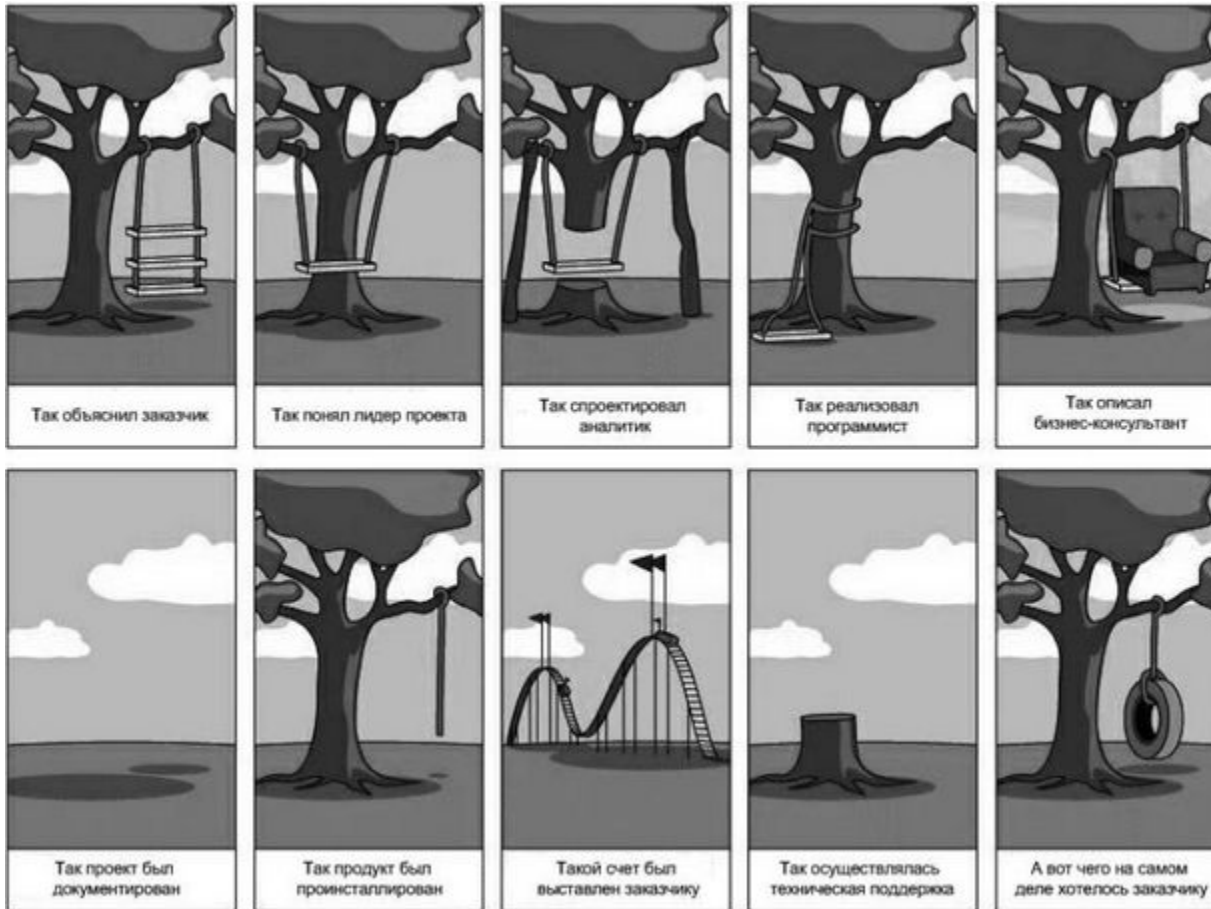


# UML – Unified Modeling Language

Откуда взялся The UML? Если говорить коротко, то UML вобрал в себя черты нотаций Грейди Буча (Grady Booch), Джима Румбаха (Jim Rumbaugh), Айвара Якобсона (Ivar Jacobson) и многих других.

В не такие уж и далекие 80-е годы было множество различных методологий моделирования. Каждая из них имела свои достоинства и недостатки, а также свою нотацию. То смутное время получило название "войны методов". Проблема в том, что разные люди использовали разные нотации, и для того чтобы понять, что описывает та или иная диаграмма, зачастую требовался "переводчик". Один и тот же символ мог означать в разных нотациях абсолютно разные вещи! На рисунке ниже можно увидеть лишь малую часть многообразия методов, которые существовали в то время и в какой-то мере повлияли на UML.

# Сложности при разработке программного обеспечения



# Сложности при разработке программного обеспечения

Здесь мы видим все проблемы программной инженерии, в частности проблемы с коммуникацией и пониманием, вызванные отсутствием четкой спецификации создаваемого продукта. Так вот, авторы UML определяют его как графический язык моделирования общего назначения (т. е. его можно применять для проектирования чего угодно - от простой качели, как на рисунке, до сложного аппаратно-программного комплекса или даже космического корабля), предназначенный для **спецификации, визуализации, проектирования и документирования** всех артефактов, создаваемых в ходе разработки.

Итак, UML в первую очередь - это спецификации. Заглянем снова в глоссарий и обнаружим, что

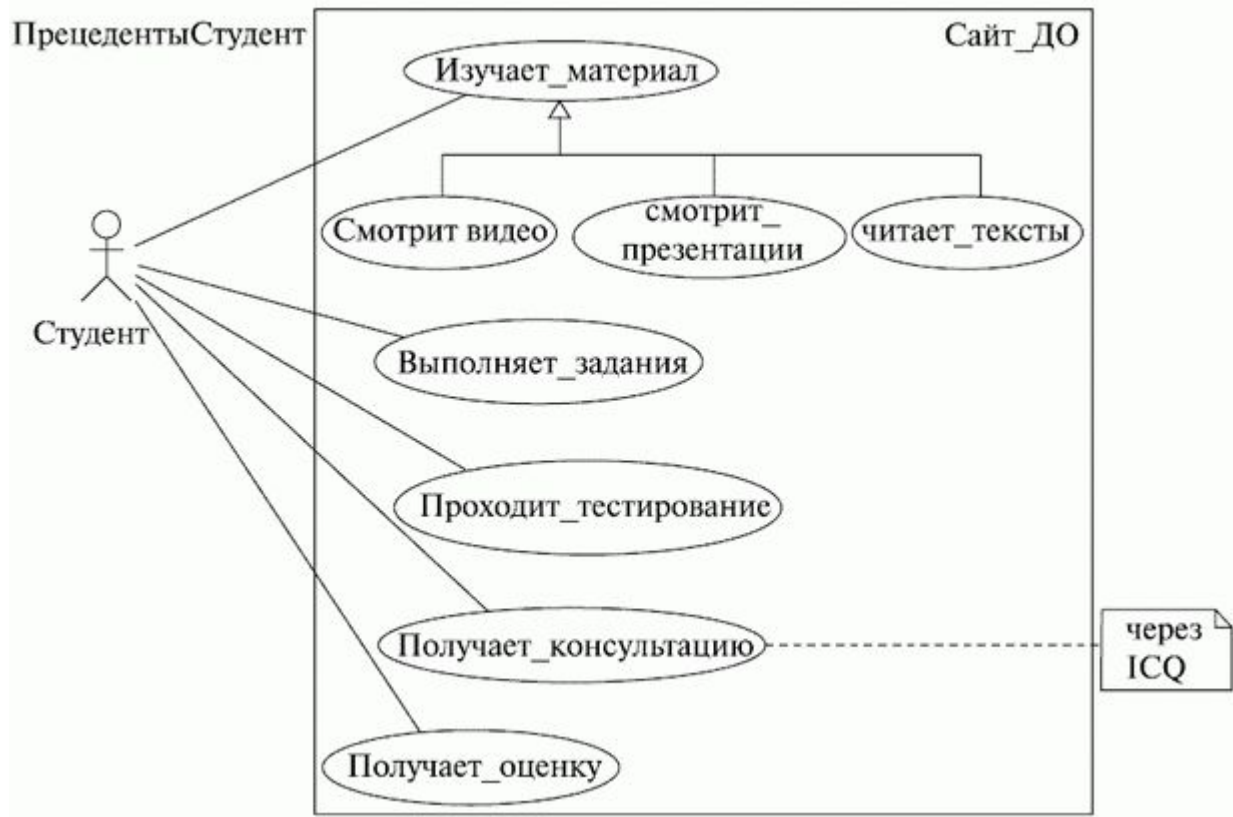
- **Спецификация** - подробное описание системы, которое полностью определяет ее цель и функциональные возможности.
- словесные спецификации на естественном языке;
- модельные спецификации;
- формальные спецификации.

# Сложности при разработке программного обеспечения

Когда мы говорим о том, что UML - это средство визуализации, мы имеем в виду модельные спецификации. Все мы знаем, как иногда трудно заставить себя "вникнуть" в суть материала, излагаемого в очередном учебнике или мануале. Изучение чего-то нового идет гораздо проще, если документ содержит не только текст, а еще и иллюстрации к нему. А если руководство или учебник выглядят как картинки с подписями, то усвоение нового материала происходит еще проще и эффективнее.

Так вот, такие картинки с подписями наглядны и интуитивно понятны, причем почти однозначно понимаются любыми заинтересованными лицами, так что могут использоваться в качестве средства общения между людьми. UML позволяет создавать такие простые и понятные картинки (модели), описывающие систему с разных сторон, которые можно показать заказчику и обсудить с ним, т. е. служит средством коммуникации в команде.

# Сложности при разработке программного обеспечения



# Строительные блоки UML

1. Сущности
2. Связи
3. Диаграммы

## Виды сущностей UML

1. Структурные
2. Поведенческие
3. Группирующие
4. Аннотирующие

# Структурные сущности

- Класс
- Интерфейс
- Кооперация
- Вариант использования
- Активный класс
- Компонент
- Артефакт
- Узел

# Поведенческие сущности

- Взаимодействие
- Автомат (state machine)
- Деятельность (activity)



# Группирующие сущности

- Пакет
- Каркас (framework)
- Модель

# Аннотирующие сущности

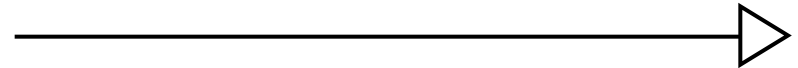
- Примечание

# СВЯЗИ

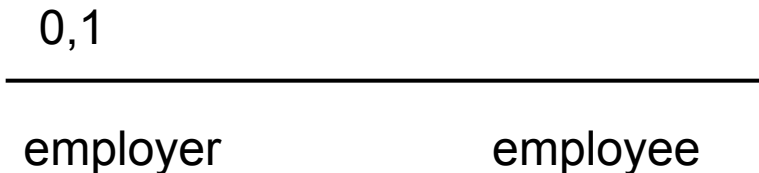
Зависимость – связь между двумя элементами модели, в которой изменение одного элемента может привести к изменению другого.



Обобщение – выражает специализацию, в которой специализированный элемент (потомок) строится по спецификациям обобщенного элемента (родителя).



Ассоциация – это структурная связь между классами, которая описывает набор связей, существующих между классами.



Реализация – это семантическая связь, когда один из классификаторов специфицирует соглашение, которого второй обязан придерживаться.

