Которые разделяются на

Требования к подсистемам

Которые реализуются как

Требования к компонентам

Требования и процесс выполнения проекта



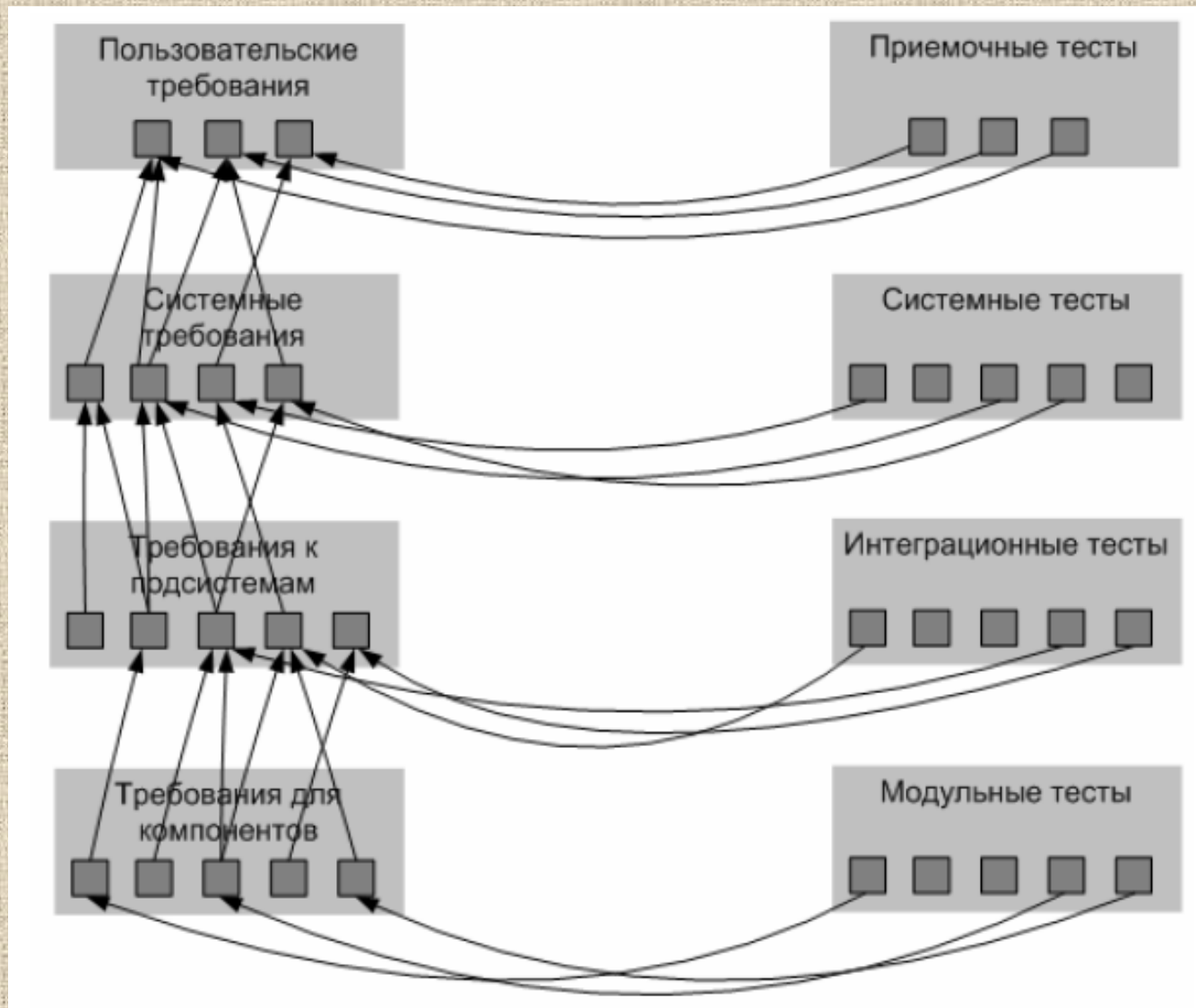
Использование требований для связи между проектами



Риски управления изменениями, связанные с взаимосвязанностью требований



Связи между требованиями

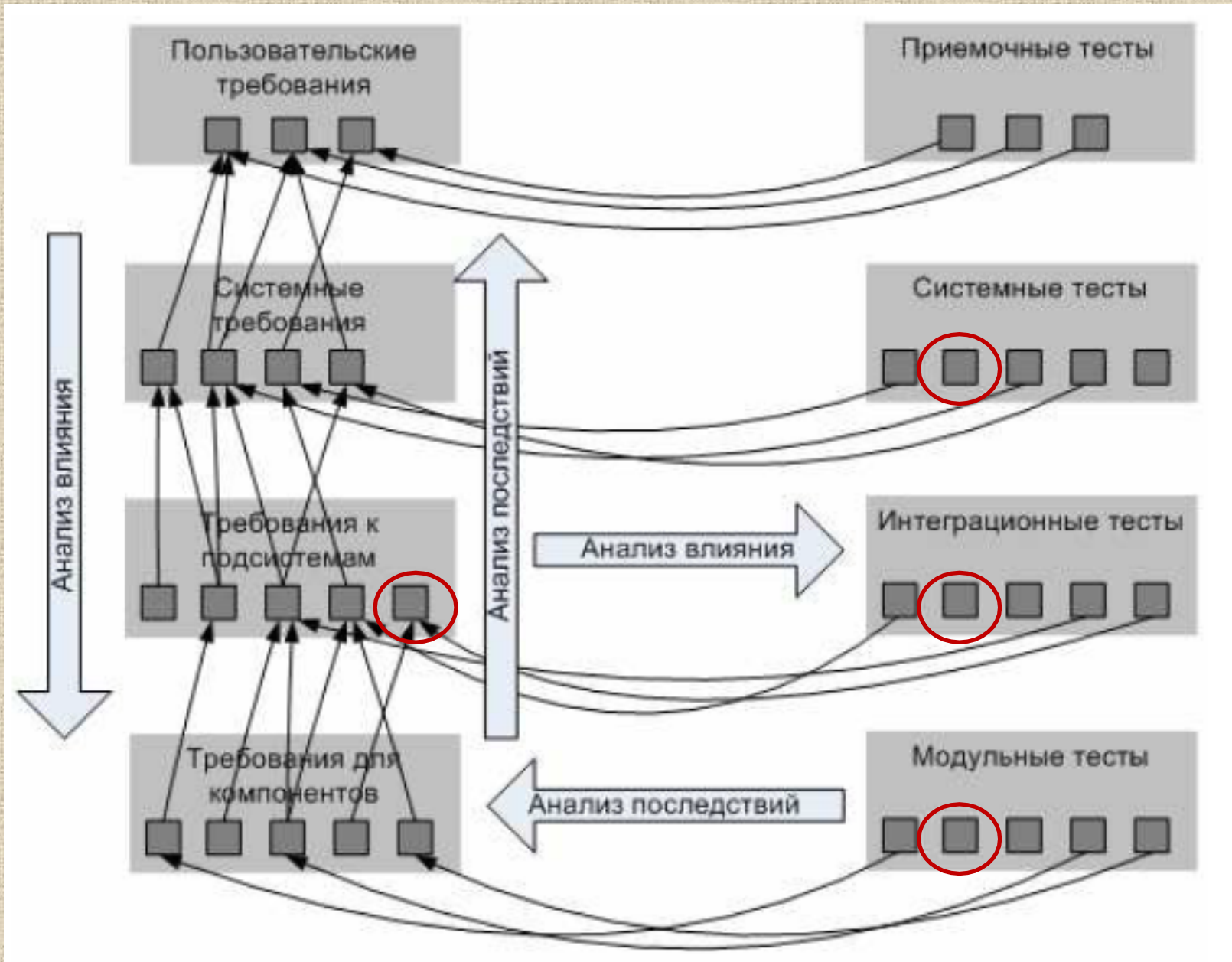


Стрелка указывает на **источник информации**

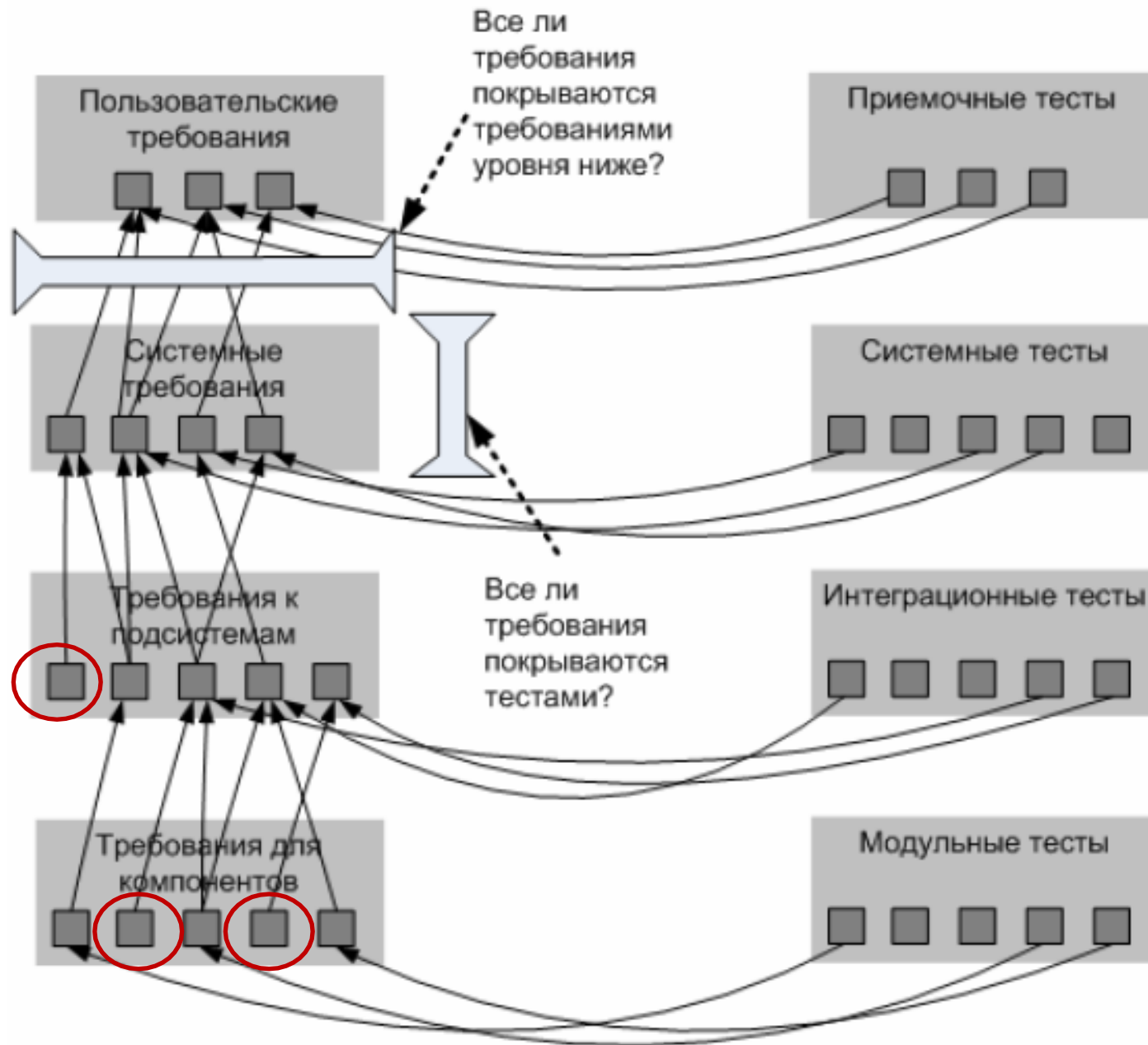
Методы анализа связей требований

Метод анализа	Описание	Поддерживаемый процесс
Анализ влияния	Анализ входящих связей с целью ответа на вопрос: «Что будет если изменить это требование?»	Процесс изменения
Анализ последствий	Анализ исходящих связей с целью ответа на вопрос: «Нам это действительно нужно?»	Анализ экономической целесообразности
Анализ покрытия	Анализ связей с целью ответа на вопрос: «Все ли учтено?» Обычно используется для оценки прогресса работы.	Проектирование. Отчетность руководству

Графическое представление анализа влияния и анализа последствий



Анализ покрытия



Выгоды от использования связей между требованиями

1. БОльшая уверенность в достижении целей
2. Возможность оценивать влияние изменений
3. Возможность оценить вклад подрядчиков и субподрядчиков
4. Возможность контролировать ход проекта и оценивать объем выполненной работы
5. Возможность сопоставлять затраты и возможную выгоду (определять экономическую целесообразность)

Область проблем и область решений

Уровень требований	Область	Точка зрения	Цель

Пример: система управления дорожным движением

пользовательские требования



Заинтересованные лица: Организовать круговое движение на перекрестке, чтобы максимизировать транспортный поток перекрестка при минимизации риска возникновения дорожных происшествий и минимизации стоимости обслуживания решения



Пример: система управления дорожным движением системные требования



Системный архитектор:

Предлагает различные системные решения:
установка светофоров, организация кругового движения, возведение моста



Пример: система управления дорожным движением

требования к подсистеме



Архитектор:

Разрабатывает варианты проекта виадуков с учетом существующих физических ограничений, налагаемых окружающей средой

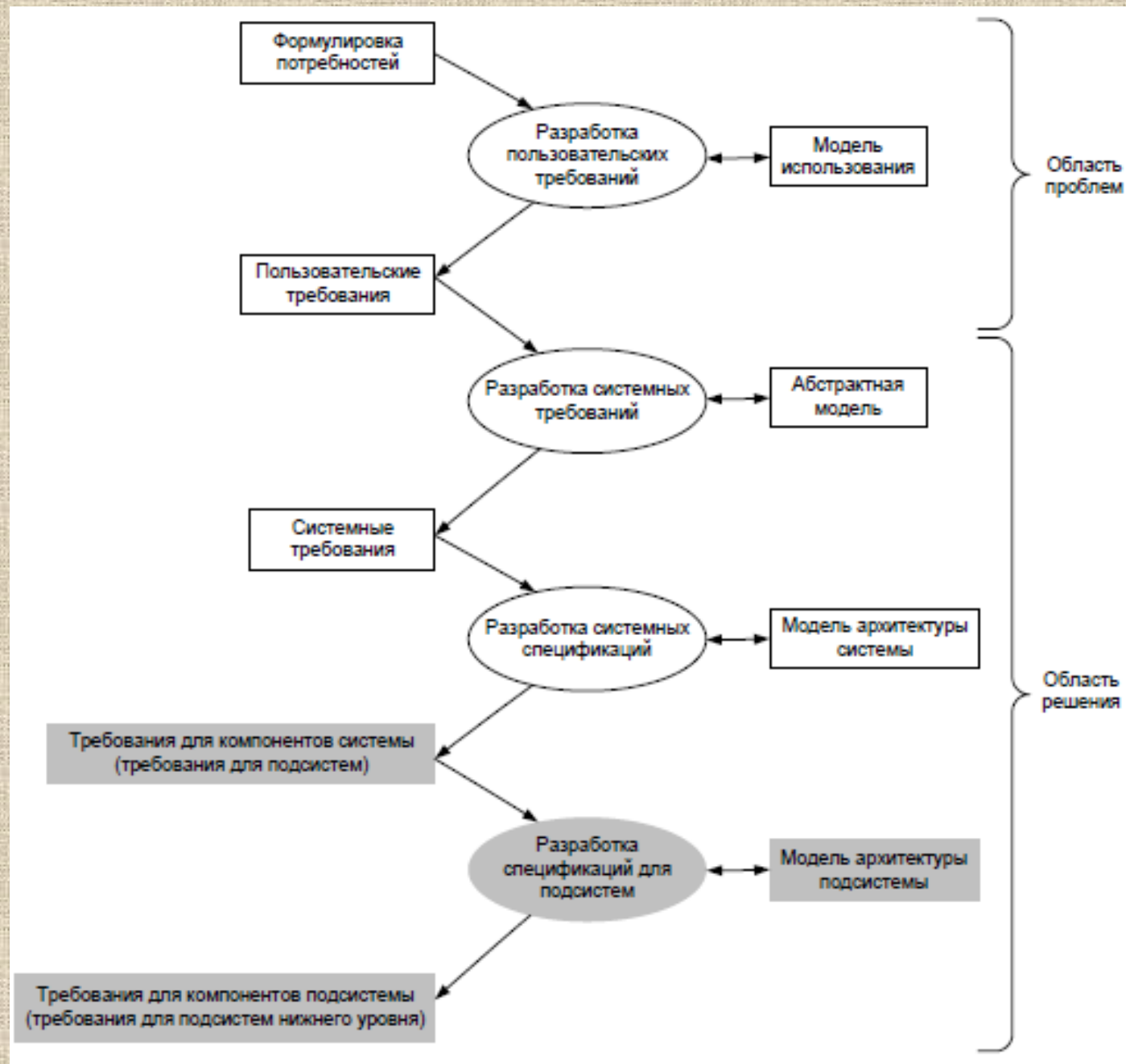


**Если нет четкого разделения между
областью проблем и областью решений, тогда:**

1. Недостаточное понимание существующих проблем
2. Невозможность определить границы (масштаб) системы и понять какой функционал должен в нее входить, а какой нет
3. Доминирование разработчиков и исполнителей в дискуссиях о системе, поскольку единственное описание, существующее для системы, описывает ее в терминах реализации, а не в формулировках проблем
4. Невозможность нахождения наилучшего решения из-за ограничений свободы в выборе решения.

ОБЩИЙ ПРОЦЕСС РАЗРАБОТКИ ТРЕБОВАНИЙ

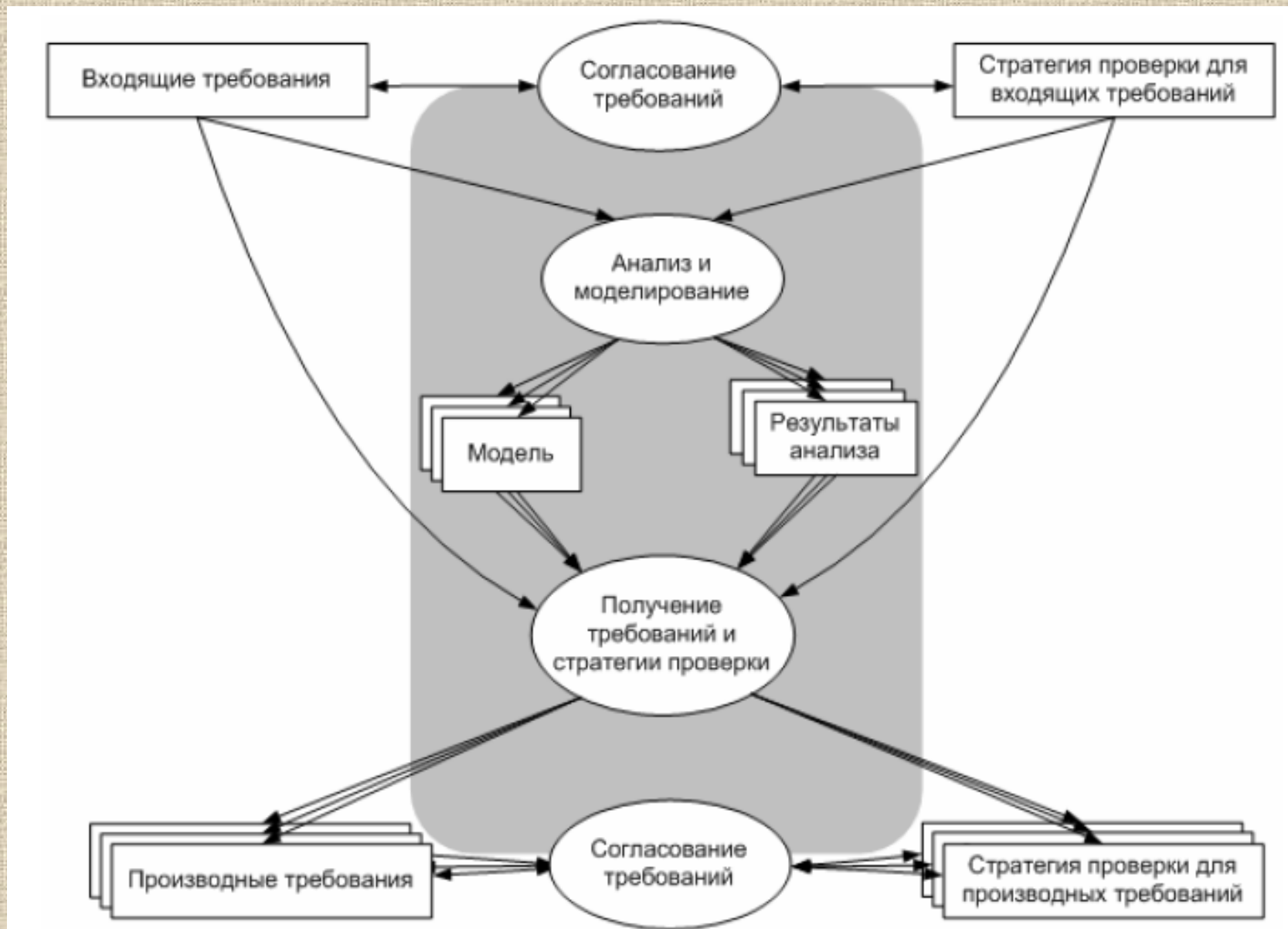
Процесс разработки информационной системы



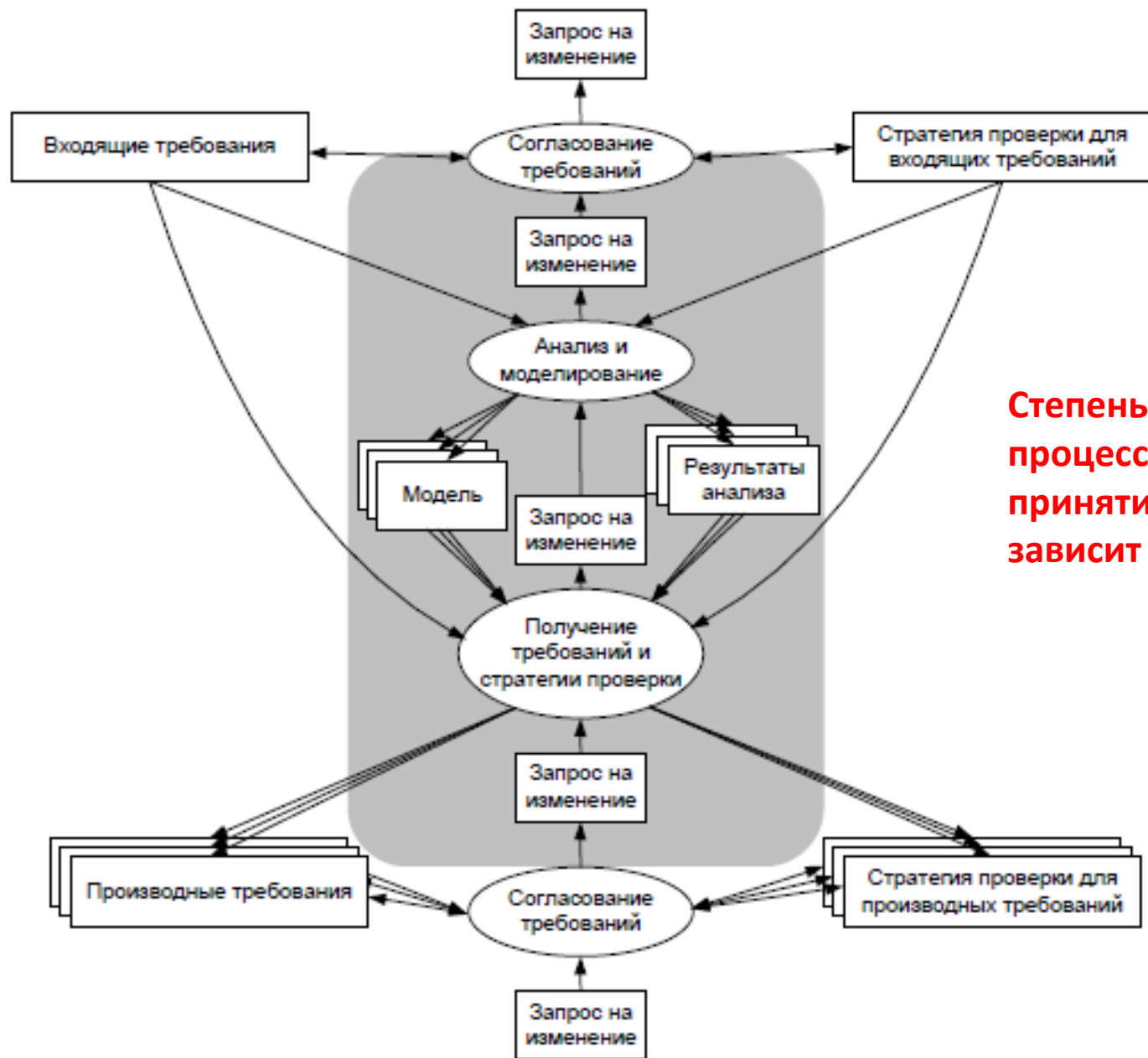
Стратегия проверки и критерии приемки



Процесс разработки требований для идеального мира



Разработка требований в контексте изменений



Степень формализации процесса и «легкость» принятия изменений зависит от стадии проекта.

Статусы требований

1. Согласования
2. Проверки
3. Удовлетворения

Диаграмма состояний статуса согласования

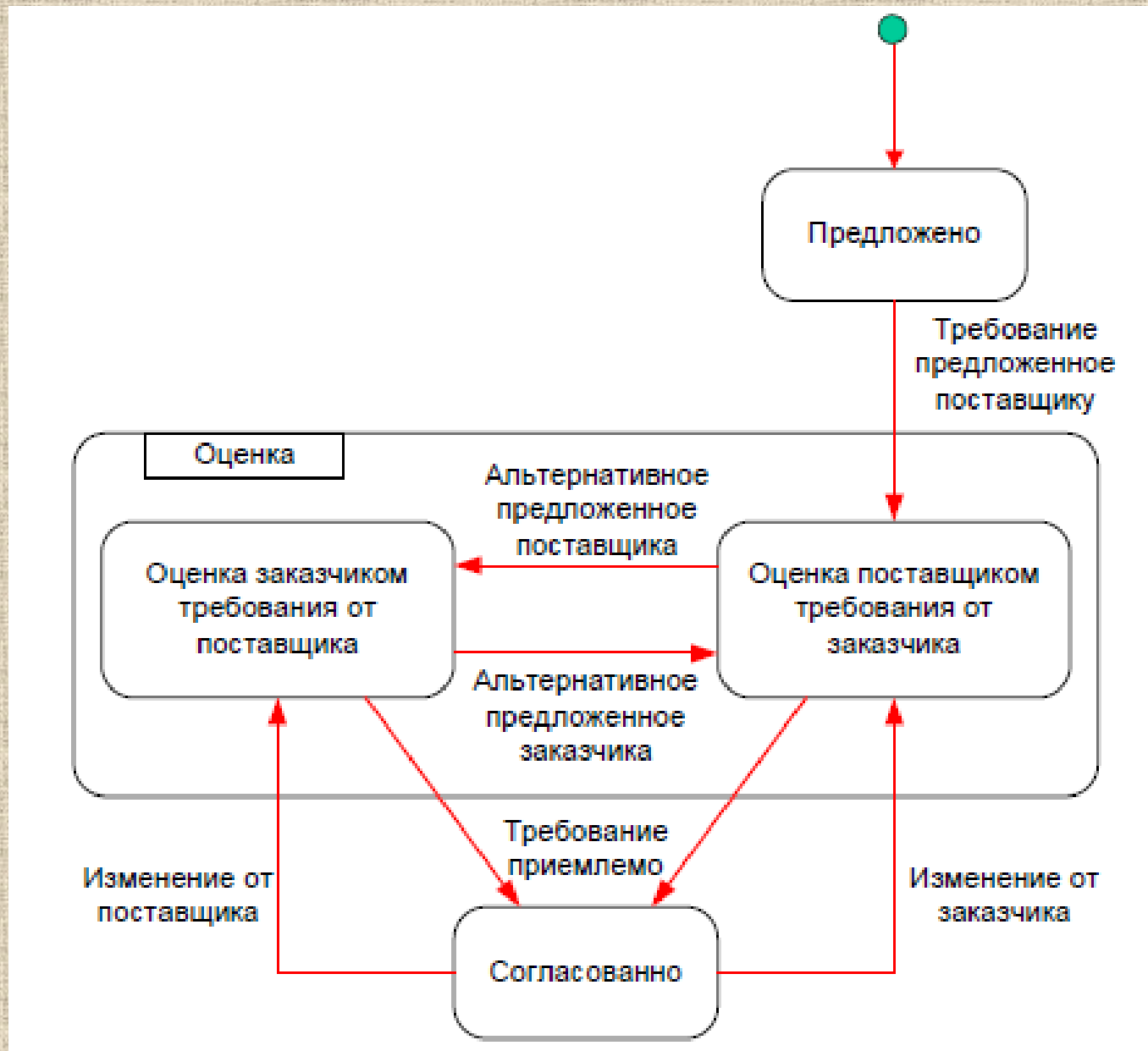


Диаграмма состояний статуса проверки

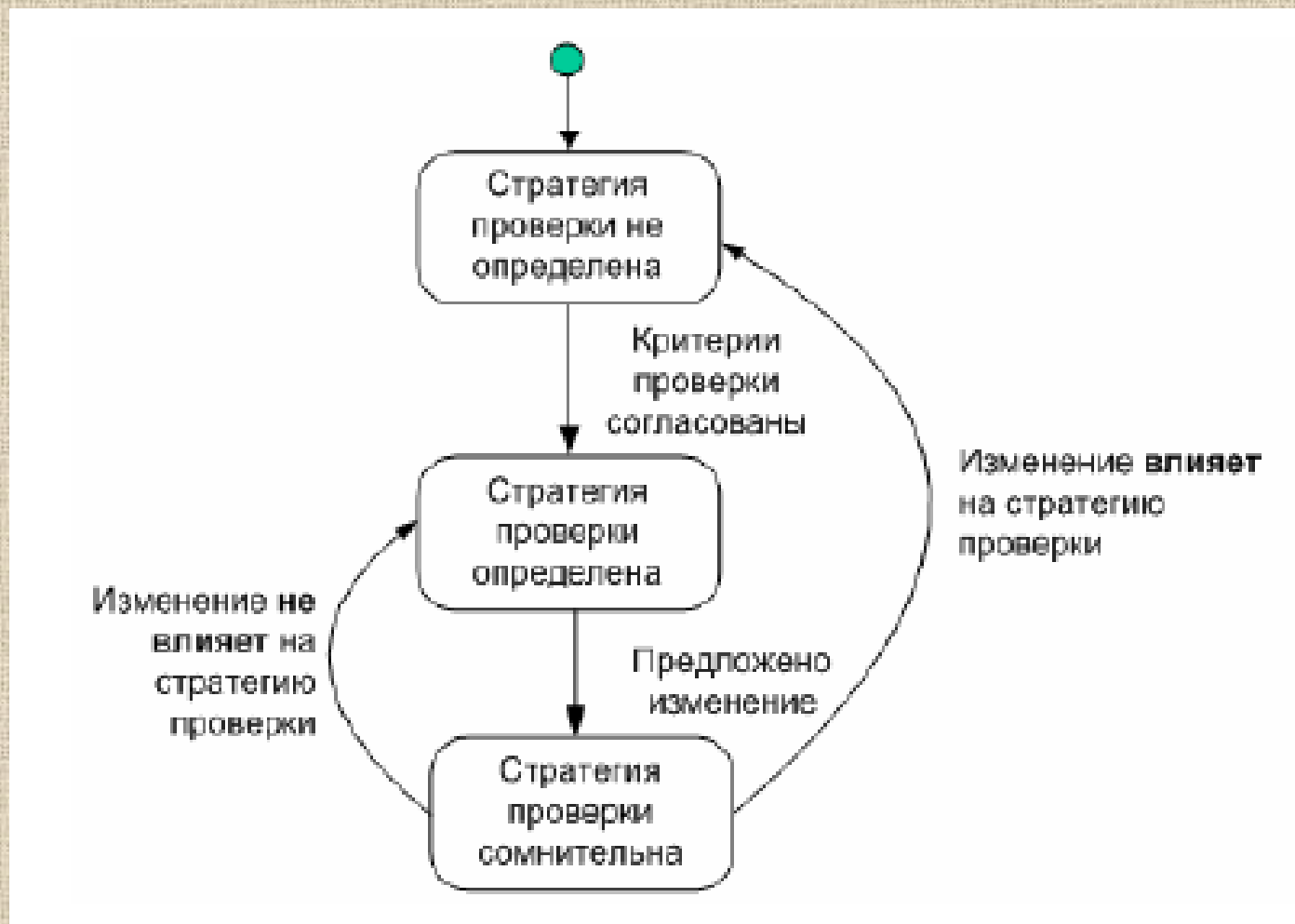


Диаграмма состояний статуса удовлетворения



Оценка приемлемости требований

- ✓ Является ли требование полным?
- ✓ Является ли требование ясным?
- ✓ Является ли требование существенным?
- ✓ Является ли план проверки требований понятным и приемлемым?

Причины отклонений требований

- Отсутствие части информации в описании требования
 - TBA (to be agreed) – необходимо согласовать
 - TBC (to be complete) – необходимо завершить
 - TBD (to be decided) – необходимо принять решение
- Недостаток ясности
- Невозможность реализации
- План проверки неприемлем

Общий процесс разработки требований: итоги

- ✓ согласование **входящих требований**
- ✓ анализ **входящих требований** для определения рисков и потенциальных проблем, связанных с удовлетворением требований
- ✓ создание одной или нескольких моделей для исследования возможных стратегий получения последующих требований
- ✓ формирование требований, полученных из входящих требований при помощи результатов **анализа и моделирования**
- ✓ согласование **производных требований** с командой (командами) сотрудников, ответственных за их реализацию
- ✓ установление связей «удовлетворяющего» типа между **входящими** и **производными требованиями**
- ✓ установление связей «проверяющего» типа между **производными требованиями** и соответствующей стратегией проверки

Идеальное состояние требования

- ✓ Согласовано между заказчиком и исполнителем
- ✓ Имеет согласованную стратегию его проверки
- ✓ Удовлетворяется требованиями более низкого уровня (или спецификацией системы)

НАПИСАНИЕ И АНАЛИЗ ТРЕБОВАНИЙ

(Требования для требований)

Определение заинтересованных лиц – отправная точка при разработке требований

Заинтересованная сторона	Роль
Автор	Создает требования и оформляет изменения
Издатель	Выпускает и архивирует документ требований
Цензор	Рецензирует требования и предлагает изменения
Конструктор, системный аналитик, разработчик	Анализирует требования и обсуждает изменения

Написание требований

- ✓ Документ с требованиями должен быть удобным для чтения
- ✓ Наборы требований должны быть удобными для работы с ними

Возможности, которые должен предоставлять документ с требованиями

- Однозначно идентифицировать каждое требования
- Классифицировать каждое требование различными способами, например:
 - по важности
 - по типу (функциональность, производительность, ограничение, безопасность)
 - по срочности (дата реализации)
- Отслеживать статус каждого требования, такой как:
 - согласования
 - удовлетворения
 - проверки
- Оценивать требование с разных сторон, таких как:
 - стратегии проверки
 - критерии тестирования
 - рациональность
- Анализировать каждое требование в контексте целого документа, т.е. в окружении требований
- Находить в документе определенное требование по контексту, классификации или другим признакам
- Устанавливать связи между требованиями и обеспечивать легкий переход по этим связям между требованиями

Разработка структуры требований: возможности

- ✓ *Минимизировать* общее количество требований
- ✓ *Лучше осмыслить* большой объем информации
- ✓ *Отыскать* наборы требований, относящихся к определенной теме
- ✓ *Выявить* пробелы и повторения
- ✓ *Исключить* конфликт (противоречие) между требованиями
- ✓ *Управлять* этапами реализации
- ✓ *Отклонить* малоинформативные требования
- ✓ *Оценить* требования
- ✓ *Повторно использовать* требования в последующих проектах

Концепция ключевых требований

KURs – key user requirements

ключевые пользовательские требования

*Если решение не предполагает <эту> возможность
(функцию, опцию и т.д.), стану ли я в этом случае его приобретать?*

Если система не обеспечивает <этого>, будет ли система все еще нужна мне?

KPIs – key performance indicators

ключевые показатели производительности

Связанность и согласованность требований

Каждое требование обычно имеет *единственную первичную* классификацию (например, его местоположение в контексте документа) и *множественное количество вторичных* классифицирующих свойств, использующих возможности атрибутов и связей.

Это дает возможность искать, фильтровать и сортировать требования.

Использование атрибутов требований

[SH234] Система управления скорой помощью должна быть способна принимать до 100 ста вызовов одновременно

Автор:	R. Thomas
Приоритет:	Обязательное
Релиз:	1
Статус рецензирования:	Одобрено
Возможность проверки:	Да
Способ проверки:	Симуляция, затем системные тесты

Категории атрибутов требований

- ✓ **Идентификация** (идентификатор, название)
- ✓ **Внутренние характеристики** (основной тип, подтип, тип продукта/процесса, количественный/качественный тип, фаза жизненного цикла)
- ✓ **Приоритет и важность**
- ✓ **Источник и владелец**
- ✓ **Контекст** (набор требований/документ, объект, границы/рамки)
- ✓ **Проверка и утверждения**
- ✓ **Поддержка процесса** (статусы согласования, проверки, удовлетворения, рецензирования)
- ✓ **Уточнение** (необходимость, комментарии, вопросы, ответы)
- ✓ **Прочее** (зрелость/стабильность, уровень риска, оценочная стоимость, фактическая стоимость, релиз продукта)

Важность требования

О (*обязательный*) :

обязательный верхний (или нижний) предел значения величины

Ж (*желаемый*) :

желаемое значение

Н (*наилучший*) :

наилучшее значение

«Система должна поддерживать функционирование [О:50, Ж:100, Н:200] одновременно работающих пользователей».

Язык требований. Требования области проблемы

<Тип пользователя> должен иметь возможность <описание возможности>

<Тип пользователя> должен иметь возможность <описание возможности>
с <показатель производительности> от <момент отсчета>,
находясь в <условия эксплуатации>

Оператор должен иметь возможность *произвести выстрел*
в течение 3 секунд с момента обнаружения цели радаром,
находясь в *сложных морских условиях*

Язык требований. Системные требования

**<Система> должна <выполняемая функция>
не менее чем <количество> <объект>
функционируя в <условия эксплуатации>.**

*Телекоммуникационная система должна обеспечивать телефонную связь
не менее чем с 10 абонентами,
функционируя в условиях отсутствия внешнего источника электропитания*

**<Система> должна <выполняемая функция> <объект>
каждые <показатель производительности> <единица измерения>**

*Кофе-машина должна производить горячий напиток
каждые 10 секунд*

Использование шаблонов требований



Примеры шаблонов требований с ограничениями

Тип ограничения	Шаблон
Производительность/возможность	<Система> должна <выполняемая функция> <объект> не менее чем <производительность> раз в <единица измерения>
Производительность/возможность	<Система> должна <выполняемая функция> <объект> типа <характеристика> в течение <производительность> <единица измерения>
Производительность/мощность	<Система> должна <выполняемая функция> не менее чем <количество> <объект>
Производительность/своевременность	<Система> должна <выполняемая функция> <объект> в течение <производительность> <единица измерения> с момента <событие>
Производительность/периодичность	<Система> должна <выполняемая функция> не менее чем <количество> <объект> в течение <производительность> <единица измерения>
Способность к взаимодействию/мощность	<Система> должна <выполняемая функция> <объект> состоящий из не менее чем <производительность> <единица измерения> с <внешняя сущность>
Устойчивость/периодичность	<Система> должна <выполняемая функция> <объект> с <производительность> <единица измерения> каждые <производительность> <единица измерения>
Окружение/работоспособность	<Система> должна <выполняемая функция> <объект> функционируя в <условия эксплуатации>

Критерии для написания текста отдельного требования

- 1. Атомарность.** Каждое утверждение (формулировка требования) должно представлять собой один элемент иерархии, пригодный для установления связей с ним.
- 2. Уникальность.** Каждое требование должно иметь собственный уникальный идентификатор.
- 3. Выполнимость.** Требование должно быть технически реализуемо в установленные сроки, в рамках выделенного бюджета.
- 4. Законность.** Требование не должно противоречить применяемому законодательству.
- 5. Ясность.** Требование должно быть понятно сформулировано (исключать неоднозначное толкование).
- 6. Точность.** Требование должно быть точным и лаконичным.
- 7. Проверяемость.** Должна существовать возможность проверки реализации каждого конкретного требования.
- 8. Абстрактность.** Формулировка не должна навязывать определенные технические решения, характерные для более низких уровней требований (спецификаций).

Критерии, применимые к набору требований

- 1. Полнота:** все необходимые требования зафиксированы.
- 2. Непротиворечивость:** не существует требований, противоречащих друг другу.
- 3. Отсутствие избыточности:** каждое требование, сформулировано только один раз (нет повторов)
- 4. Модульность:** требования, близкие друг другу по смыслу, содержаться в одном разделе
- 5. Структурированность:** наличие ясной и четкой структуры документа с требованиями
- 6. Удовлетворенность:** достигнут требуемый уровень покрытия требований связями типа «удовлетворяется (посредством)»
- 7. Тестируемость:** достигнут требуемый уровень покрытия требований тестами.

Пример требования

1. Система должна обеспечивать максимальный уровень производительности в течение всего времени работы, за исключением аварийных ситуаций, при которых она должна обеспечивать уровень производительности до 125%, но только если аварийная ситуация не длится более чем 15 минут, - в противном случае система должна уменьшить уровень производительности до 105%; но в случае, если удастся достигнуть уровня производительности только 95%, система должна активировать режим «исключительно малого уровня» и поддерживать этот уровень в пределах 10% от начального значения в течение, как минимум, 30 минут.

Проблемы:

Вместо одного требования, нужно писать 12.

Лучше всего выделить 4 отдельных условия эксплуатации – нормальное, аварийное, аварийное более 15 минут, режим «исключительно малого уровня», - и описать требования для каждого из этих условий.

Пример требования

2. Система должна обеспечивать основные функции текстового редактора, удобные для использования необученным персоналом, и должна работать в условиях «тонкого» Ethernet'a, проложенного по воздушной системе кабельных каналов с интегрированными сетевыми адаптерами, поставляемыми с дополнительными модулями памяти, при необходимости.

Проблемы:

Совершенно непонятен объем требования.

Например, это требование можно интерпретировать и так: «Система должна обеспечивать основные функции текстового редактора... при необходимости».

Так требуется ли текстовый редактор или нет?

Правила формулирования требований

- 1. Избегать хаоса:** формулируя требование, необходимо сконцентрироваться на самом важном.
- 2. Избегать «лазеек»:** например, таких выражений, как «если это необходимо», поскольку такие «лазейки» делают требование неоднозначным и зачастую бесполезным
- 3. Избегать размещения более одного требования в один параграф:** зачастую наличие в одном параграфе более одного требования легко определить по наличию союза «и»
- 4. Избегать рассуждений**
- 5. Избегать «размытых» понятий и слов:** обычно, в основном, часто, нормально, типично
- 6. Избегать использования неопределенных терминов:** например, удобный в использовании, универсальный, гибкий
- 7. Избегать принятия желаемого за требуемого:** например, 100% надежный, приятный для всех пользователей, безопасный, подходящий для всех платформ, не должен никогда ломаться, обрабатывать все неожиданные сбои.

Последовательность шагов в процессе работы с требованиями

- ✓ Определите структуру требований и попытайтесь совершенствовать ее в процессе работы с требованиями.
- ✓ Фиксируйте (записывайте) требования как можно раньше, даже если они весьма далеки от совершенства.
- ✓ Заранее определите атрибуты требований, которые в последующем будут использоваться для классификации и детализации требований.
- ✓ Как можно быстрее подготовьте первый вариант требований, для того чтобы стимулировать получение отзывов.
- ✓ Постоянно совершенствуйте (улучшайте) требования в процессе работы, удаляя повторения, преждевременные и недозволенные технические решения, противоречивость.
- ✓ Постоянно обсуждайте требования, собирайте замечания и, не откладывая в долгий ящик, корректируйте требования, создавая новые их версии (принцип «мозгового штурма»).
- ✓ Демонстрация пользователям наработок гораздо лучше, чем «экспертный анализ специалистов».

Разработка требований в области проблем. Подробнее

Сбор требований в области проблем

Область проблем – окружение, область или среда, в которой система будет использоваться.

Первый вопрос к потенциальному пользователю

Что вы хотите, чтобы система делала?

Зачем вам нужна эта система?

Какие задачи должна решать необходимая вам система?

Что вы хотите, чтобы *Вы* могли делать?

Я хотел бы иметь возможность ...

**Требования типа «возможность»
являются ключевыми в области проблем**

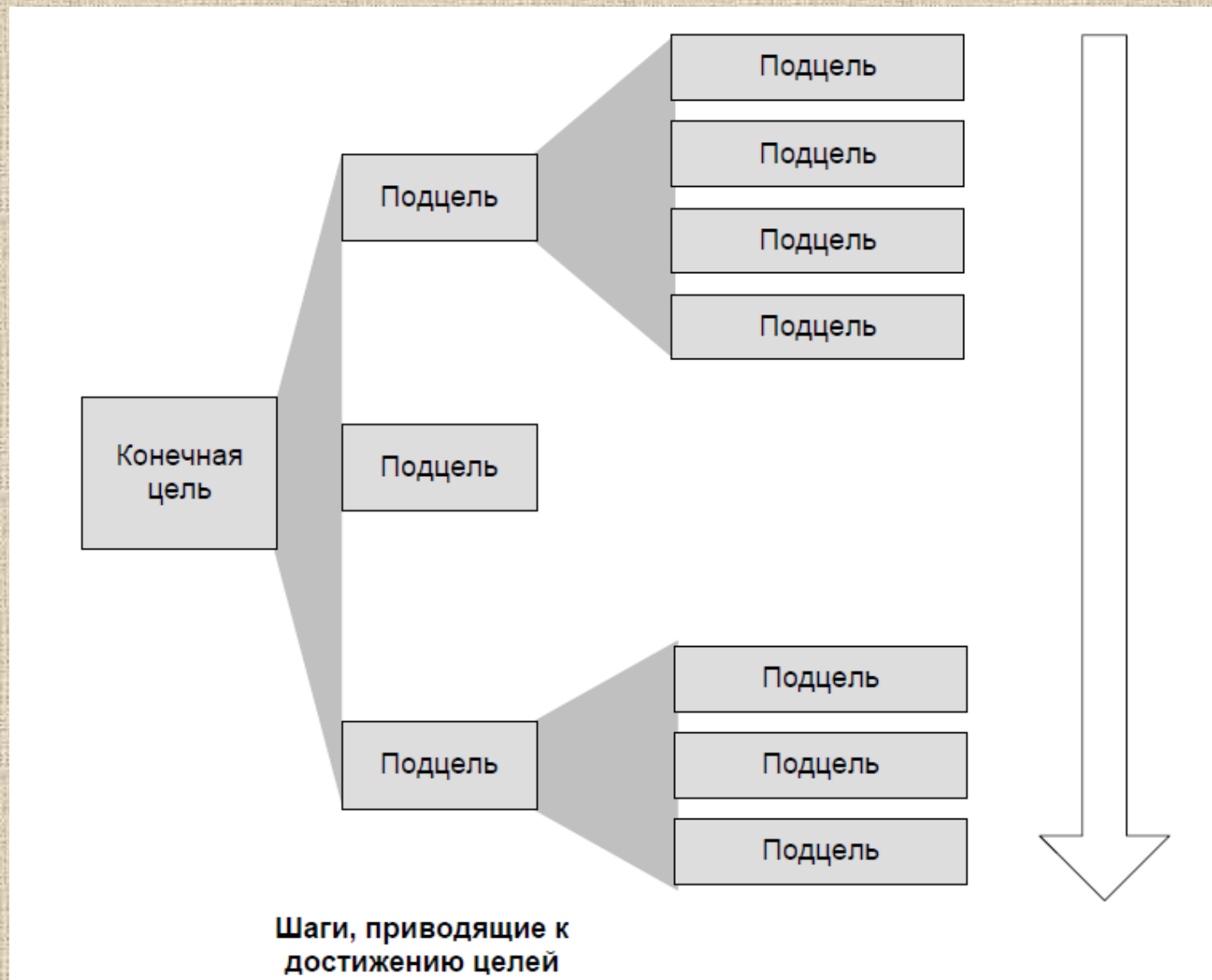
Ключевые вопросы разработки требований в области проблем



Заинтересованные стороны (stakeholders)

- **Руководство**
- **Инвесторы**
- **Пользователи системы**
- **Обслуживающий персонал**
- **Обучающий персонал**
- **Покупатели**
- **Маркетинг и продавцы**
- **Эксперты по эргономике и эффективности**
- **Эксперты по области применения**
- **Правительство**
- **Стандартизирующие органы**
- **Общественное мнение**
(географический и национальный факторы)
- **Регулирующие органы**

Метод разработки сценариев использования



Сценарий использования как иерархия целей

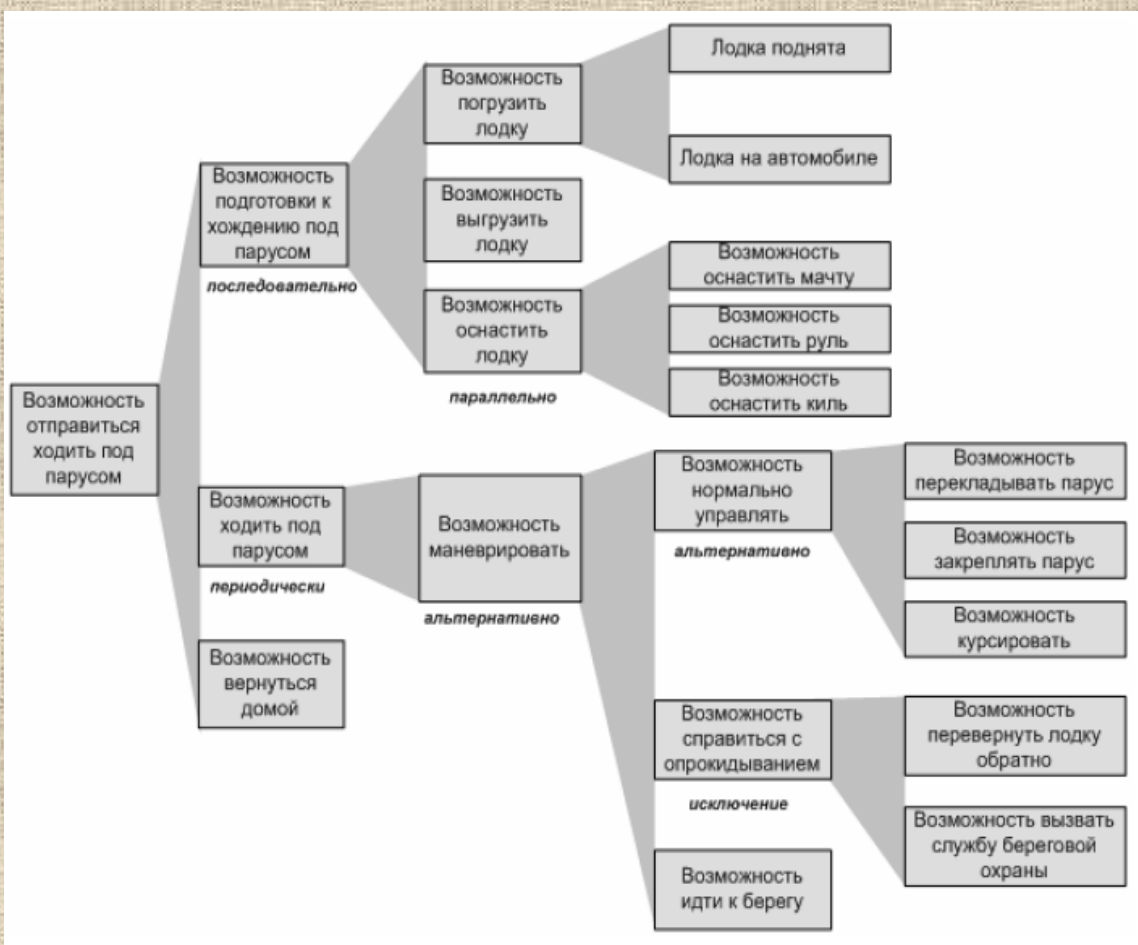
Порядок действий для получения сценария использования

1. Получить характеристики конечной цели (состояния)
2. Установить необходимые возможности (пользовательские требования) для достижения конечной цели
3. Разбить крупные шаги на более мелкие (промежуточные)
4. Строго соблюдать иерархическую структуру
5. Перепроверять информацию на каждом этапе
6. Избегать описания конкретных решений

Преимущества использования сценариев

- ✓ Позволяют представителям заинтересованных сторон пошагово проигрывать сценарии функционирования
- ✓ Позволяют находить пропущенные шаги
- ✓ Различные заинтересованные стороны могут предлагать различные сценарии
- ✓ Позволяют строить хронологические последовательности

Характеристики сценариев использования



- ✓ Временная последовательность в общих чертах
- ✓ Узловые элементы сценария являются возможностями верхнего уровня
- ✓ Альтернативы
- ✓ Периодически повторяющееся поведение
- ✓ Моменты, когда последовательность не имеет значения (параллельные ветви)
- ✓ Исключения

Пример сценария использования, описывающего проведение выходного дня на парусной лодке, которую можно транспортировать на легковом автомобиле

Источники пользовательских требований

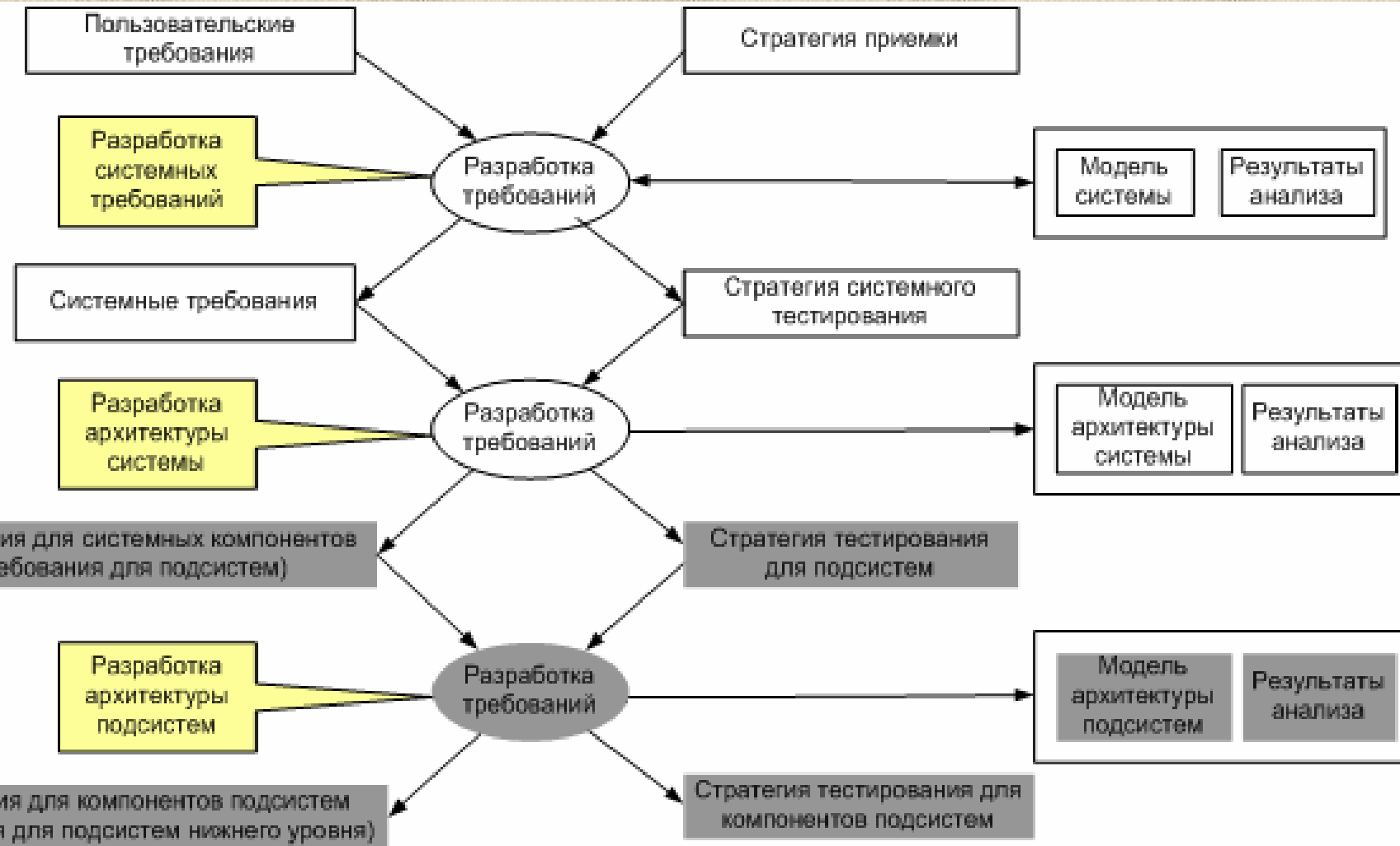
- Интервью с представителями заинтересованных сторон
- Сценарии использования (обычно получаемые в результате интервью)
- Описательная документация (например, отчеты, анализ рынка и т.д.)
- Действующие системы, которые необходимо модернизировать
- Проблемы, обнаруженные в существующих системах, и идеи как их исправить
- Опыт работы с аналогичными системами
- Прототипы, макеты, эскизы
- Возможности новых технологий (согласованные с заинтересованными сторонами)
- Результаты исследований
- Опросы
- Наблюдение за работой персонала или изучение видеозаписей их работы

Разработка требований в области проблем: заключение

1. Пользовательские требования должны быть краткими и понятными, насколько это возможно.
2. Пользовательские требования не должны быть техническими, но в то же время они должны быть реалистичными (реализуемыми).
3. Наиболее часто встречающиеся проблемы:
4. Пользовательские требования должны быть разработаны как можно раньше
5. Пользовательские требования должны быть распределены по зонам ответственности заинтересованных сторон
6. Пользовательские требования должны быть хорошо структурированными и (там, где это возможно) должны быть увязаны с первоисточником информации, из которого они были получены.
7. Заинтересованные стороны должны нести ответственность за содержание требований.
8. За сбор и оформление требований должны отвечать обученные специалисты-аналитики.

Разработка требований в области решений

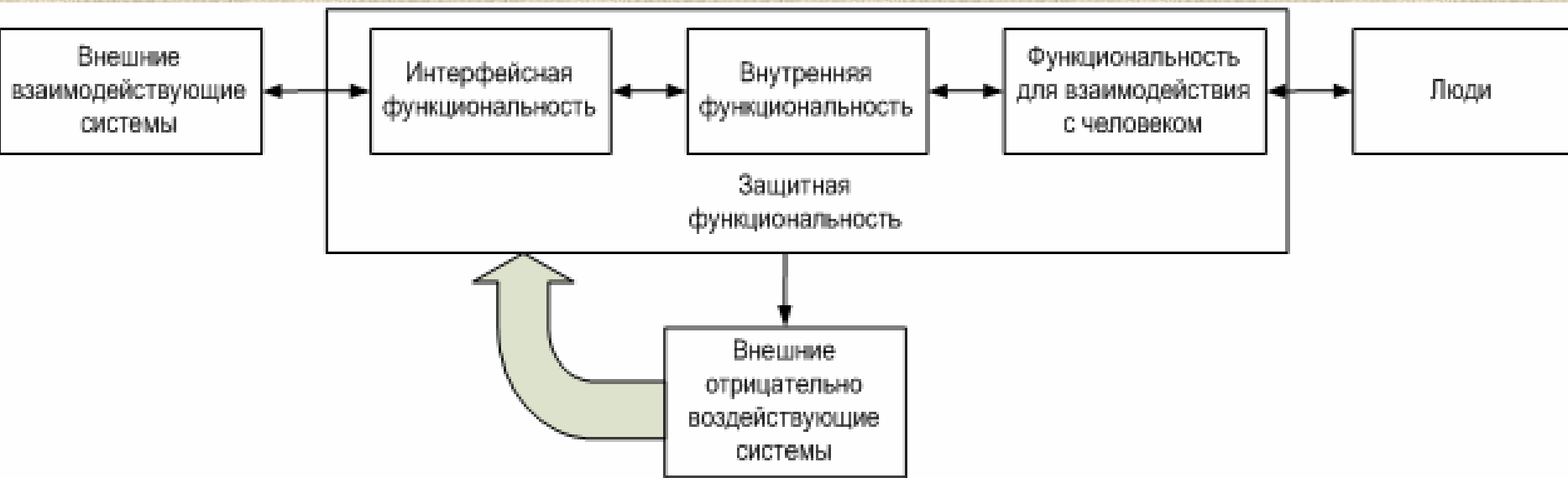
Последовательность шагов разработки требований в области решений



Содержание модели системы для разработки системных требований

- ✓ **Внутренняя функциональность**, которую система должна реализовывать. Описывается **ЧТО** система должна реализовать, а не **КАК** это делать.
- ✓ Функциональность, необходимая для **взаимодействия системы со своим окружением** (с внешними системами).
- ✓ Функциональность, позволяющая **людям взаимодействовать с системой**.
- ✓ Функциональность, которая **предохраняет систему от «ложного срабатывания»**, что может быть вызвано вредным (опасным) воздействием со стороны других систем их ее окружения.

Контекст системы и типы функциональности



Разработка требований в области решений: заключение

1. В области решений происходит последовательное итерационное преобразование пользовательских требований в системные и последующее преобразование системных требований в требования для подсистем.
2. При описании системных требований должны быть описаны четыре типа функциональности. Помимо внутренней функциональности необходимо рассматривать интерфейсную функциональность, функциональность для обеспечения взаимодействия с людьми, защитную функциональность.
3. На более поздних этапах могут появляться требования, представляющие собой ограничения различного характера.

Спасибо за внимание!

Вопросы?