



Нижегородский государственный университет им. Н.И. Лобачевского

Введение в анализ требований к программному продукту

При поддержке компании Intel

Козинов Е.А., кафедра математического
обеспечения ЭВМ, факультет ВМК

Содержание

- ❑ Понятие требования
- ❑ Виды требований
- ❑ Свойства требований
- ❑ Цели выделения и управления требованиями
- ❑ Методы выделения требований
- ❑ Способы описания требований
- ❑ Описание проекта
- ❑ Трассировка требований

Цели

- ❑ Рассмотреть основные понятия и виды требований с точки зрения проекта, связанного с разработкой программного обеспечения.
- ❑ Рассмотреть частный случай выделения требований в рамках конкретного примера.
- ❑ Рассмотреть элементы системного моделирования и связь его с требованиями.

ТРЕБОВАНИЕ



Понятие требования...

□ Имеет несколько значений

[<http://ru.wikipedia.org/wiki/Требование>]:

- **Требование (действие)** – действие, выражающееся в настойчивой, категорической, просьбе исполнить что-либо.
- **Требование (условие)** – условие, стандарт (пример – технические требования к продукту).
- **Требование (документ)** – документ или его раздел, документально фиксирующий потребности. Требование может быть приложением к приказу, договору, запросу и другим документам, либо являться самостоятельным указывающим, побуждающим либо информационным документом.
- **Требование (платежное)** – расчетный документ, содержащий требование кредитора (поставщика) к должнику (плательщику) об уплате определенной денежной суммы через банк.



Понятие требования

- ❑ **Требования к программному обеспечению** – совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации.

- ❑ **Требование** – это (IEEE Standard Glossary of Software Engineering Terminology):
 - условия или возможности, необходимые пользователю для решения проблем или достижения целей;
 - условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам;
 - документированное представление условий или возможностей для пунктов 1 и 2.



ТИПЫ ТРЕБОВАНИЙ



Типы требований

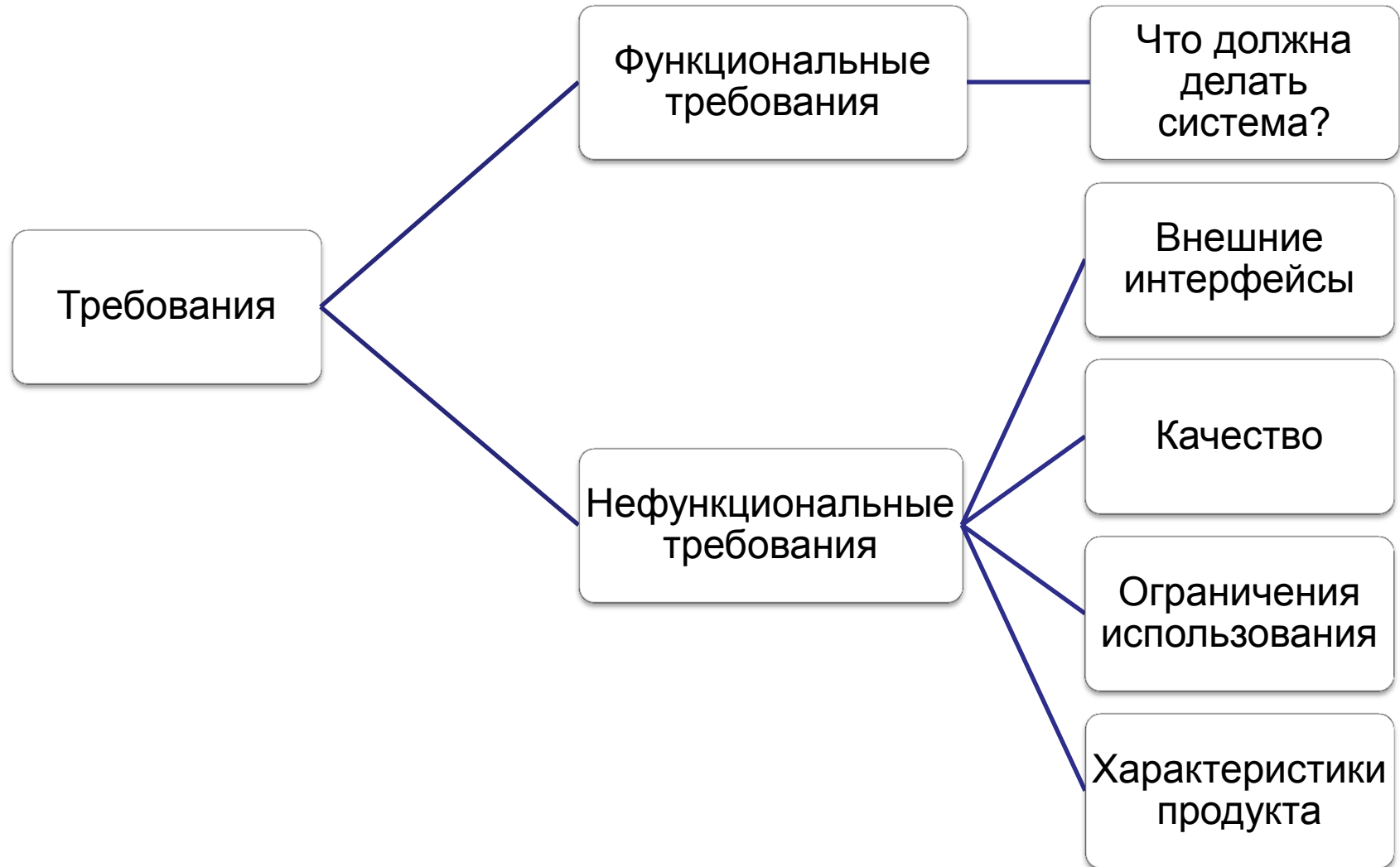
❑ *Пользовательские требования*

- Предложения на естественном языке + диаграммы сервисов, обеспечиваемых системой, и их эксплуатационные ограничения.
- Разрабатываются для заказчика.

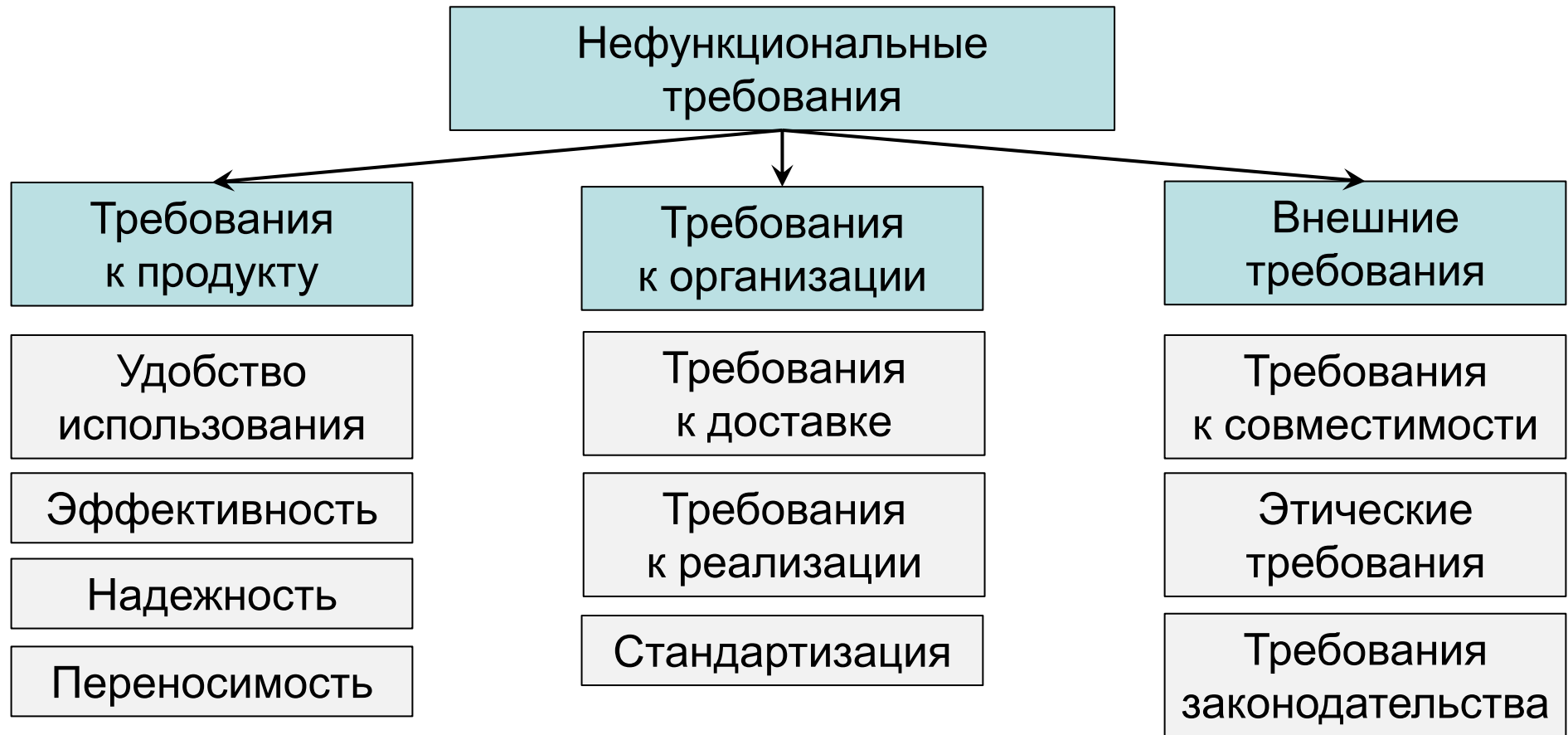
❑ *Системные требования*

- Структурированный документ, содержащий детальное описание функций системы, сервисов и эксплуатационных ограничений.
- Определяют, что должно быть реализовано в системе.

Виды требований...



Виды требований...



**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*

Количественные показатели измерения нефункциональных требований

Свойство системы	Показатели
Скорость	Количество обработанных транзакций в секунду Время отклика на события Время обновления экрана
Размер	МБ Количество внешних носителей
Простота использования	Время обучения Количество окон помощи
Надежность	Среднее время бесперебойной работы Вероятность недоступности Оценка возможности падения системы
Прочность	Время перезапуска системы после сбоя Процент событий, приводящих к падению системы Вероятность порчи данных в результате падения
Переносимость	Количество целевых систем

Виды требований

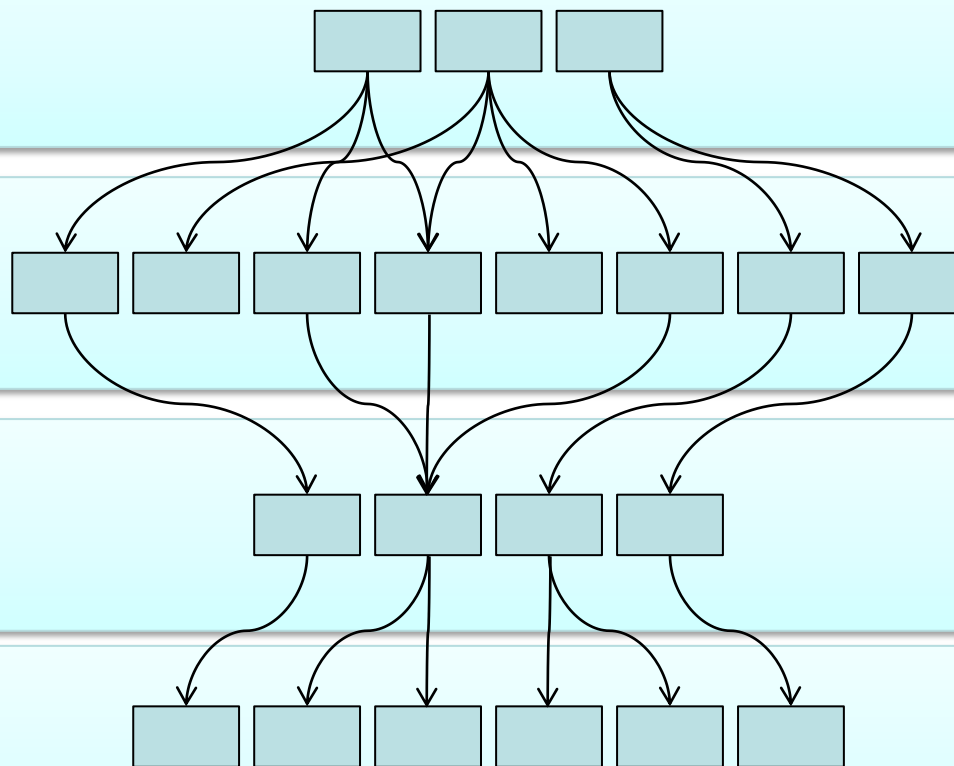
- Требования могут быть разных уровней:

Бизнес-требования

Требования пользователей

Функциональные требования

Требования к подсистемам



Виды требований

❑ **Пример бизнес-требования:**

- Компании авиаперевозчику необходимо увеличить продажи билетов в два раза за счет применения разрабатываемой системы.

❑ **Примеры требований пользователя:**

- В системе необходимы визуальные формы, позволяющие задать точку отправления и прибытия клиента.
- После выбора маршрута должна быть возможность забронировать билеты.
- В верхней части окна заказа должен располагаться логотип компании.

❑ **Пример функциональных требований:**

- Маршрут может быть создан, отредактирован, сохранен и удален.
- Перед взиманием оплаты за проезд маршрут должен быть забронирован.



Классификация требований с точки зрения RUP...

- ❑ Требования каких категорий должны быть описаны в документе с требованиями?

Классификация требований с точки зрения RUP...

- ❑ Требования каких категорий должны быть описаны в документе с требованиями?
- ❑ Модель FURPS+ (стандарт IEEE Std 610.12.1990):
 - Функциональность (Functionality).
 - Применимость (Usability).
 - Надежность (Reliability).
 - Производительность (Performance).
 - Эксплуатационная пригодность (Supportability).

Классификация требований с точки зрения RUP

- В рамках модели отдельно выделяют еще следующие типы требований:
 - Ограничения проекта.
 - Требования выполнения.
 - Требования к интерфейсу.
 - Физические требования.
 - Требования, указывающие на необходимость соответствия некоторым юридическим и нормативным актами.
 - Требования к лицензированию.
 - Требования к документированию.



СВОЙСТВА ТРЕБОВАНИЙ



Свойства требований...

□ Свойства требований:

- Полнота.
- Ясность.
- Корректность и согласованность.
- Верифицируемость.
- Осуществимость.
- Модифицируемость.
- Трассируемость.
- Наличие количественной метрики и упорядоченность по важности и стабильности.

**Анализ требований к автоматизированным информационным системам*
[\[http://www.intuit.ru/studies/courses/2188/174/info\]](http://www.intuit.ru/studies/courses/2188/174/info)



Свойства требований...

□ Полнота

– При описании требований полнота должна достигаться в двух смыслах:

- Полнота **требования**

Пример:

1. (неполное требование) В системе необходимо предусмотреть поиск рейса.
2. (более полное требование) При заданных точках отправления и прибытия, а также при наличии соединения с сервером с пропускной способностью канала не менее 5 Мб/с система должна обеспечить обнаружение рейса или установить факт его отсутствия за время, не превышающее одной минуты.

- Полнота **совокупности всех** требований

Свойства требований...

□ **Ясность**

- Требование обладает свойством ясности, если оно сходным образом воспринимается всеми участниками проекта (в том числе, заказчиком)
 - Для достижения ясности рекомендуется составлять **гlossарий** – терминологический словарь.
- Для достижения ясности рекомендуется не излагать много требований в рамках одного предложения.

Свойства требований...

❑ **Корректность и согласованность**

- Требования не должны противоречить друг другу.
- Без полноты требований, как правило, невозможно проверить непротиворечивость требований.

❑ **Верифицируемость**

- Проверяемое только требование, обладающее свойством полноты и корректности.
- Объективно проверяемы требования, результаты для которых выражены в измеримых метриках.
- Проверяемые требования – основа контракта между Заказчиком и Исполнителем.
- Любое требование должно содержать способ проверки.

Свойства требований...

❑ **Осуществимость**

- При разработке требований важно требования описать максимально полно с учетом ограничений.
- При отсутствии ограничений возможны проблемы при сдаче проекта.

Пример:

1. (неполное требование) В системе необходимо предусмотреть поиск рейса.
2. (более полное требование) При заданных точках отправления и прибытия, а также при наличии соединения с сервером с пропускной способностью канала не менее 5 Мб/с система должна обеспечить обнаружение рейса или установить факт его отсутствия за время, не превышающее одной минуты.

- Первый вариант требования не всегда осуществим.
- Осуществимость так же определяется финансовыми и временными ограничениями, накладываемыми на проект.

Свойства требований...

□ *Модифицируемость*

- Требования не должны содержать детали реализации.
- Требования могут указывать на необходимость использования конкретных компонент.

Свойства требований...

□ *Трассируемость*

- На любом этапе разработки программного обеспечения необходимо иметь возможность отследить связь между артефактами (документ проекта, диаграмма прецедентов использования, компонент, класс, проверка условия ...) и исходными требованиями.
- Трассировка позволяет выявить нереализованные требования.
- Трассировка позволяет выявить части системы, не связанные с требованиями («мусор»).
- При изменении требований трассировка позволяет выявить зависимые артефакты.



Свойства требований

- ❑ ***Наличие количественной метрики и упорядоченность по важности и стабильности***
 - Наличие количественной метрики позволяет оценить важность требования и оценить границы системы.
 - Нестабильные требования лучше отложить при реализации.

ВЫДЕЛЕНИЕ И УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ



Цели выделения и управления требованиями...

- Причины провалов проектов (35,4 %):
 - **Неполнота требований 13.1%**
 - **Недостаточное привлечение пользователей 12.4%**
 - Недостаток в ресурсах 10.6%
 - **Нереалистические ожидания 9.9%**
 - Недостаток поддержки от руководства 9.3%
 - Изменение требований/спецификаций 8.7%
 - Недостаточное планирование 8.1%
 - Потеря необходимости 7.5%

**Источник: Халл Элизабет, Джексон Кен, Дик Джереми. Разработка и управление требованиями // Gray Publishing, USA (2-е изд. - 2005г.). - 229 с. пер. Корнипаев Илья. Практическое руководство пользователя.*



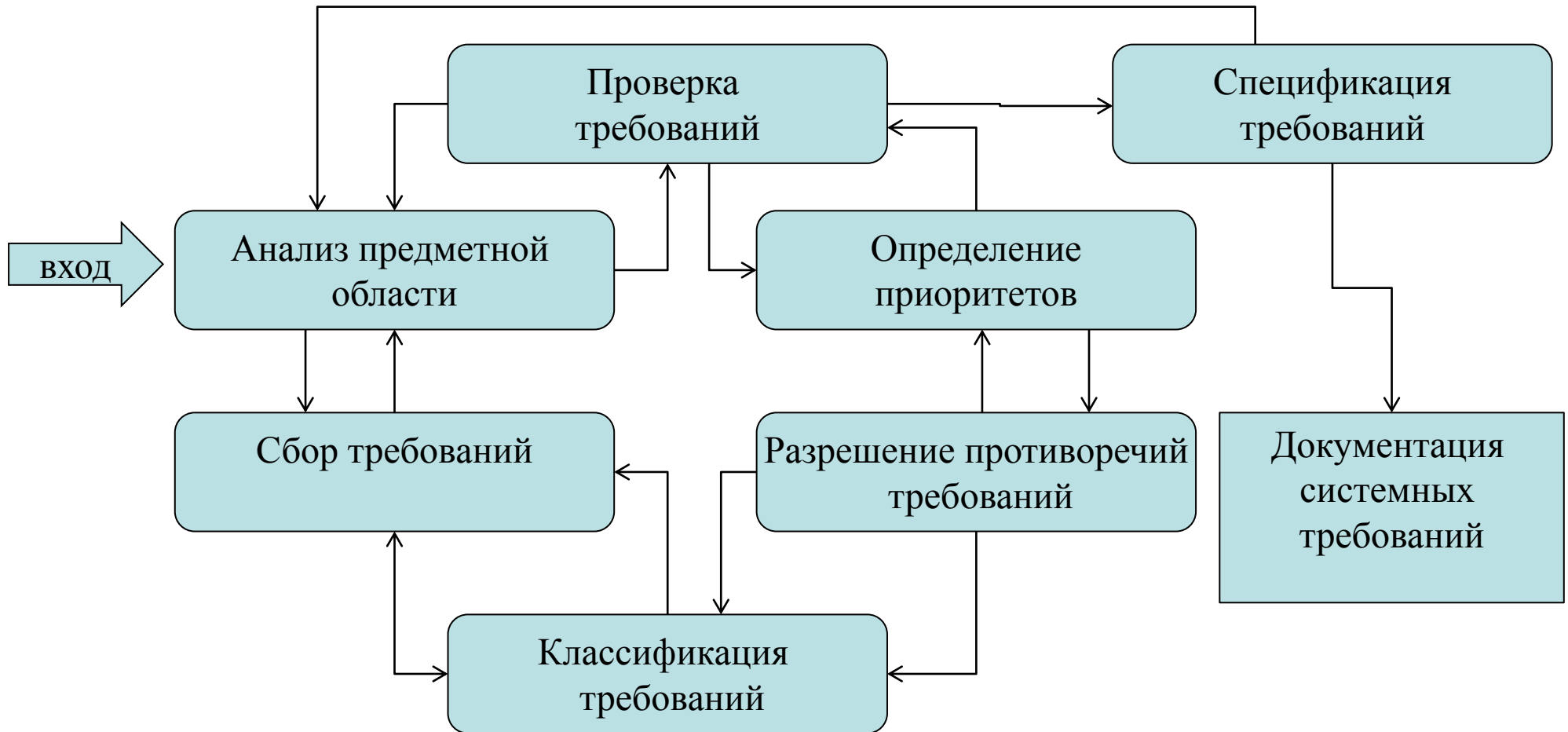
Цели выделения и управления требованиями

- Факторы успехов проектов (37,1%):
 - **Вовлечение пользователей 15.9%**
 - Поддержка руководства 13.9%
 - **Четкая и ясная постановка требований 13.0%**
 - Хорошее планирование 9.6%
 - **Реалистичные ожидания 8.2%**
 - Частые контрольные точки 7.7%
 - Компетентная команда 7.2%
 - Владение (требованиями) 5.3%

**Источник: Халл Элизабет, Джексон Кен, Дик Джереми. Разработка и управление требованиями // Gray Publishing, USA (2-е изд. - 2005г.). - 229 с. пер. Корнипаев Илья. Практическое руководство пользователя.*



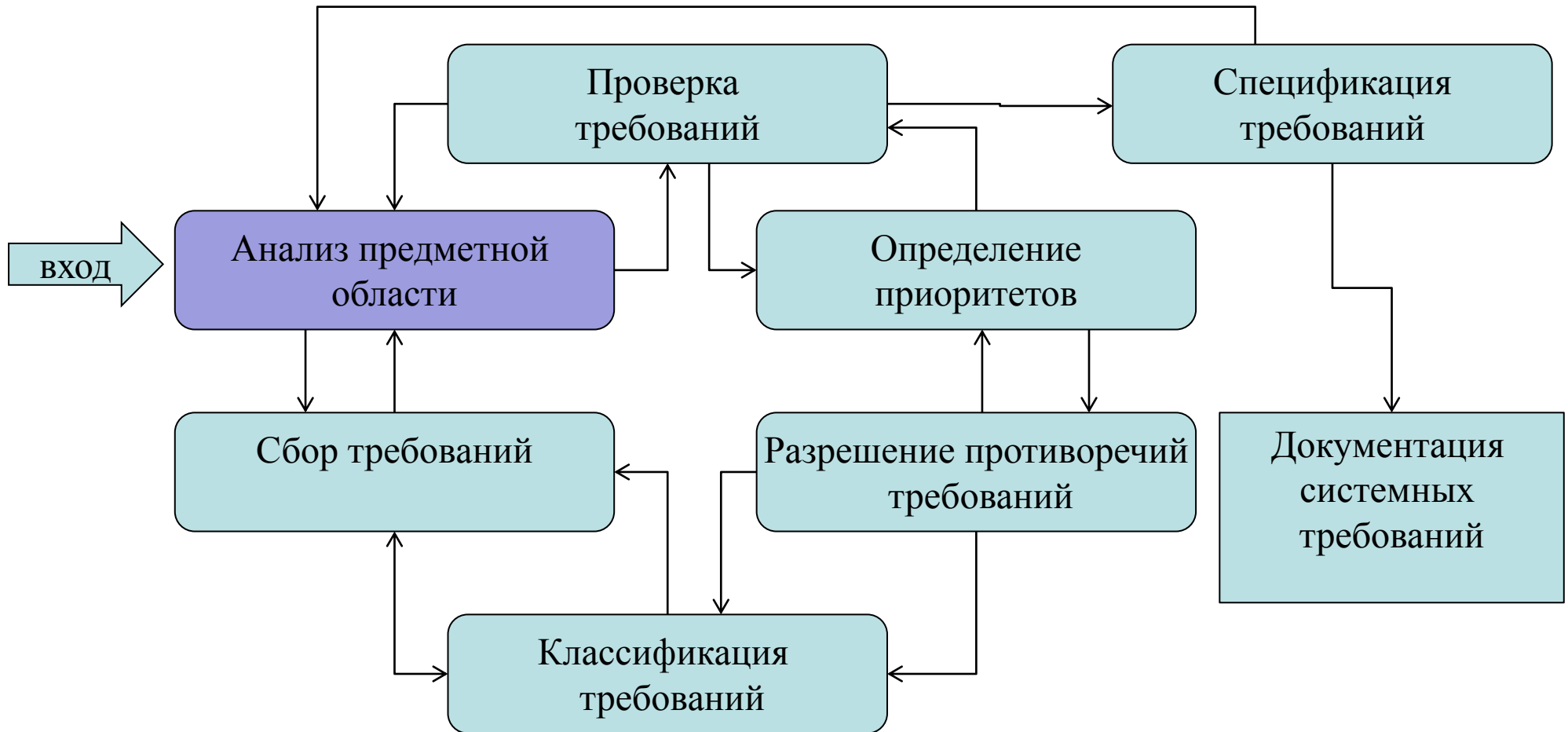
Процесс выделения требований



**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*



Процесс выделения требований



**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*

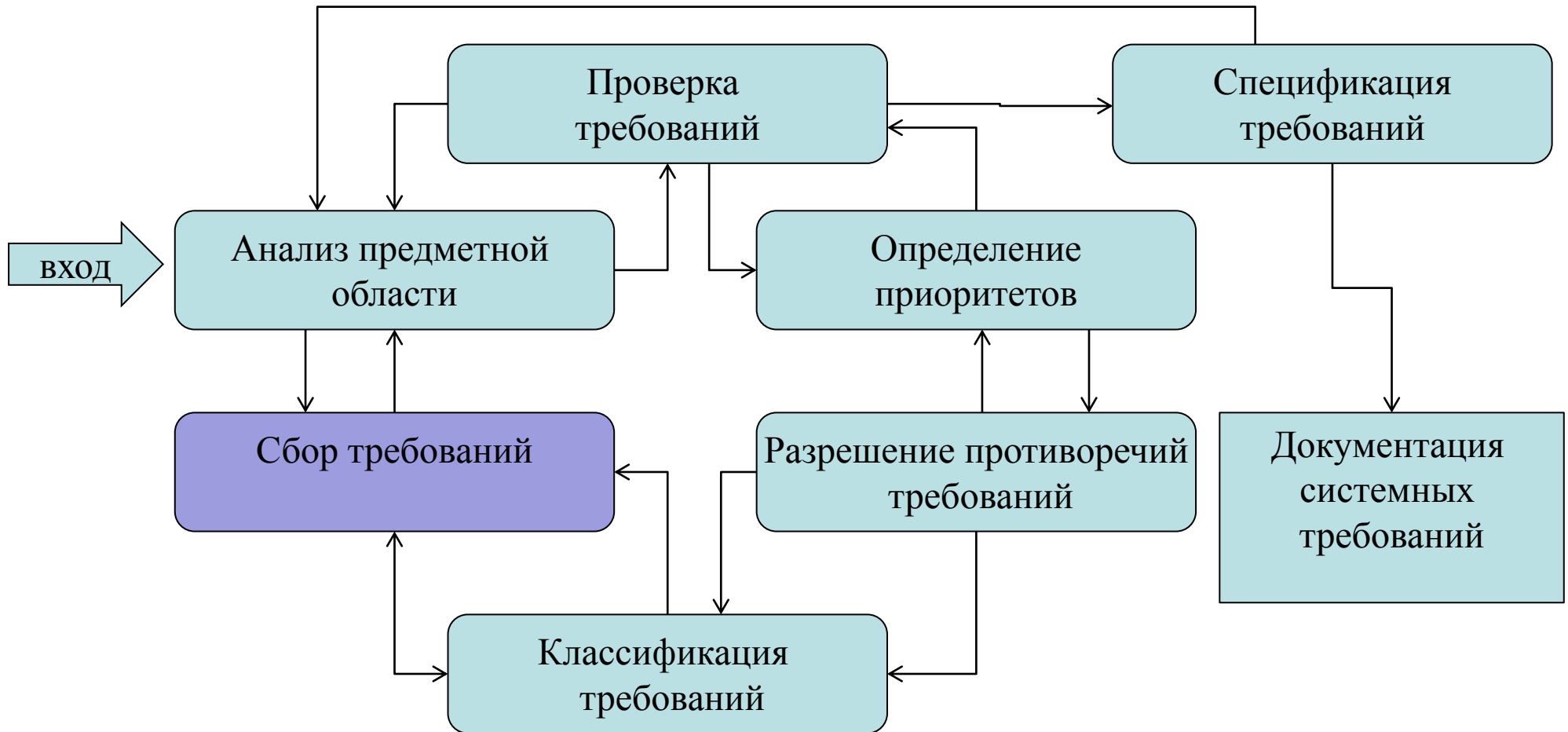


Анализ предметной области

- ❑ Аналитики должны изучить предметную область, где будет использоваться разрабатываемая система.

- ❑ Источники информации:
 - Рабочий персонал.
 - Менеджеры.
 - Наблюдение за персоналом.
 - Документация, используемая при работе персонала.

Процесс выделения требований



**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*

Сбор требований

□ Источники требований:

– Основные источники требований:

- Заказчики.
- Пользователи.
- Инженеры и администраторы, использующие систему.
- Обслуживающий персонал.
- Эксперты из предметной области.
- Представители группы маркетинга.
- Партнеры.

– Дополнительные источники требований:

- Отраслевые аналитики.
- Информация о конкурентах и предлагаемых ими продуктах.
- Документы компании (например, «книга жалоб и предложений»).

Методы выделения требований...

- Методы выделения требований:
 - «Мозговой» штурм.
 - Интервьюирование пользователей.
 - Погружение в среду пользователя.
 - Изучение аналогов.
 - Изучение усовершенствований, сделанных пользователями.
 - Совместный семинар.
 - Демонстрация прототипа заинтересованным лицам.

Методы выделения требований...

❑ *«Мозговой» шторм:*

- Приглашаются члены команды, заказчик и все заинтересованные стороны.
- Важно: **запрещена любая критика.**
- Время ограничено несколькими часами.
- Цель – рассмотреть систему с разных точек зрения использования.

❑ *Интервьюирование пользователей:*

- Цель – понять, что **реально** необходимо пользователю.

❑ *Погружение в среду пользователя:*

- Работа в среде, где будет установлена система, позволяет **лучше понять** нужды клиента.



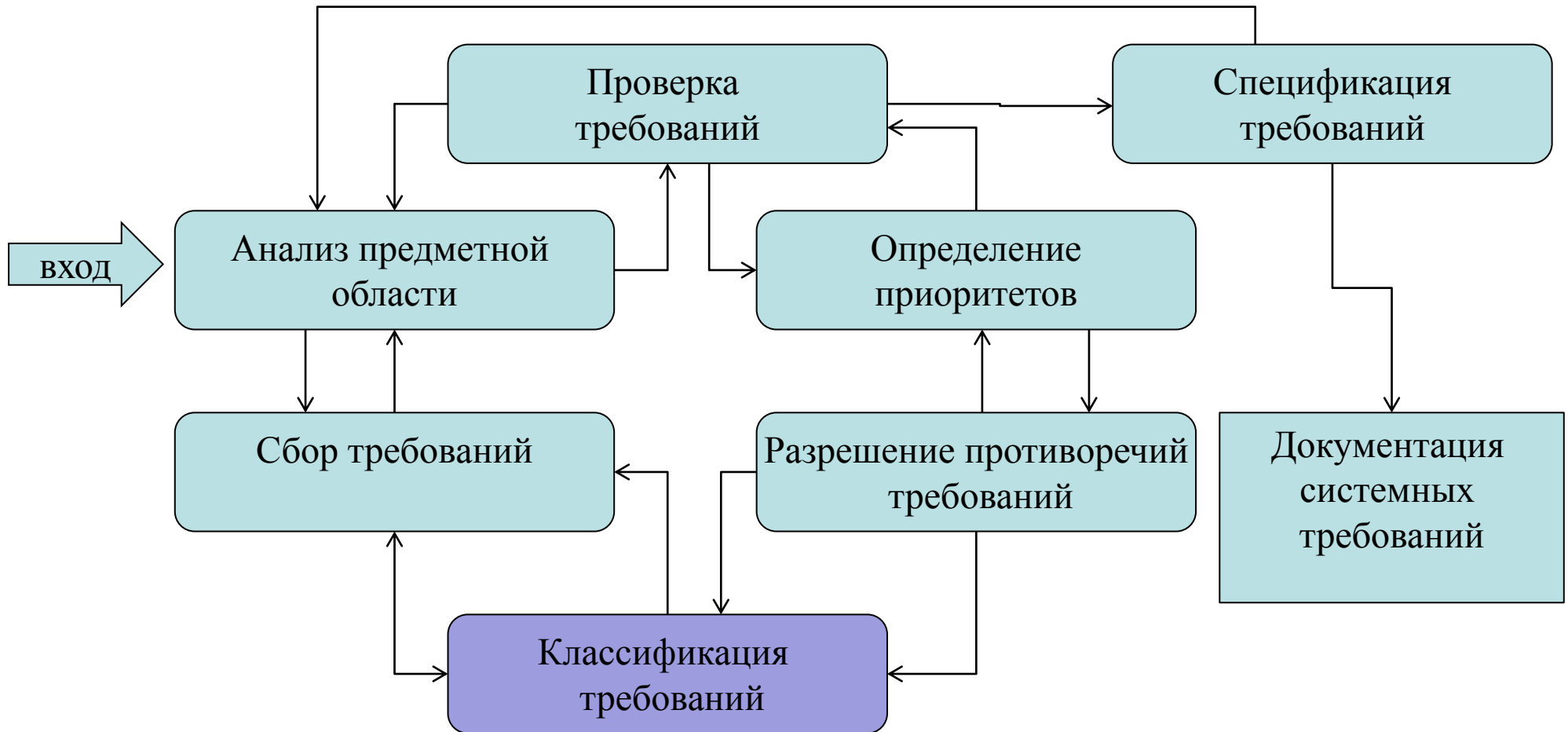
Методы выделения требований...

- ❑ *Изучение усовершенствований, сделанных пользователями:*
 - Пользователи часто **сами предлагают хорошие решения.**
- ❑ *Изучение аналогов:*
 - Как правило, заказчик в своих желаниях опирается на **ранее увиденные технологии.**
 - Аналоги могут содержать **хорошие идеи.**
 - Даже если аналогичные продукты решают другие проблемы, аналоги могут содержать **путь решения.**

Методы выделения требований

- ***Совместные семинары и демонстрация прототипа заинтересованным лицам:***
 - Семинары и демонстрации позволяют на ранних этапах выявить неправильно интерпретированные требования.

Процесс выделения требований



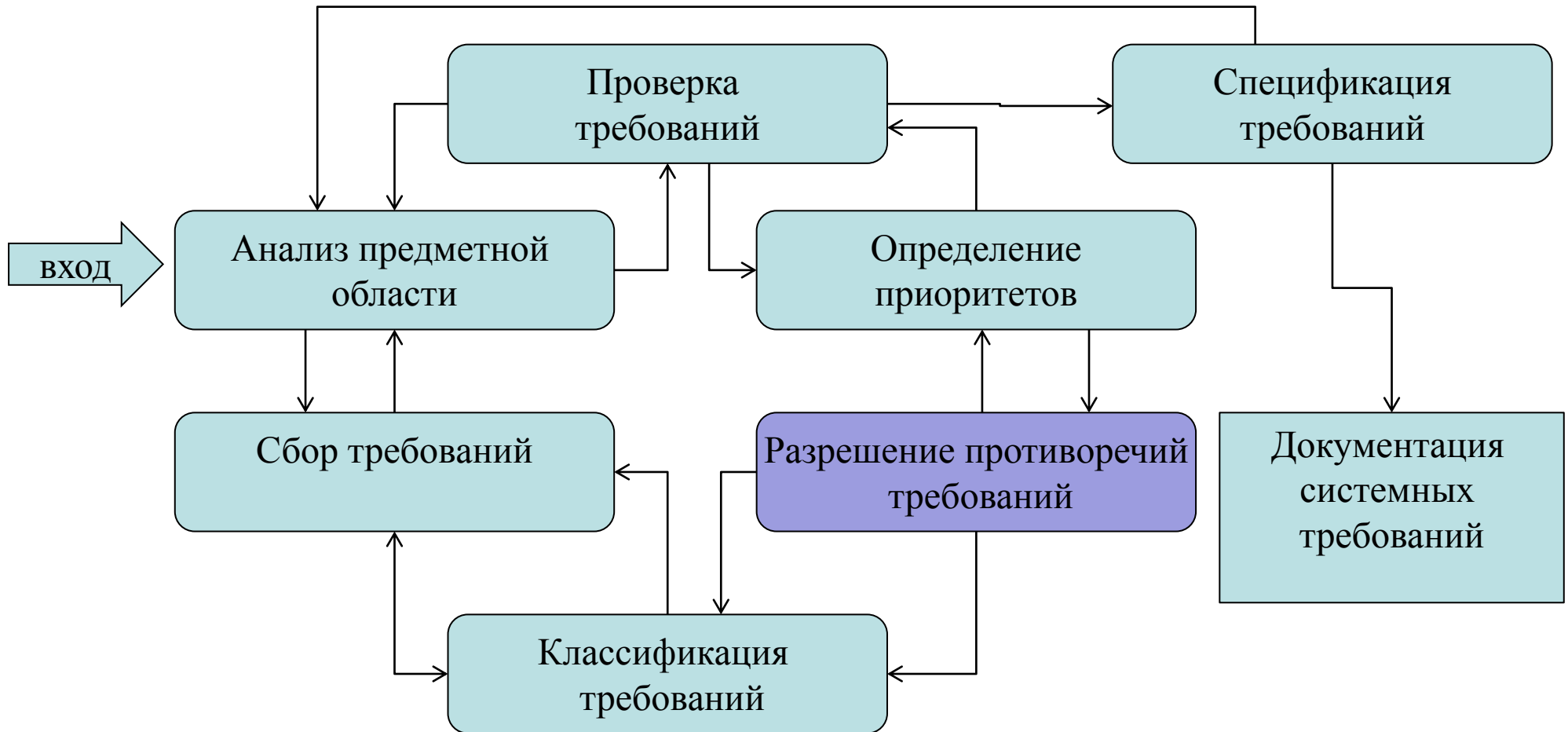
**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*



Классификация требований

- ❑ Разрозненные требования пользователей анализируются и преобразуются в набор связных требований.
- ❑ Классифицированные требования являются набором требований, которые являются кандидатами на добавление в систему.

Процесс выделения требований

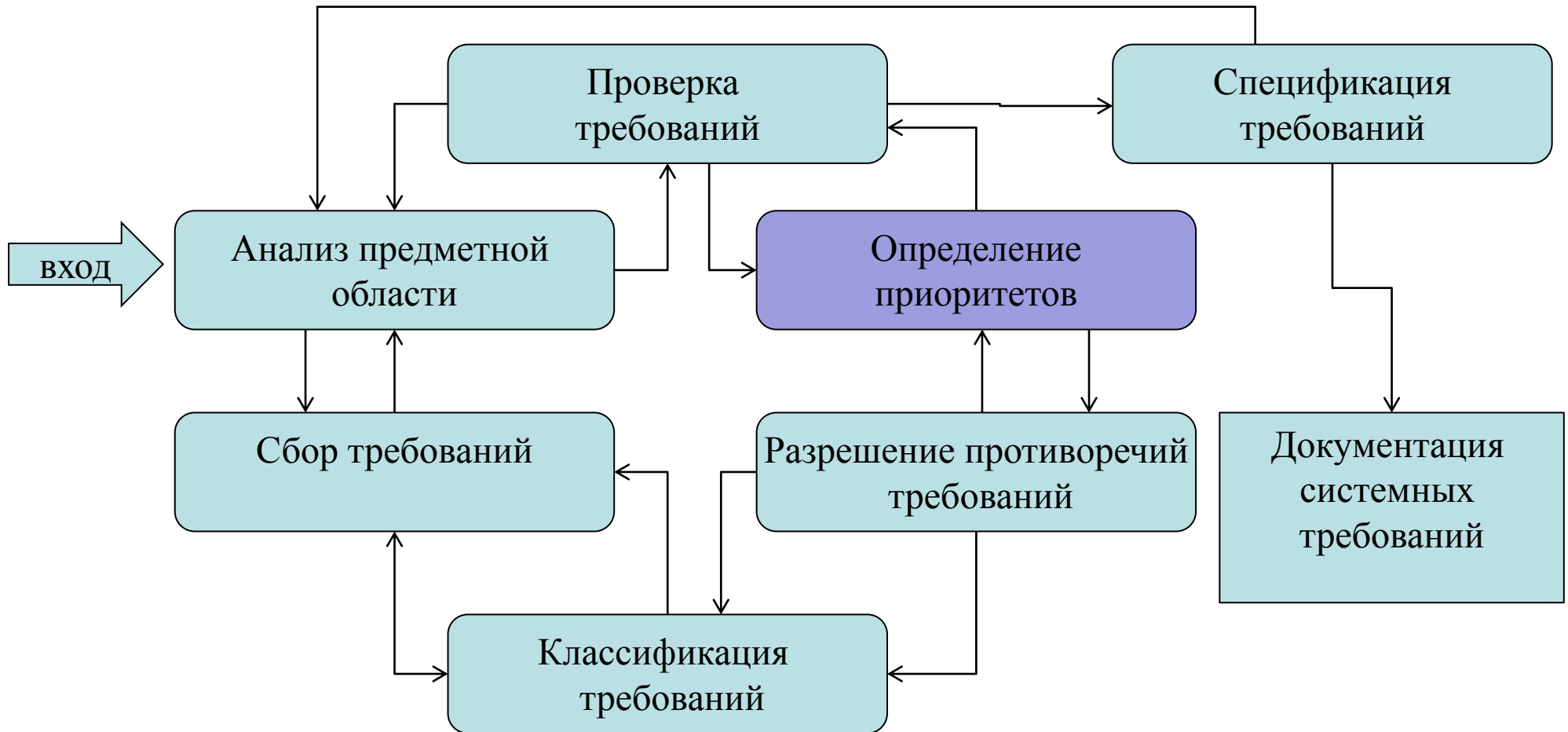


**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*

Разрешения противоречий

- ❑ Выделяемые требования из разных источников могут противоречит друг другу.
- ❑ На этапе необходимо разрешить противоречия.
- ❑ Для поиска противоречий может помочь трассировка требований.

Процесс выделения требований

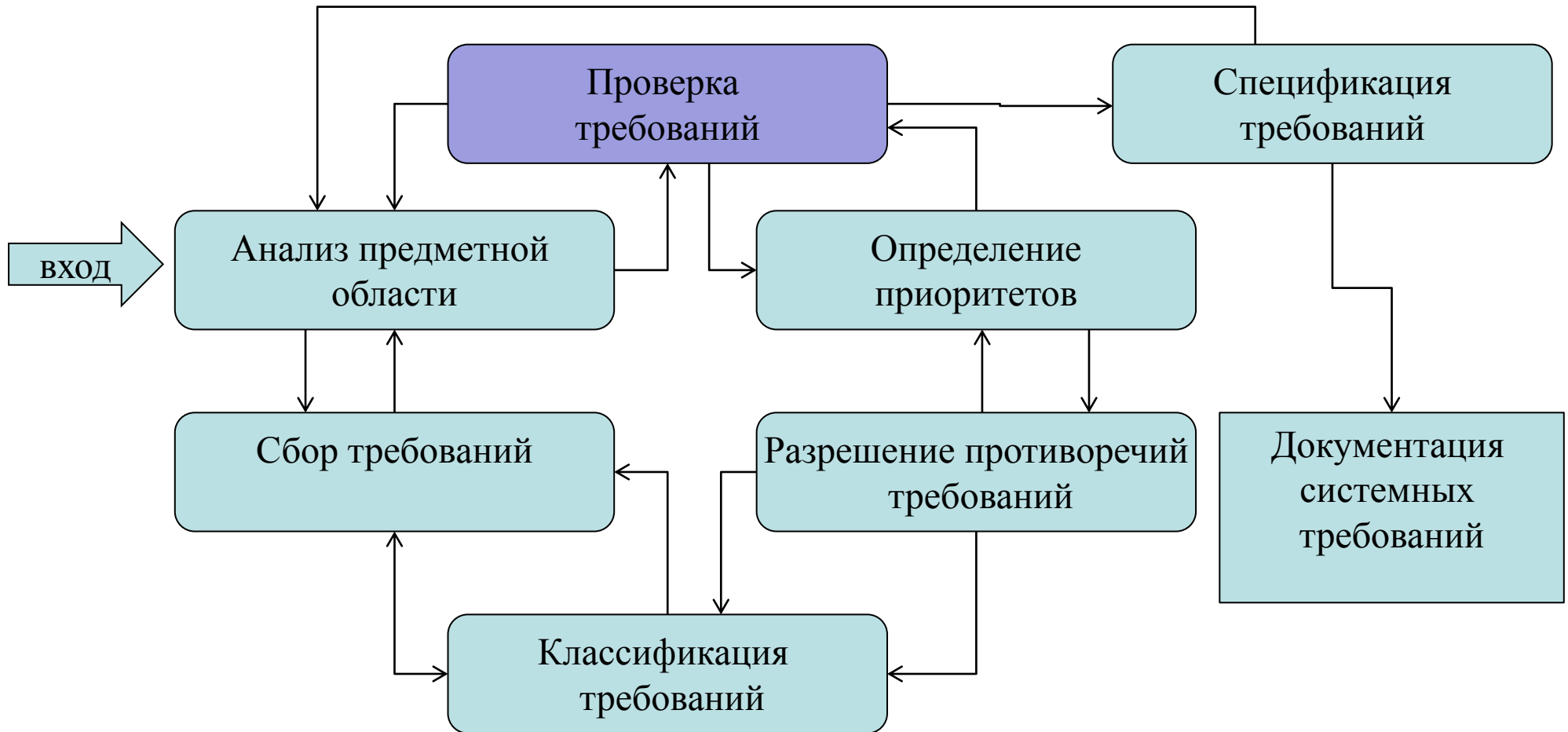


**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*

Определение приоритетов

- ❑ Требования могут обладать разными приоритетами для заказчика.
- ❑ Требования следует описывать все, но некоторые требования могут быть отнесены за границы системы.

Процесс выделения требований



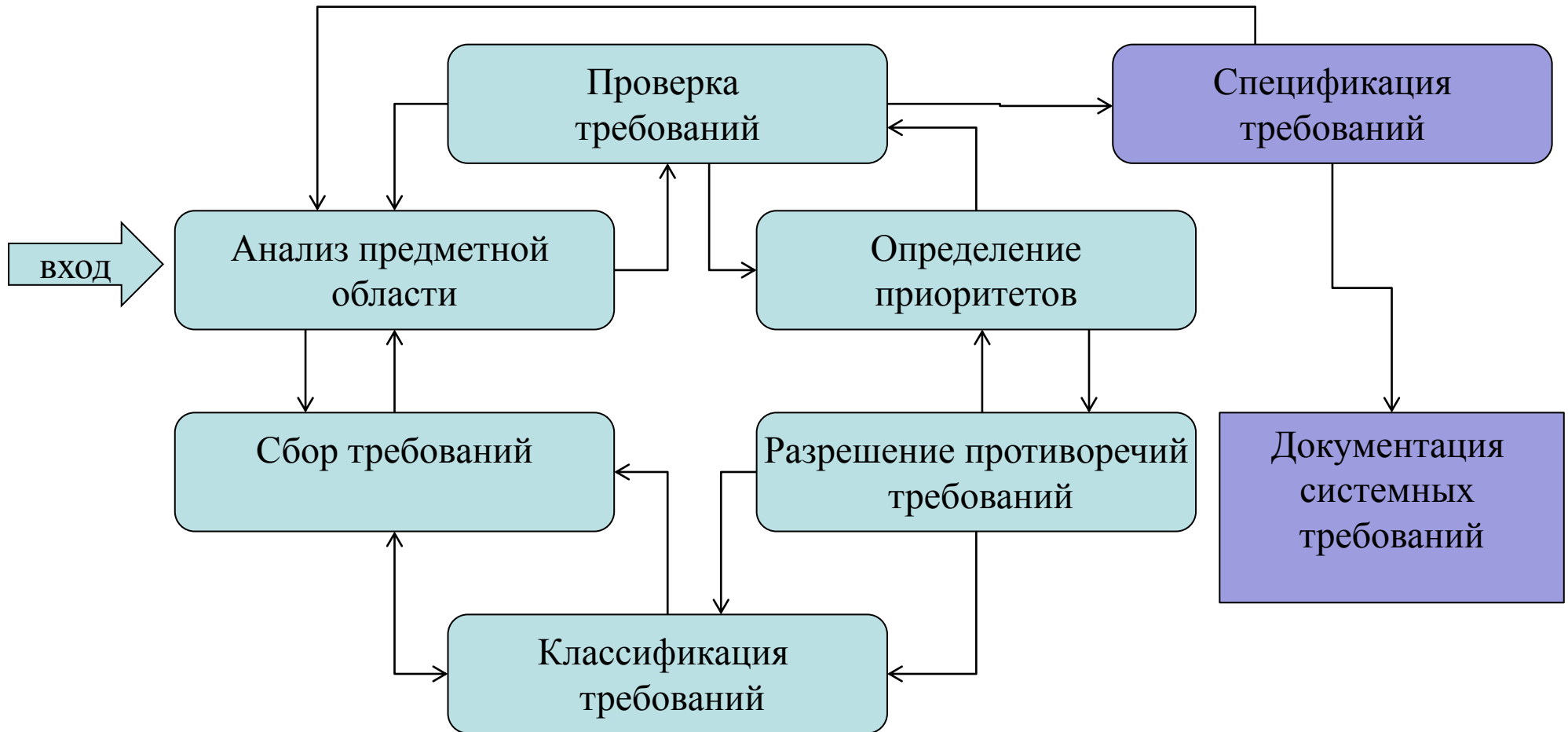
**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*



Проверка требований

- ❑ На этапе проверки требования проверяются на выполнение основных свойств требований.

Процесс выделения требований



**Соммервиль И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 624 с.*

Способы описания требований...

- ❑ Основной способ описания требований – **текстовый документ**.
- ❑ **Состав документа:**
 1. Цель разработки системы – 1 абзац.
 2. Границы системы – кратко о той функциональности, которую необходимо реализовать, а какую реализовывать не нужно.
 3. Определения и сокращения, используемые в документе.
 4. Глоссарий.
 5. Ссылки на другие документы проекта.
 6. Полезно иметь обзор содержания документа.
 7. Сами требования, разделенные по категориям.



Способы описания требований...

- ❑ Пример категорий, по которым можно разделить требования:
 - Функциональность.
 - Удобство использования системы.
 - Надежность системы.
 - Производительность системы.

- ❑ Замечание: список категорий может меняться.

Способы описания требований...

□ Описание требований

- При описании требований необходимо руководствоваться свойствами, которые должны быть присущи требованиям.
- Для возможности анализа требований рекомендуется требованиям добавлять атрибуты.
- Пример атрибутов:
 - Идентификатор требования.
 - Принадлежность к группам требований.
 - Описание требования.
 - Приоритет (значение, например, от 1 до 10).
 - Степень риска (значение, например, от 1 до 10).
 - Источник требования (автор).
 - Связанные требования.



Способы описания требований

- Для чего нужны атрибуты?
 - Фильтрация требований.
 - Выделение ключевых требований.
 - Анализ непротиворечивости требований при добавлении / изменении.
 - Атрибуты позволяют разделить требования на более простые, сократив объем документа, и упростить понимание проблемы (появляется иерархичность представления знаний о проблеме).

Пример описания требования...

- ❑ Рассмотрим пример требования и его описания:
 - Требование с точки зрения пользователя, полученное после интервью.

Пример:

При заданных городах отправления и прибытия, а также при наличии соединения с сервером с пропускной способностью канала не менее 5 Мб/с система должна обеспечить обнаружение рейса или установить факт его отсутствия за время, не превышающее одной минуты при количестве одновременно работающих пользователей, не превышающем 100. Хотелось бы, чтобы при вводе городов было реализовано автозаполнение полей. Если поле с городом отправления или прибытия заполнено не верно необходимо выдать сообщение об ошибке.

- ❑ Анализировать такое требование не представляется возможным. Далее показано, как данное требование представить в иерархическом виде.

Функциональность

Requirement ID	F001
Group	Главная функциональность
Description	Необходимо реализовать поиск рейсов при корректно введенных городах отправления и прибытия. При возникновении ошибки пользователю необходимо выдать окно с диагностикой. ошибки.
Priority	1
Source	Иванов И.И.
Risk	1
References	F002;F003;U001;R001; R002;P001;P002

Requirement ID	F002
Group	Главная функциональность / Ввод данных
Description	Необходимо реализовать возможность ввода названия городов с проверкой корректности ввода.
Priority	1
Source	Иванов И.И.
Risk	1
References	F003

Requirement ID	F003
Group	Главная функциональность
Description	Реализовать возможность подключения к серверу для получения списка производимых рейсов.
Priority	1
Source	Иванов И.И.
Risk	1
References	P001



Надежность

Requirement ID	R001	
Group	Ввод данных	
Description	Если название города не существует, вывести диагностику и создать исключение NoCity.	
Priority		4
Source	Сидоров С.С.	
Risk		7
References		

Requirement ID	R002	
Group	Ввод данных	
Description	Если отсутствует соединение с сервером, вывести диагностику и создать исключение NoConnect.	
Priority		4
Source	Сидоров С.С.	
Risk		7
References		

Производительность

Requirement ID P001	
Group	Получение данных
Description	При наличии соединения с сервером с пропускной способностью канала не менее 5 Мб/с и количестве одновременных пользователей меньше 100 необходимо возвращать информацию о доступных рейсах за время, не превышающее 10 сек.
Priority	3
Source	Сидоров С.С.
Risk	2
References	R002

Requirement ID P002	
Group	Получение данных
Description	Поиск рейса необходимо выполнять менее, чем за 1 мин.
Priority	5
Source	Сидоров С.С.
Risk	4
References	P001

Удобство использования

Requirement ID	U001	
Group	Ввод данных	
Description	Реализовать автозаполнение названий городов при вводе.	
Priority		6
Source	Петров А.В.	
Risk		10
References		

ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ТРЕБОВАНИЙ



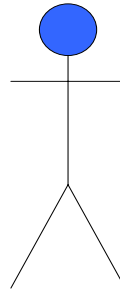
Системное моделирование

- ❑ Требования позволяют точно описать, что необходимо сделать.
- ❑ Если требований много, то понять все требования в совокупности сложно.
- ❑ Для упрощения понимания и общения используется моделирование (в частности, визуальное).
- ❑ Для моделирования могут применяться, например:
 - Диаграмма прецедентов (сценариев) использования (Use-case diagram).
 - Диаграмма деятельности (Activity diagram).
 - Диаграмма последовательности (Sequence diagram).

Диаграмма прецедентов использования...

□ **Диаграмма прецедентов использования** содержит два элемента:

- **Актёр в UML** – человек, машина или программа, воздействует на систему, является внешним по отношению к ней.



- **Прецедент использования в UML** – описание последовательности действий – (часто прецедент связан с диаграммой деятельности).

UseCase

Диаграмма прецедентов использования...

□ Связи:

- Актеры и варианты использования общаются посредством посылки сообщений.
- Сообщения могут идти в обе стороны.
- Стрелка показывает инициатора общения (актер на рисунке) и может быть опущена.

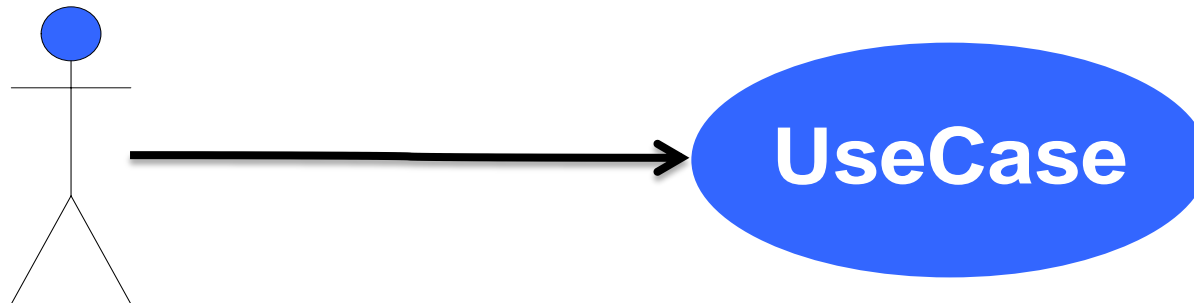


Диаграмма прецедентов использования...

- ❑ Диаграмма прецедентов использования не должна содержать прецедентов, не имеющих отношения к разрабатываемой системе.

Пример:

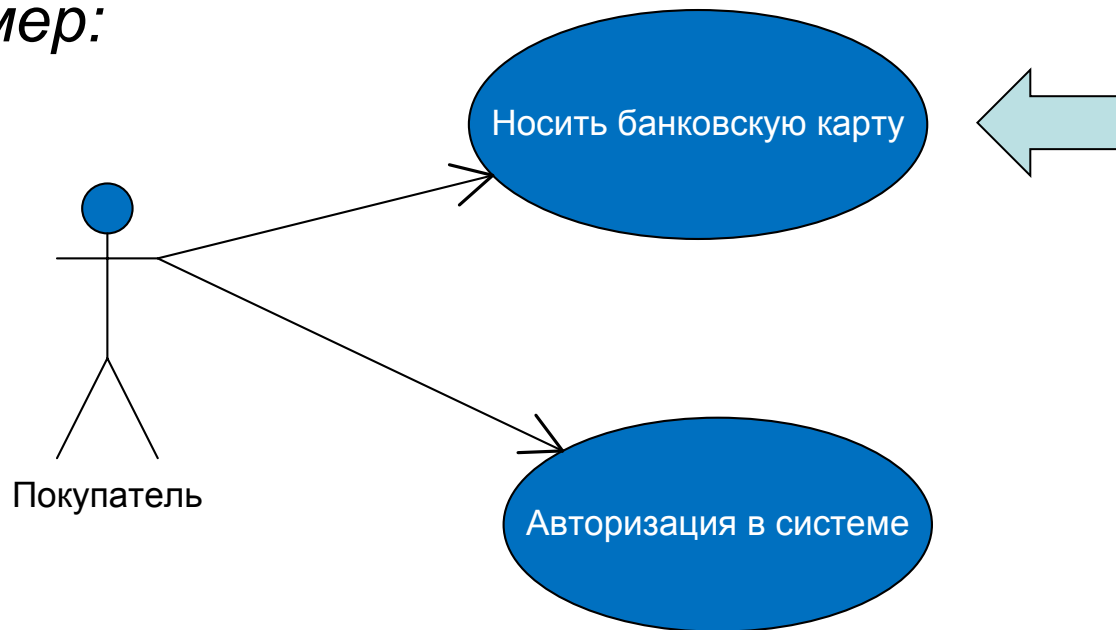


Диаграмма прецедентов использования...

- ❑ Диаграмма прецедентов использования не должна содержать «висячих» прецедентов.

Пример:

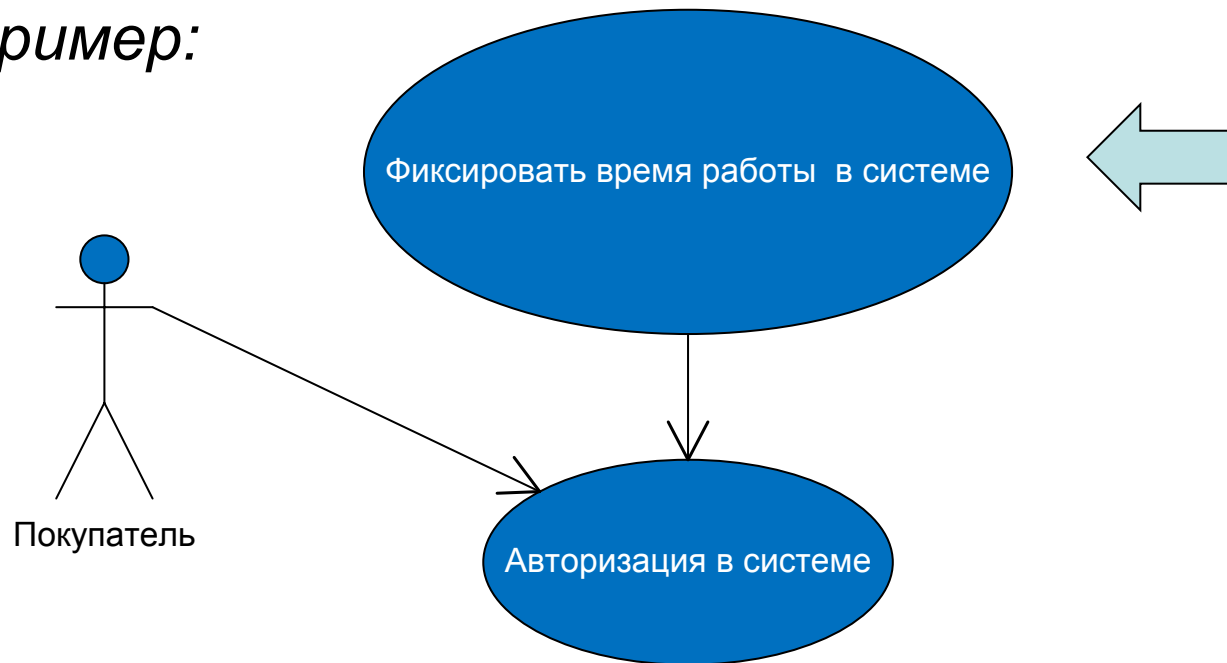


Диаграмма прецедентов использования...

- ❑ Диаграмма прецедентов использования не должна отображать последовательность действий.

Пример:

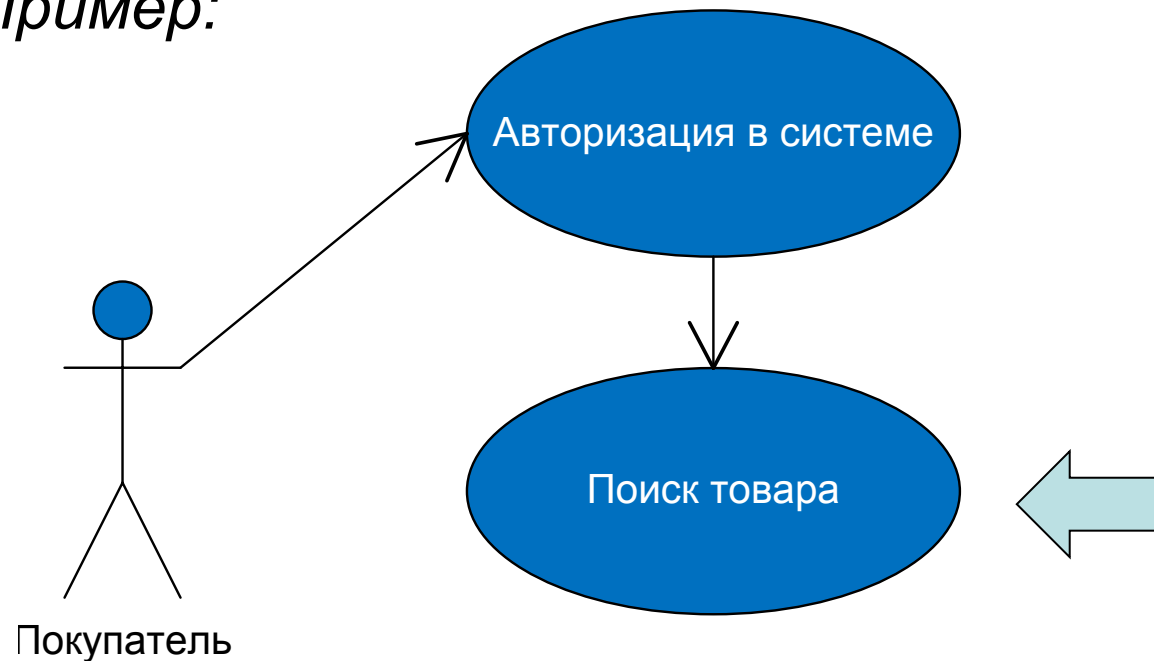


Диаграмма прецедентов использования...

- ❑ Диаграмма прецедентов использования не должна содержать актеров, дублирующих роли.

Пример:

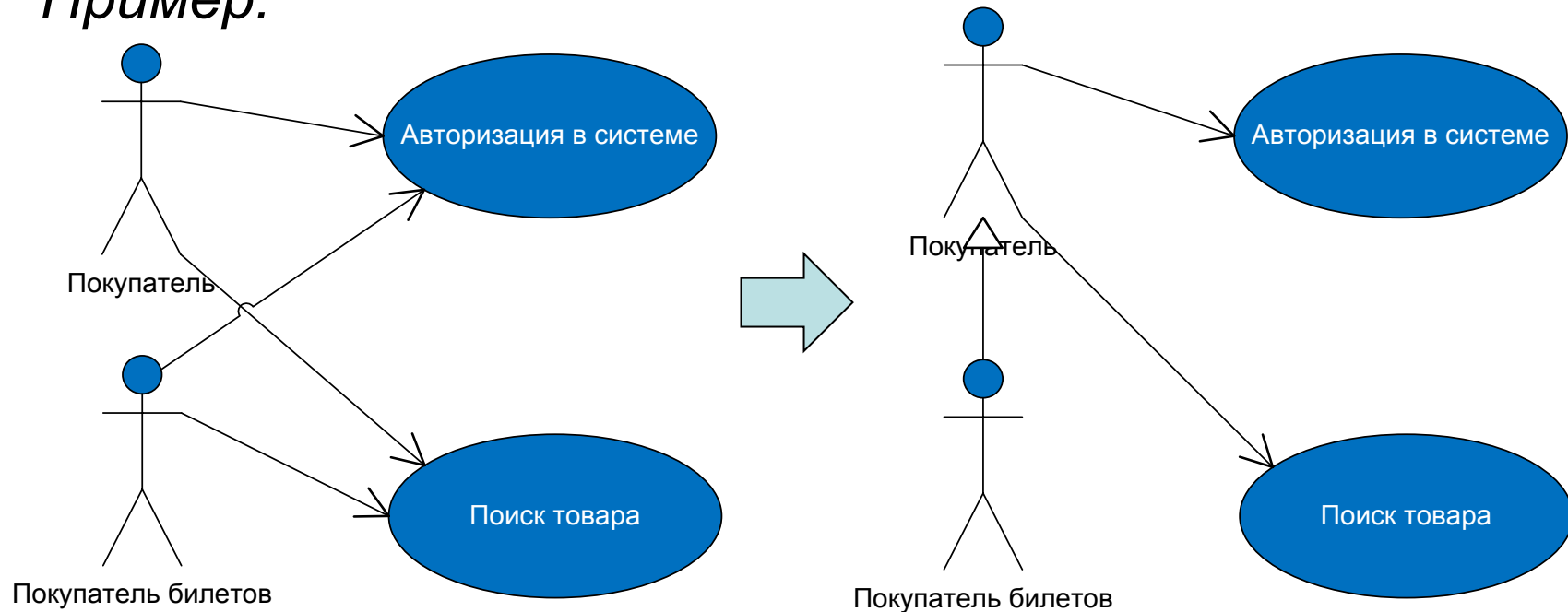


Диаграмма прецедентов использования

- Способ получения (например, через «мозговой штурм»):
 1. Выделяем как можно больше участвующих сторон – актеры.
 2. Выделяем прецеденты и соединяем их с актерами.
 3. Для каждого из прецедентов определяем, нужна ли связь с другими актерами или прецедентами.
 4. Описываем по абзацу каждого актера и каждый прецедент или выставляем связи с существующими требованиями.
 5. Дополняем глоссарий новыми определениями (если есть).
 6. Проверяем прецеденты на удовлетворение ранее описанным требованиям.
 7. Выделяем наиболее важную функциональность.



Диаграмма деятельности ...

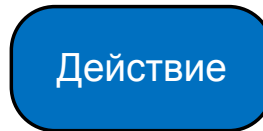
❑ Диаграмма деятельности отображает последовательность действий.

❑ Основные элементы:

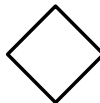
– Начало действий



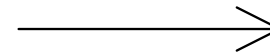
– Действие



– Условие



– Передача потока управления



– Конец действий



Диаграмма деятельности ...

Пример:

При заданных городах отправления и прибытия, а так же при наличии соединения с сервером с пропускной способностью канала не менее 5 Мб/с система должна обеспечивать возможность обнаружения рейса или установить факт его отсутствия за время, не превышающее одной минуты при количестве одновременно работающих пользователей, не превышающем 100. Хотелось бы, чтобы при вводе городов было реализовано автозаполнение полей. Если поле с городом отправления или прибытия заполнено не верно, необходимо выдать сообщение об ошибке.

Диаграмма деятельности ...

- Диаграмма прецедентов использования:

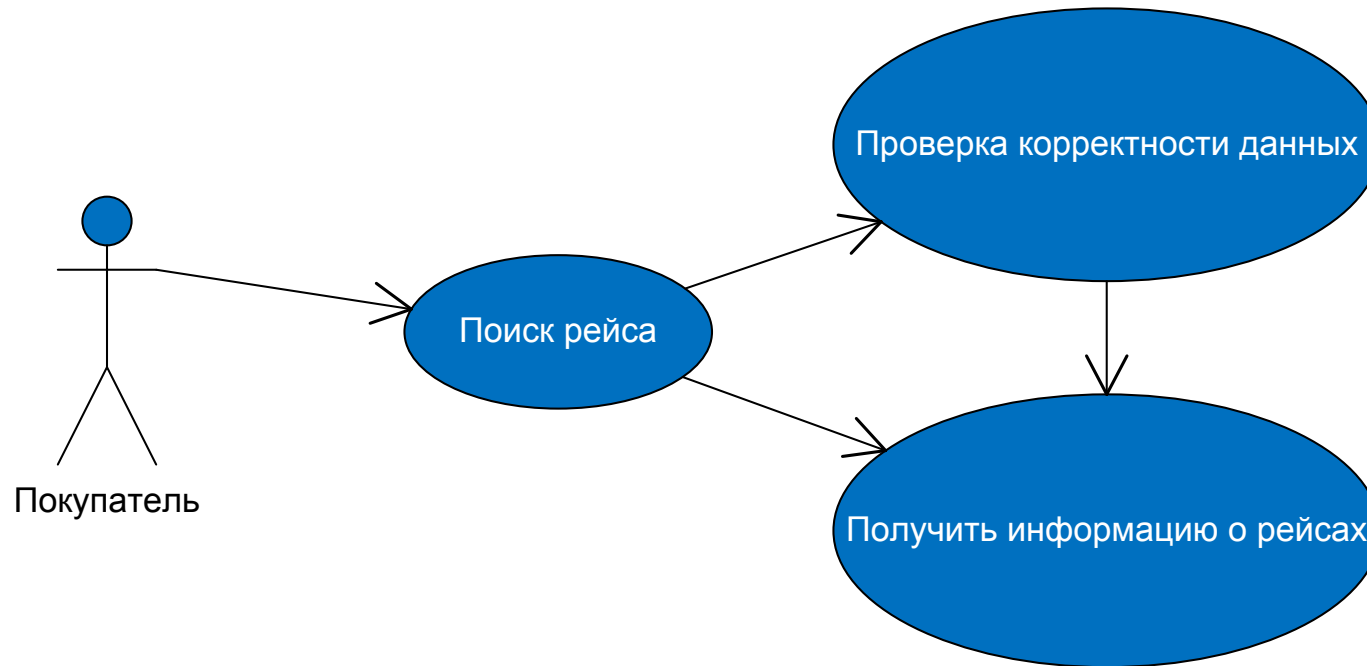


Диаграмма деятельности

□ Пример диаграммы:

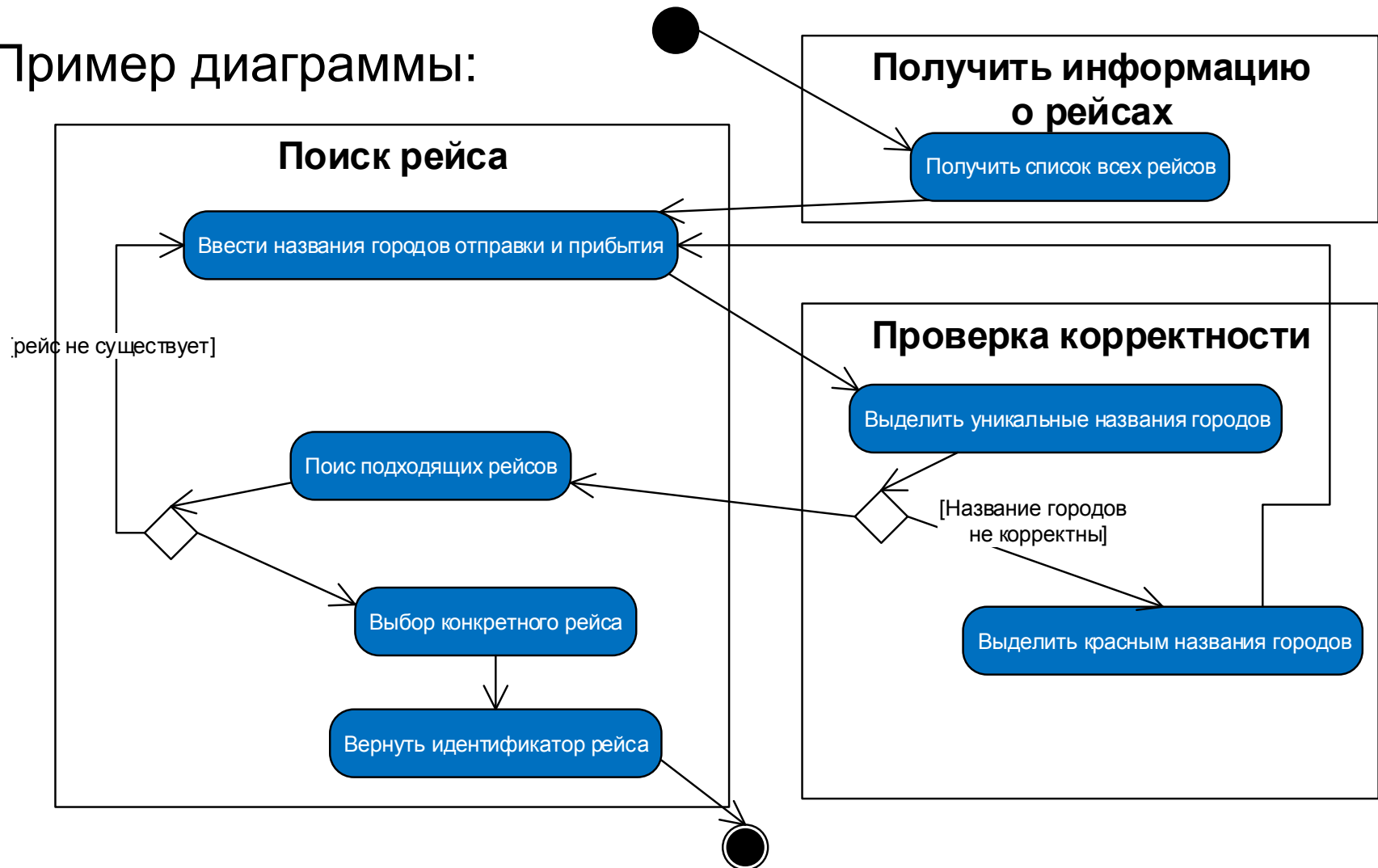


Диаграмма последовательности...

- ❑ Диаграмма последовательности представляет активности в системе с точки зрения объектов.
- ❑ Основные элементы:
 - Объект
 - Линия жизни объекта
 - Работа метода (активность)
 - Вызов метода

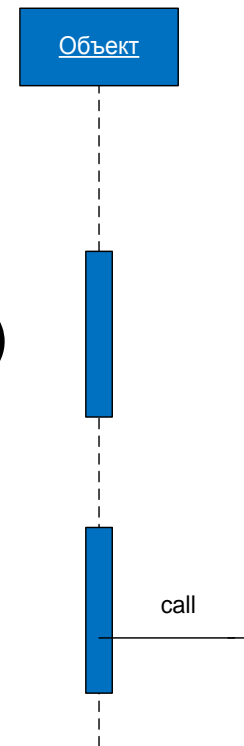
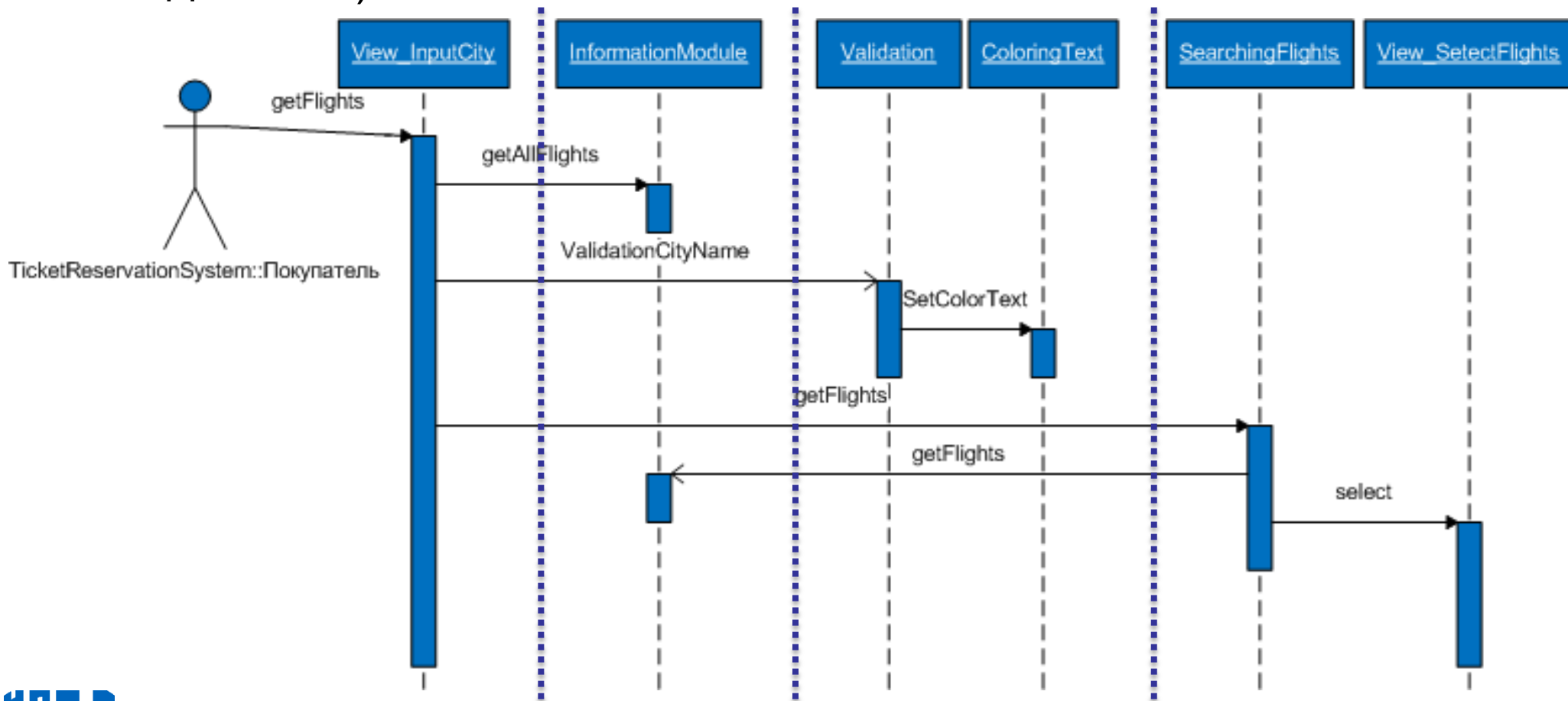


Диаграмма последовательности

- Пример диаграммы (синие штрихи – условные границы подсистем):



Трассировка требований...

- Трассировка необходима (обсуждалось ранее).
 - В некоторых инструментах вводятся специальные диаграммы, позволяющие отображать связи между требованиями и моделями.
 - Например, связь прецедента и требований может быть представлена следующим образом:



- Похожие связи можно отслеживать и на других диаграммах.

Трассировка требований

- ❑ Визуальное представление связей требований и конкретных элементов модели помогает на разных уровнях моделирования системы отслеживать статус разработки системы.
- ❑ Отслеживать разные показатели такие как:
 - Покрытие требований.
 - Достижение вех (ключевых точек проекта).
 - ...

Заключение

- ❑ Рассмотрены основные понятия, связанные с требованиями.
- ❑ Изучены основы выделения требований, рассмотрен пример документирования требований.
- ❑ Рассмотрены элементы системного моделирования, необходимые для представления требований в виде визуальных диаграмм и последующего анализа.

Контрольные вопросы

- ❑ Что такое требование с точки зрения разработки программного обеспечения?
- ❑ Какими свойствами должно обладать требования?
- ❑ Какие источники информации могут быть использованы для получения требований?
- ❑ Какие способы представления требований Вы знаете?
- ❑ Какие диаграммы могут помочь в систематизации требований?
- ❑ Для каких целей используется диаграмма прецедентов?
- ❑ Для каких целей используется диаграмма деятельности?
- ❑ Для каких целей используется диаграмма последовательности?

Литература к лекции

1. **Курс** «Анализ требований к автоматизированным информационным системам» [<http://www.intuit.ru/studies/courses/2188/174/info>].
2. **Соммервиль И.** Инженерия программного обеспечения. – М.: Издательский дом "Вильямс", 2002. – 624 с.
3. **Халл Э., Джексон К., Дик Дж.** Разработка и управление требованиями. – Gray Publishing, USA , 2005г. – 229 с.
4. **Кролл П., Крачтен Ф.** Rational Unified Process – это легко. Руководство по RUP для практиков. – Изд-во “КУДИЦ-Образ”, 2004.