Лекция 14
БГТУ, ФИТ, 1 семестр

Основы программной инженерии (ПИ)

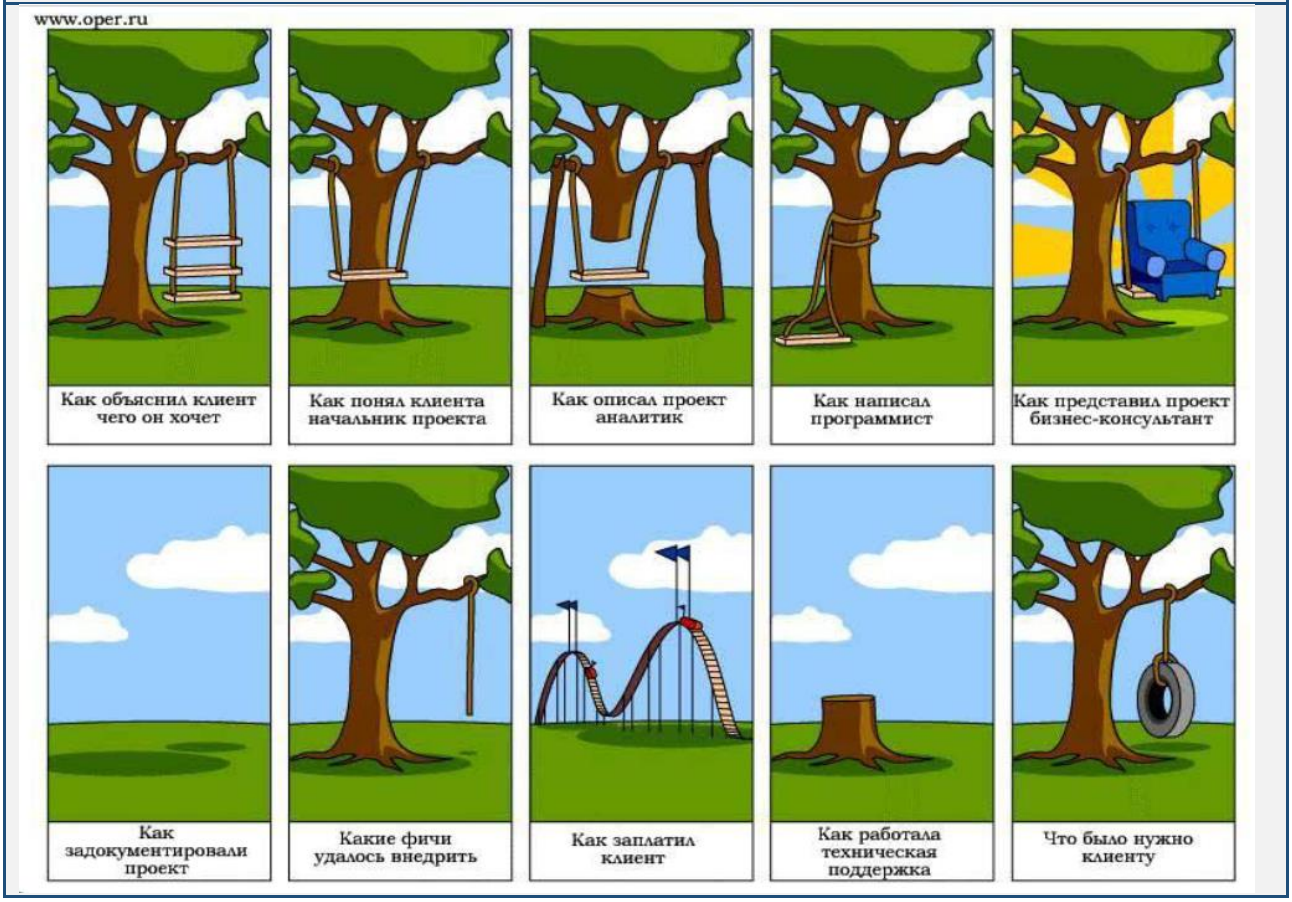
Технологии разработки ПО. Формализация функциональных требований

План лекции:
- назначение диаграммы вариантов использования;
- компоненты диаграммы вариантов использования;
- примеры.

На прошлой лекции:

1. Инженерия требований

Требование – это утверждение, которое идентифицирует эксплуатационные, функциональные параметры, характеристики или ограничения проектирования продукта или процесса, которое ***однозначно***, ***проверяемо*** и ***измеримо***.



Требование –

- ✓ ***условие или возможность, необходимые пользователю для решения его задач или достижения цели (1)***
- ✓ ***условие или возможность, которым должна отвечать или которыми должна обладать система или ее компонента, чтобы удовлетворить контракт, стандарт, спецификацию или иной формальный документ (2)***
- ✓ ***документированное представление условия или возможности, указанное в (1) или (2)***

Цели разработки требований

- ✓ обеспечение наиболее полного и точного отражения условий или возможностей, необходимых заказчику для решения его проблем и достижения бизнес-целей;
- ✓ снижение затрат на разработку, обслуживание и поддержку сложного программного обеспечения.

Вне зависимости от применяемой методологии разработки первым этапом разработки является формулировка требований к продукту.

Набор требований к продукту представляет собой техническое задание, при этом требования делятся на ***функциональные*** (то, что система позволяет сделать, желаемая функциональность) и ***нефункциональные*** (требования к оборудованию, операционной системе и т.п.).

2. Диаграмма вариантов использования

Диаграмма вариантов использования (англ. use-case diagram) –

диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Диаграмма вариантов использования* = *Диаграмма прецедентов

Диаграммы вариантов использования

- ✓ показывают взаимодействия между ***вариантами использования*** и ***действующими лицами***, отражая функциональные требования к системе с точки зрения ***пользователя***.
- ✓ являются исходной концептуальной моделью системы в процессе ее проектирования и разработки.

В Microsoft Visio 2016 использовать набор инструментов – «Схема вариантов использования»

3. Цели построения

Цели построения	1) определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования; 2) сформулировать общие требования к функциональному проектированию системы; 3) разработать исходную концептуальную модель системы для ее последующей реализации; 4) документировать функциональные требования в общем виде для взаимодействия разработчика системы с ее заказчиком и пользователями.

4. Достоинства модели вариантов использования

Достоинства модели вариантов использования	– определяет пользователей и границы системы – определяет системный интерфейс; – удобна для общения пользователей с разработчиками; – используется для написания тестов; – является основой для пользовательской документации; – хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).

5. Суть диаграммы вариантов использования

Диаграмма вариантов использования

позволяет наглядно представить ожидаемое поведение системы.

Основными понятиями диаграмм вариантов использования являются: действующее лицо, вариант использования и связь.

Основные понятия	✓ ***действующее лицо;*** ✓ ***вариант использования;*** ✓ ***связь***: • ассоциация; • отношение расширения; • отношение включения; • отношение обобщения.

6. Вариант использования

Вариант использования

определяет последовательность действий (сценариев) взаимодействия конкретного актера с проектируемой системой с целью достижения какой-либо цели, значимой для этого актера.

В качестве актера могут выступать не только люди, но и другие системы, устройства и т.п.

Вариант использования = Прецедент =Use Case

Имя ***варианта использования*** начинается с большой буквы и обозначается оборотом глагола или существительного, обозначающего ***действие***

7. Актер

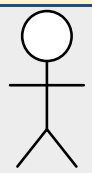
Актер

представляет собой внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей и решения частных задач.

Может рассматриваться как некая роль относительно конкретного варианта использования.

Каждый актер – отдельная роль относительно конкретного варианта использования.

Актер* = Actor = *Действующее лицо* = *Роль

Стандартное графическое изображение актера:	 Студент

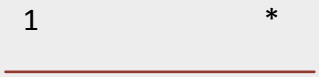
Актер всегда находится вне системы, его *внутренняя структура* никак не воспринимается.

Примеры актеров: студент, преподаватель, клиент банка, банковский служащий, продавец, сотовый телефон, гость.

Имя ***актера*** основано на использовании имени ***существительного***.

8. Отношения

> Один ***актер*** может взаимодействовать с несколькими ***вариантами использования*** и наоборот.
> Два варианта использования, определенные для одной и той же сущности, **не могут** взаимодействовать друг с другом, т.к. любой из них самостоятельно описывает законченный вариант использования этой сущности.

Виды отношений	
ассоциативное отношение (отношение ассоциации, association relationship)	
отношение расширения (extend relationship)	
отношение включения (include relationship)	
отношение обобщения (generalization relationship)	

Отношение ассоциации (association relationship):

отношение между ***вариантом использования*** и ***актером***, отражающее ***связь*** между ними.

Отношения ассоциации отражают возможность использования актером прецедента.

Отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

Обозначение: в виде прямой линии.

Могут быть дополнительные обозначения: кратность связи, направление связи, наименование связи

Пример:

Мощность (кратность, multiplicity) ассоциации определяет количество экземпляров обеих сущностей, которое может участвовать в данной ассоциации. Графически значение мощности отмечается возле линии отношения ассоциации на стороне соответствующей сущности.

В диаграммах вариантов использования определено три типа мощности ассоциации: ***один-к-одному; один-ко-многим; многие-ко-многим***.

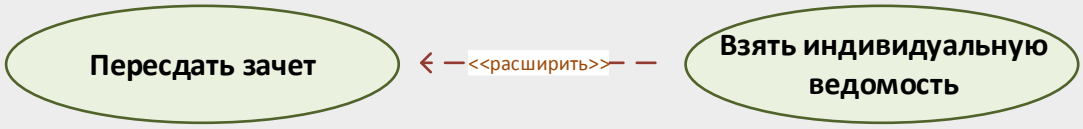
Отношение расширения (extend relationship):

определяет взаимосвязь базового варианта использования с некоторым другим более общим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении некоторых дополнительных условий.

Обозначение: <<расширить>>

Отношение расширения отражает возможное присоединение одного варианта использования к другому в некоторой точке (точке расширения).

Пример:



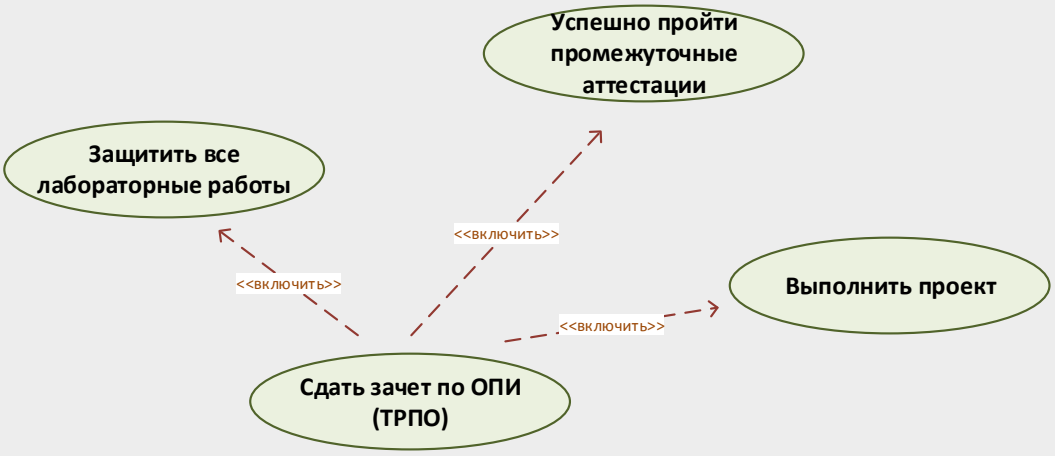
Стрелка указывает на базовый вариант использования!

Отношение включения (include relationship):

указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования.

Обозначение: <<включить>>

Пример:



Отношение обобщения (generalization relationship):

Отношение обобщения служит для указания того, что некоторый вариант использования может быть обобщен до другого варианта использования.

Обозначение: 

Пример:

A

B

B – предок по отношению к A

A – потомок B

Потомок наследует все свойства и поведение своего родителя, может быть дополнен новыми свойствами и особенностями поведения.

Стрелка указывает на родительский вариант использования.

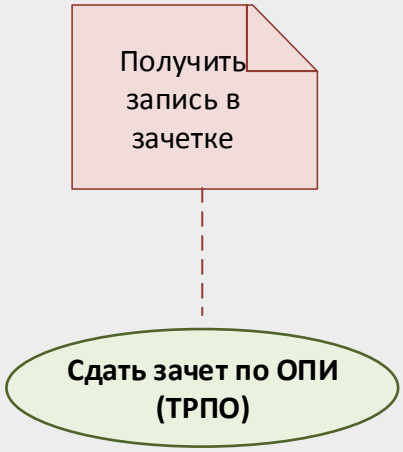
9. Примечание – элемент диаграммы вариантов использования

Примечание:

элемент диаграммы вариантов использования.

Соединяется с элементом ***штриховой линией***

Пример:



Примечание (Note) в языке UML предназначено для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта.

Примечание может относиться к любому элементу диаграммы.

10. Пример

Рассмотрим выполнение 12-ой лабораторной работы студентами.

Моделируемая система – «Ознакомление с технологиями разработки ПО».

Цель – см. цель лабораторной работы 12.

Актер – «Студент» (взаимодействует с системой).

Функциональность:

- «Студент» должен ответить на контрольные вопросы по изученному материалу;
- «Студент» определяет цели и назначение проекта;
- «Студент» выбирает название проекта;
- «Студент» выбирает модули, входящие в состав проекта в соответствии с требованиями лабораторной работы 12;
- «Студент» выполняет индивидуальное задание;
- «Студент» выполняет проектирование и разработку проекта с использованием гибких технологий разработки ПО;
- «Студент» должен выполнить тестирование;
- «Студент» должен выполнить рефакторинг;
- «Студент» должен выполнить документирование;
- «Студент» представляет проект.

При взаимодействии с актером «Студент» система должна позволять выполнять набор функциональных требований, при этом важна реакция системы.

Главная последовательность взаимодействия «Студента» с «Системой»:

Функциональность	Реакция системы
1. «Студент» должен ответить на контрольные вопросы по изученному материалу	2. «Система» обеспечивает «Студенту» возможность изучения теории по теме лабораторной работы
3. «Студент» определяет цели и назначение проекта	4. «Система» должна проконтролировать определение «Студентом» цели и назначения проекта
5. «Студент» называет проект	
6. «Студент» выбирает модули, входящие в состав проекта в соответствии с требованиями лабораторной работы 12	

7. «Студент» выполняет индивидуальное задание	8. «Система» контролирует выполнение «Студентом» индивидуального задания
9. «Студент» выполняет проектирование и разработку проекта с использованием гибких технологий разработки ПО	
10. «Студент» должен выполнить тестирование	
11. «Студент» должен выполнить рефакторинг	
12. «Студент» должен выполнить документирование	
13. «Студент» представляет проект	14. «Система» оценивает презентацию проекта скрам-команды и каждого члена команды - сдача лабораторной работы преподавателю

Альтернативная последовательность

7а: «Студент» **не выполняет** индивидуальное задание.

7в: Реакция системы: «Студент» **не** допускается до экзамена.

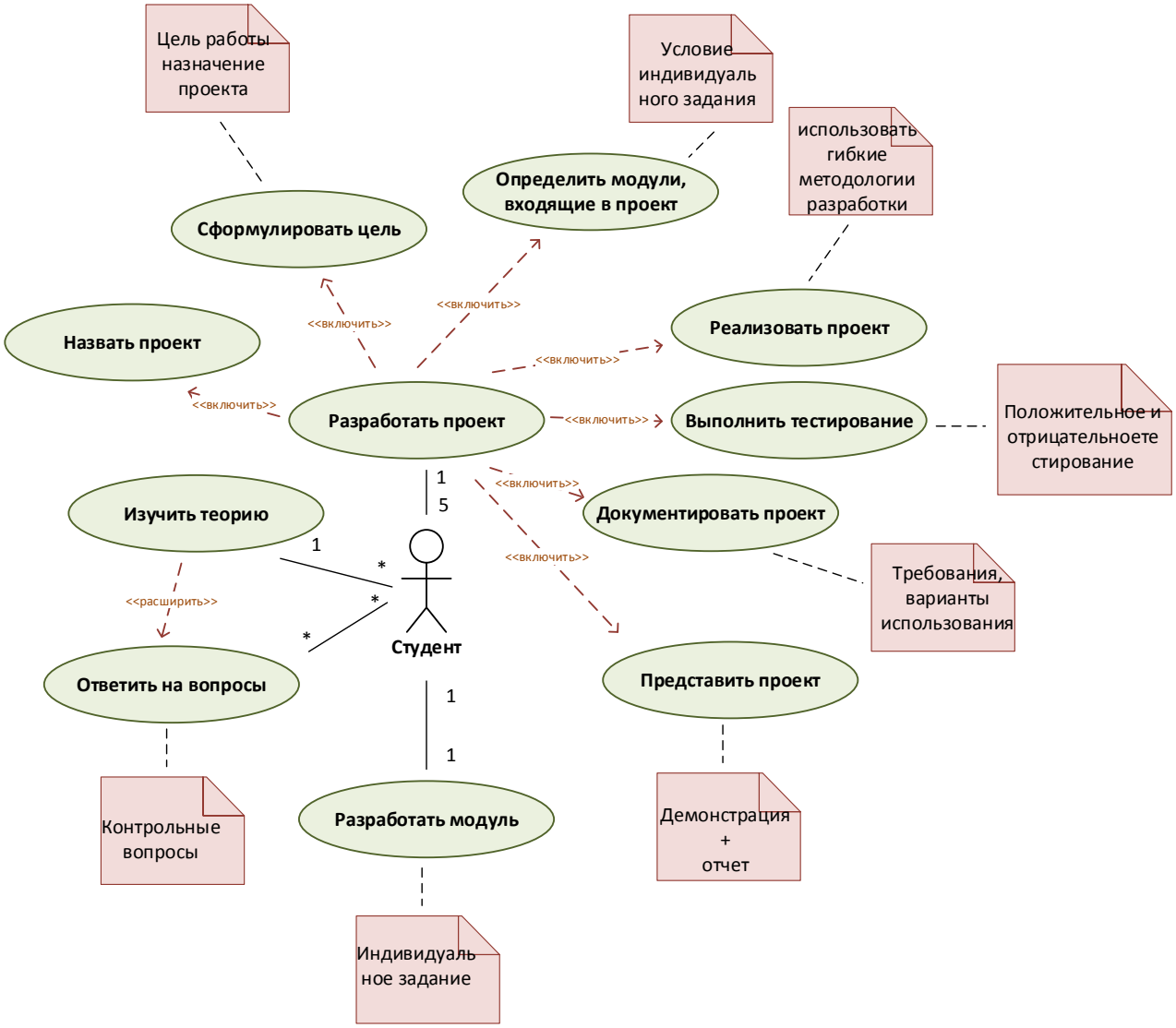
Некоторые пояснения

Помимо актеров и вариантов использования диаграмма вариантов использования содержит примечания – элементы, служащие для размещения на диаграмме поясняющей текстовой информации. Примечание может относиться к любому элементу диаграммы и соединяется с данным элементом штриховой линией.

На диаграмме показано, что у актера «Студент» и соответствующими вариантами использования существуют отношения ассоциации.

Базовый вариант использования «Ответить на вопросы» связан отношением расширения с вариантом использования «Изучить теорию».

Пример диаграммы вариантов использования системы для актера «Студент»:



Можно также расширить функциональное назначение моделируемой системы: ввести актера «Преподаватель» и построить диаграммы вариантов использования системы «Ознакомление с технологиями разработки ПО» для этой роли.

При взаимодействии с актером «Преподаватель» система должна обеспечить возможность выполнения следующих основных функциональных требований:

- «Преподаватель» разрабатывает техническое задание;
- «Преподаватель» должен организовать взаимодействие со скрам-мастером каждой команды;
- «Преподаватель» отвечает на возникающие вопросы;
- «Преподаватель» регистрирует скрам-мастера и состав команды, исходные данные проекта;

- «Преподаватель» осуществляет контроль хода выполнения индивидуального задания;
- «Преподаватель» осуществляет контроль хода выполнения проектирования и разработки проекта;
- «Преподаватель» проверить результаты выполнения индивидуального задания;
- «Преподаватель» принимает лабораторную работу.