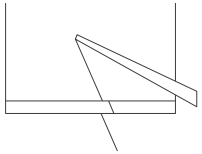
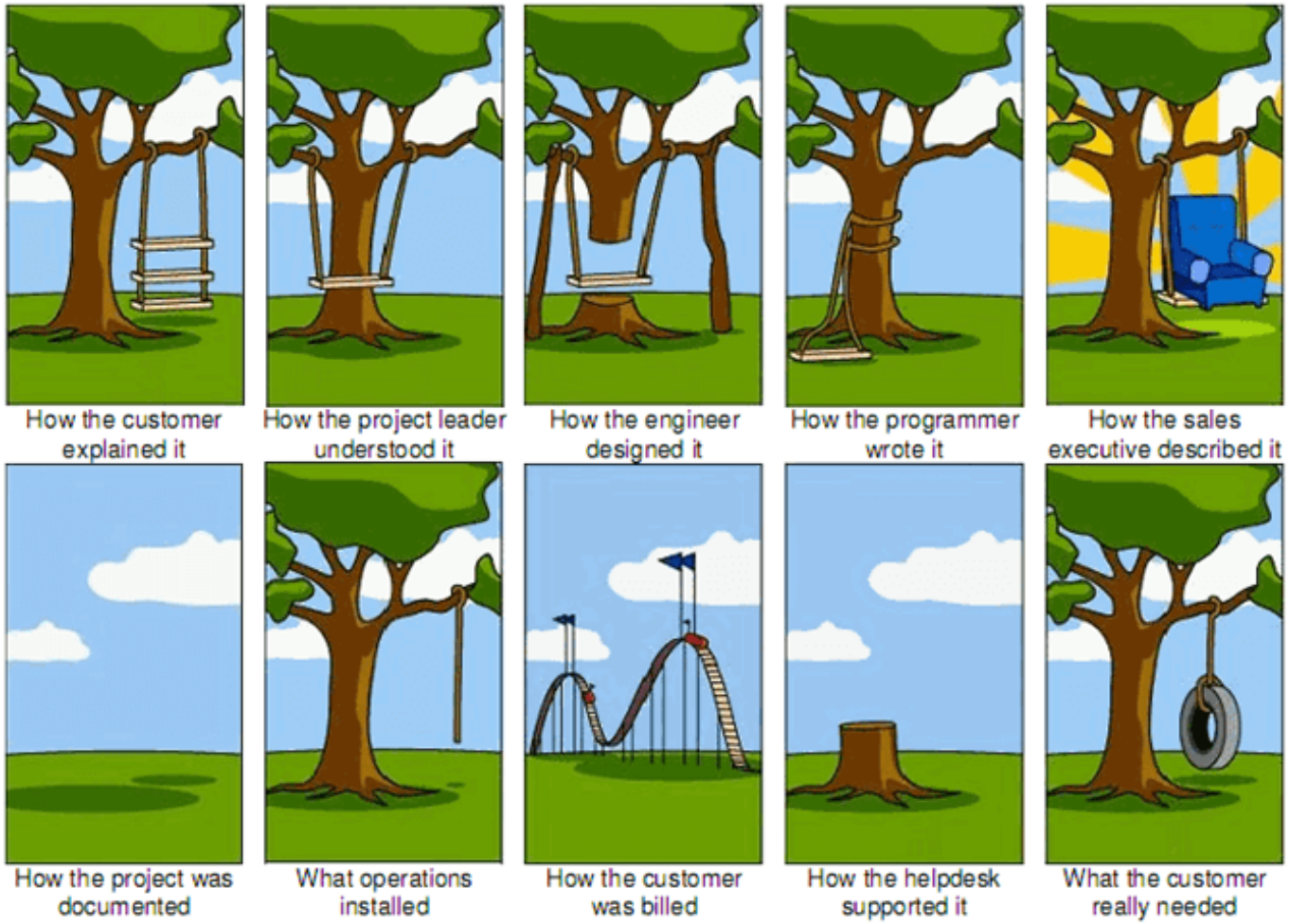
Bring ideas to life
VIA University College



Analysis - Domain Models

SWE1

Develop Domain Models



Assumptions is bad!
Assumptions is the mother of all fuck-ups!

We MUST understand the problem – don't think in software/implementation!!

Domain Model

A visual guide over a specific problem area – typical Class Diagrams

- Important concepts
- Relationships between concepts

Used for

- Common understanding between customer/end user and programmer of the problem
- Communication with stakeholders
- Finding problems in the understanding
- Finding improvements

Artefacts: UML Diagrams and descriptions

Domain Model Artefacts

Dynamic views (shows a given situation)

Actors and objects with relations

- No classes exist in a system when it is running!!!
- Shows a dynamic behaviour

Behaviour Diagrams:

- Sequence/activity/communication diagrams

Static views (show what is possible)

Classes and relationships

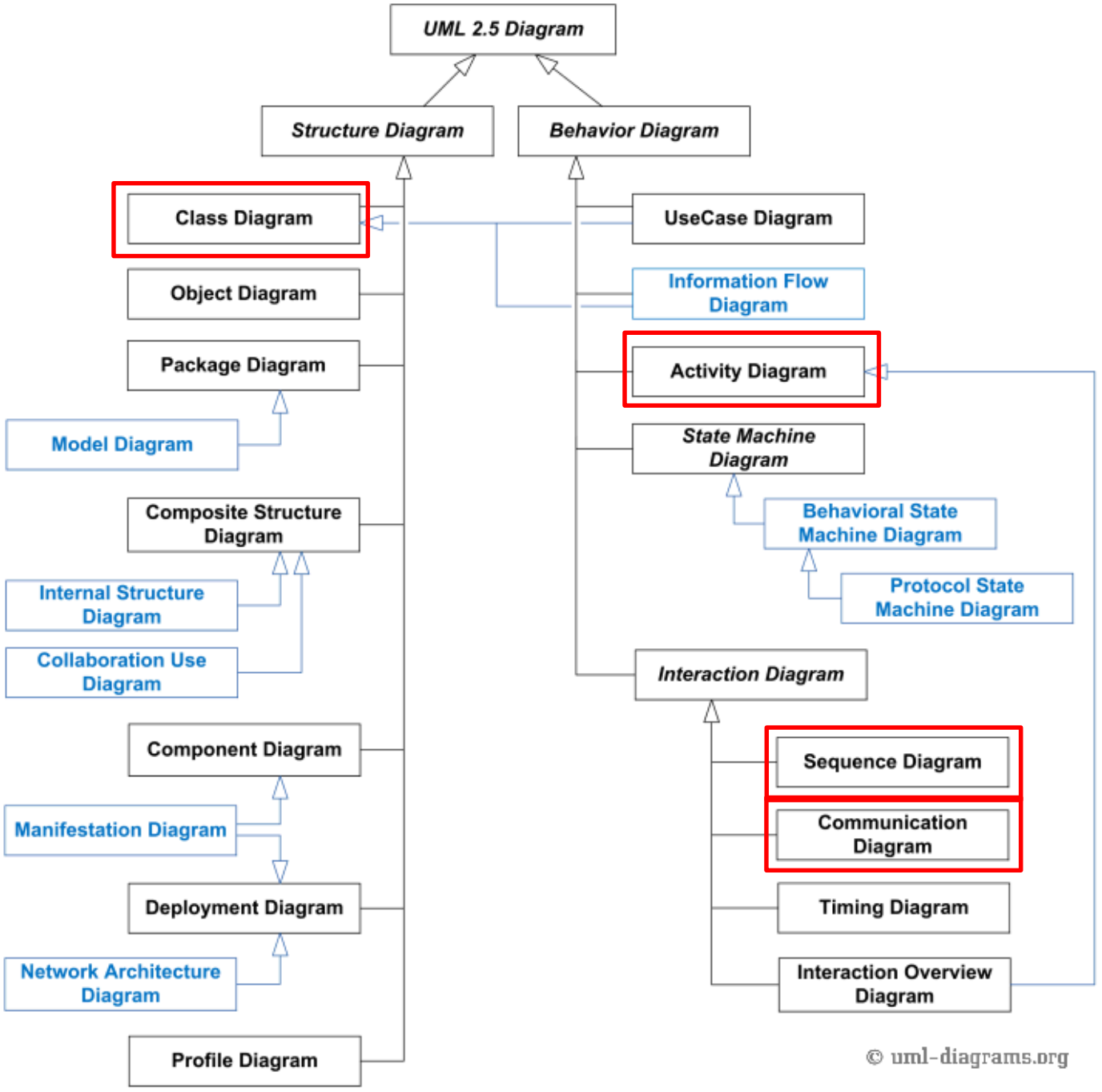
- Shows static structure of the system

Structure Diagrams:

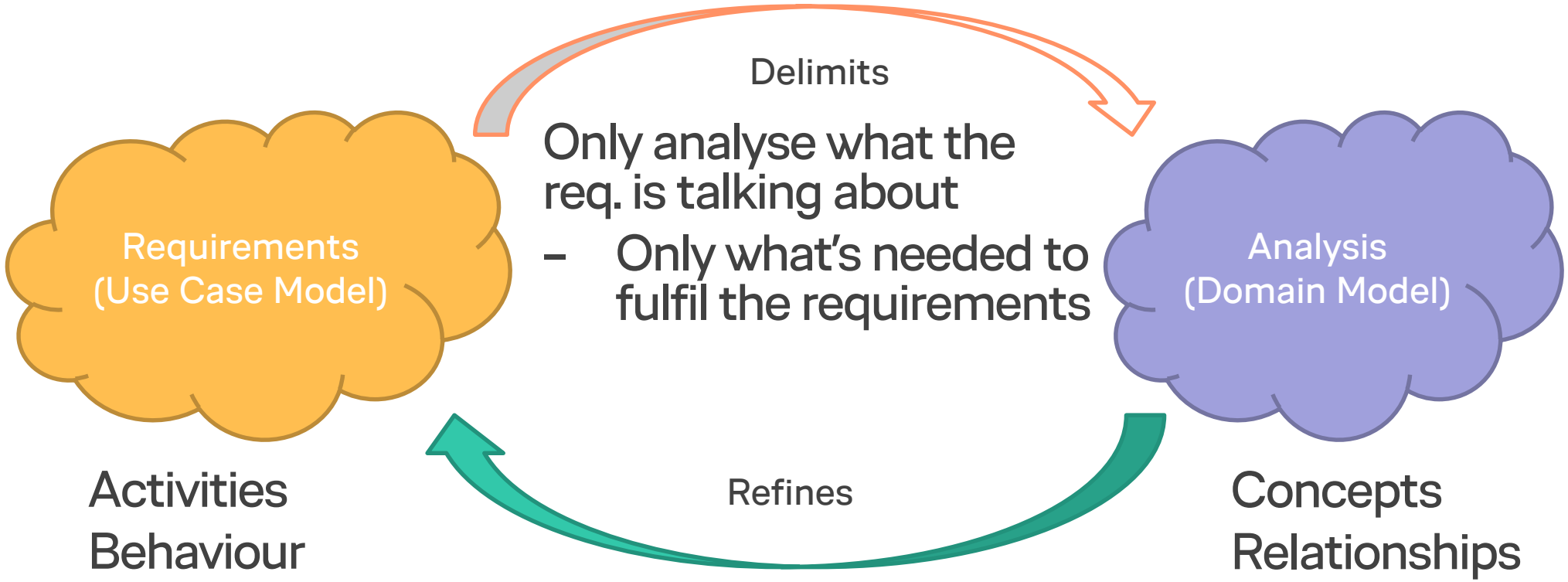
- Class diagrams

Domain Model Artefacts

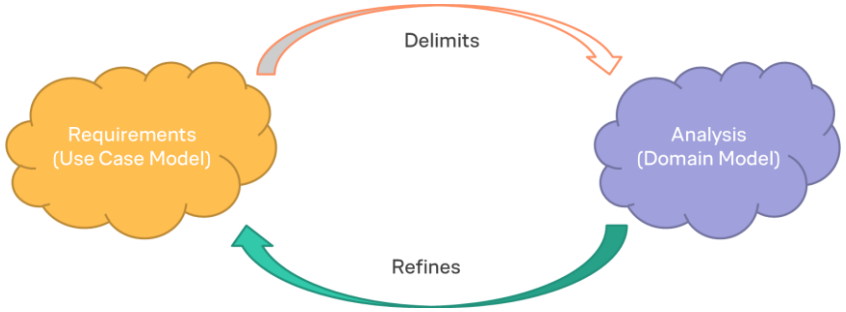
Marked UML diagrams are typical artifacts



How to come from Requirements to Domain Model



How to come from Requirements to Domain Model



Typically an interactive process

- One or two requirements are taken out for analysis in each iteration/sprint
- The Domain Model will be build up step by step

A Object/Communication diagram is often needed to really understand the Domain Model

- Shows a given situation/snapshot of the system at a given time

How to come from Requirements to Domain Model

Often you will need to be an "expert" in the customers problem domain

- Our role is to help the customer to earn money
- Give them ideas
- Raise questions to the customers requirements

In education programs

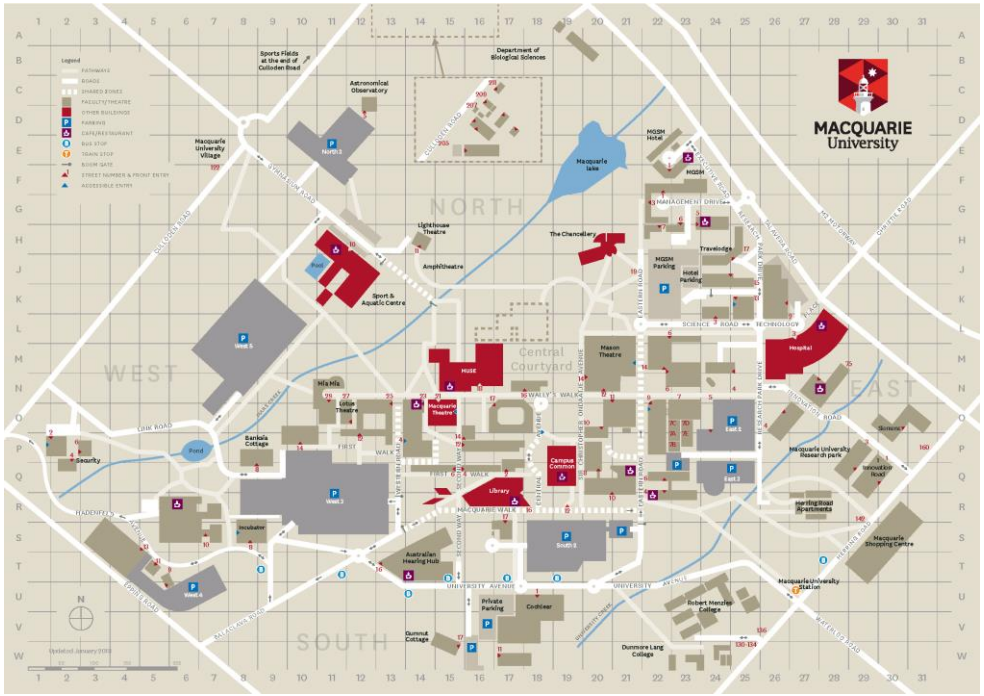
- We work in well-known problem domains like Math, Recipe systems etc.
- In real life you will meet problem domains you don't know anything about

Requirements and Domain Models helps us to understand what we need to know!

How to come from Requirements to Domain Model

Think like a Map-Maker

- Use words/vocabulary already know in the domain
- Remember: Not to many details – only what is needed to understand
- Don’t add things that doesn't exist in the domain
  - Software things
- Use experts
  - End-users, books, consultants, articles etc.
- Always discuss your Domain Model with domain experts and customer!



How to find Conceptual Classes Step – by step

Step 1

Find concept candidates

- Mark up all **Nouns** and Verbs in req. with two different colours
- Reuse existing models
- Create a category list (list of candidates)
- Brainstorm – remember no debate when brainstorming

Step 2

Remove duplicates

- Things has often different names – ask the domain which to use and be consistent

How to find Conceptual Classes Step – by step

Step 2 – continued

Classes vs. attributes

Ex. Two concept candidates is found *Item* and *Price*

Is it two separate classes?

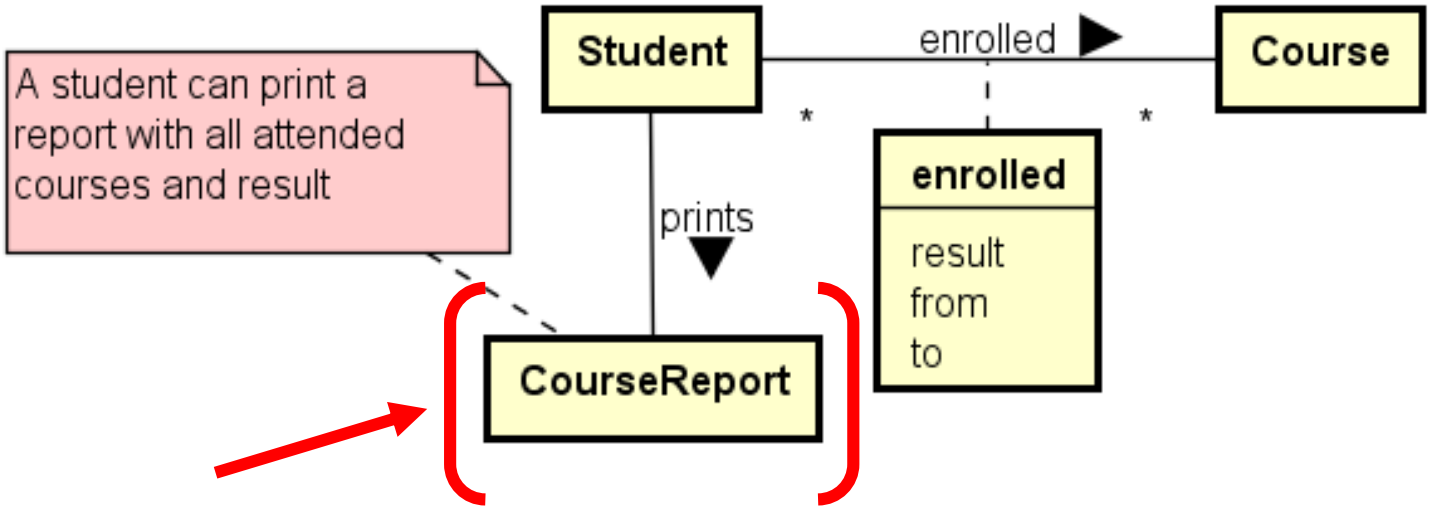
Is *Price* an attribute to an Item?

Or maybe we need for *Price* as a separate class and a relation-ship? Can an *Item* have more *Prices*

How to find Conceptual Classes Step – by step

Step 2 – continued

Find Report Classes



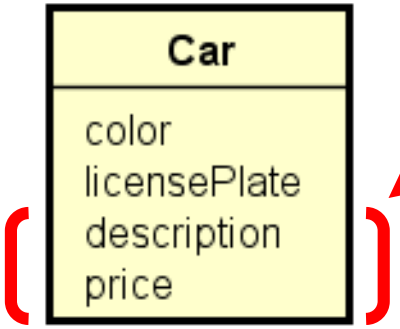
Report Class!
It is not needed, the report can be assembled and printed via the enrolled relation-ship

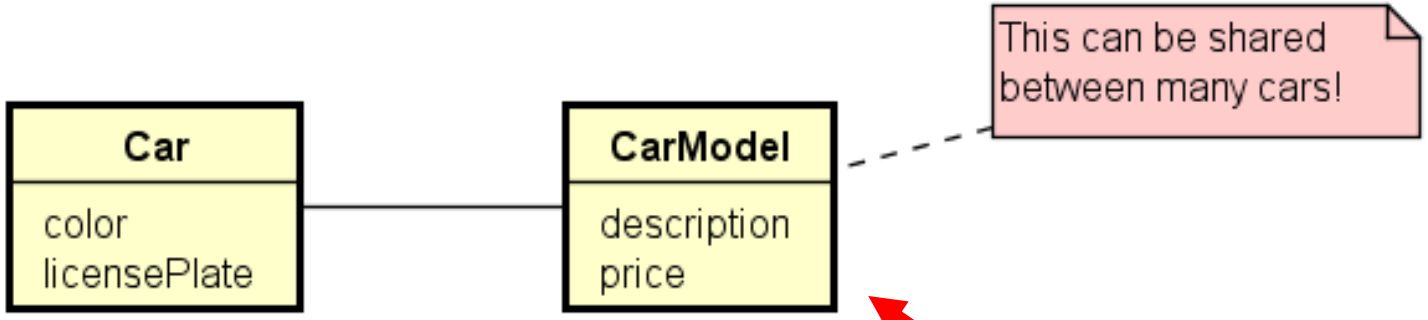
How to find Conceptual Classes Step – by step

Remember:
There is not only one right answer – it depends!!!

Step 2 – continued
Find Description Classes

Description!
Is this the same for all cars of same model? (Duplication!!)





Description Class!
If it is the same for all cars of same model – then it is a separate class/concept and duplication is avoided!

But is it true in all cases?
- Dealer of new cars
- Dealer of second hand cars

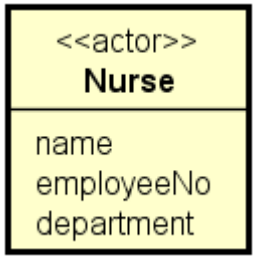
How to find Conceptual Classes Step – by step

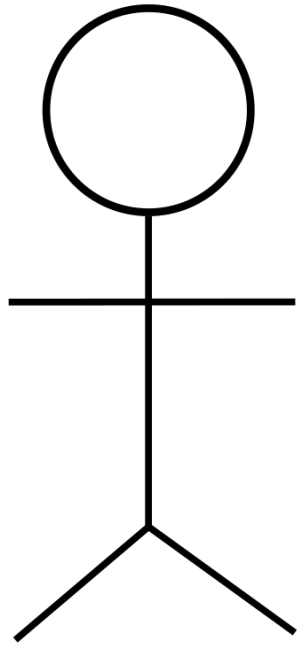
Step 2 – continued

Look at Actors

- Are they classes? – sometimes, sometimes not

  Must the system store information about them?





Nurse

Associations/Relation-ships in Domain Models

Association in Domain Model

- A named relation between two classes
- Should give meaning and be important
- Increases understanding

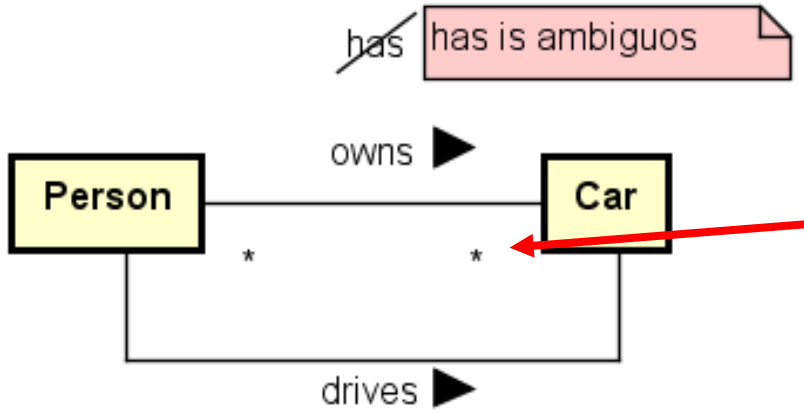
Comes from

- Requirements
  - Verbs are candidates
- Common association list
- Are there things that must be remembered over time?

Associations/Relation-ships in Domain Models

Guidelines

- Use meaningful names – **not** has/contains/consist of etc.
- Show multiplicity if needed for understanding



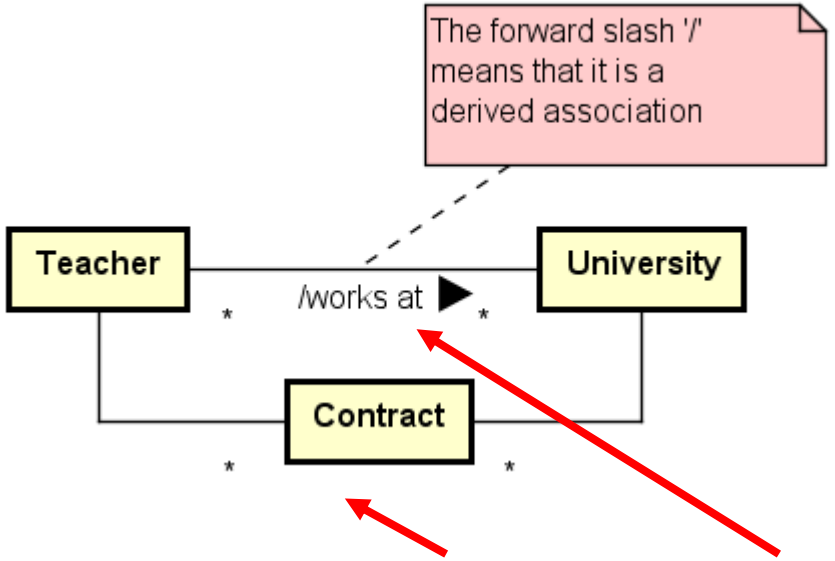
Are the multiplicities necessary? Or is it obvious?

Is it right that a Car can have no owners?

Associations/Relation-ships in Domain Models

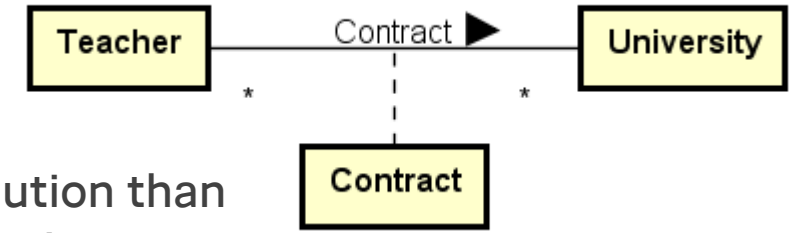
Derived associations

- Something that can be found or calculated
- Are shown if they get better readability/understanding



The *works at* association can be derived from the *Teacher – Contract – University* associations

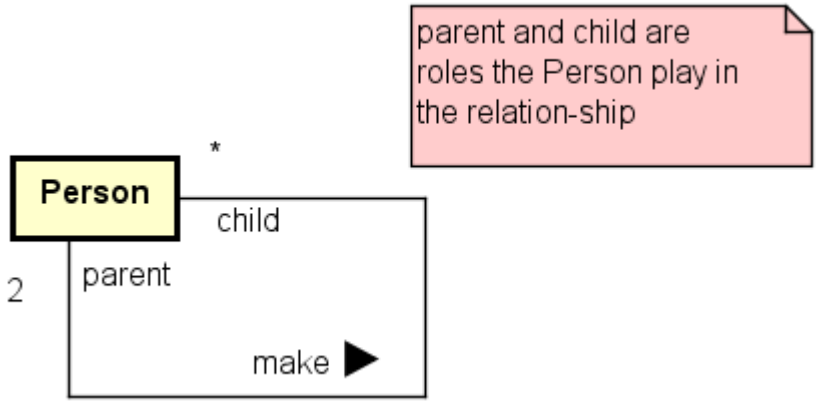
Association Classes



This is a better solution than the derived association

Associations/Relation-ships in Domain Models

Reflexive associations



Can you draw up a object/communication diagram showing a family with two children?

Attributes in Domain Models

The information/data that is needed to support the requirements

- focus on attributes that are Odd, Strange, Advanced – not the obvious ones

Candidates:

- Look at what is coming in- and out from a system
- What data is handled by the system?

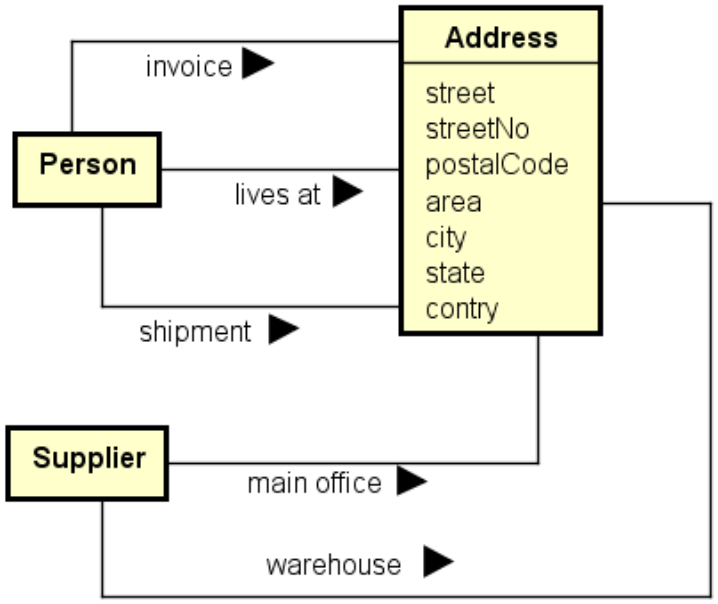
Attributes in Domain Models

Data type classes

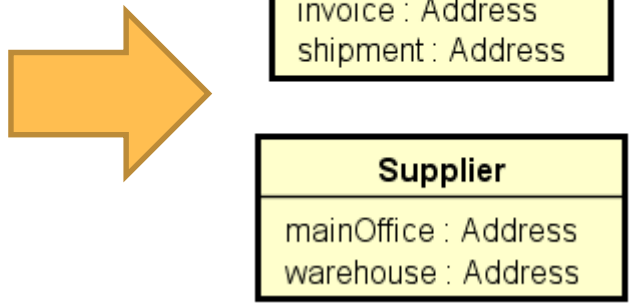
Ex: A Person has an address

What is an address?

What if a Person has more than one address?



This is chaotic!!



This is better

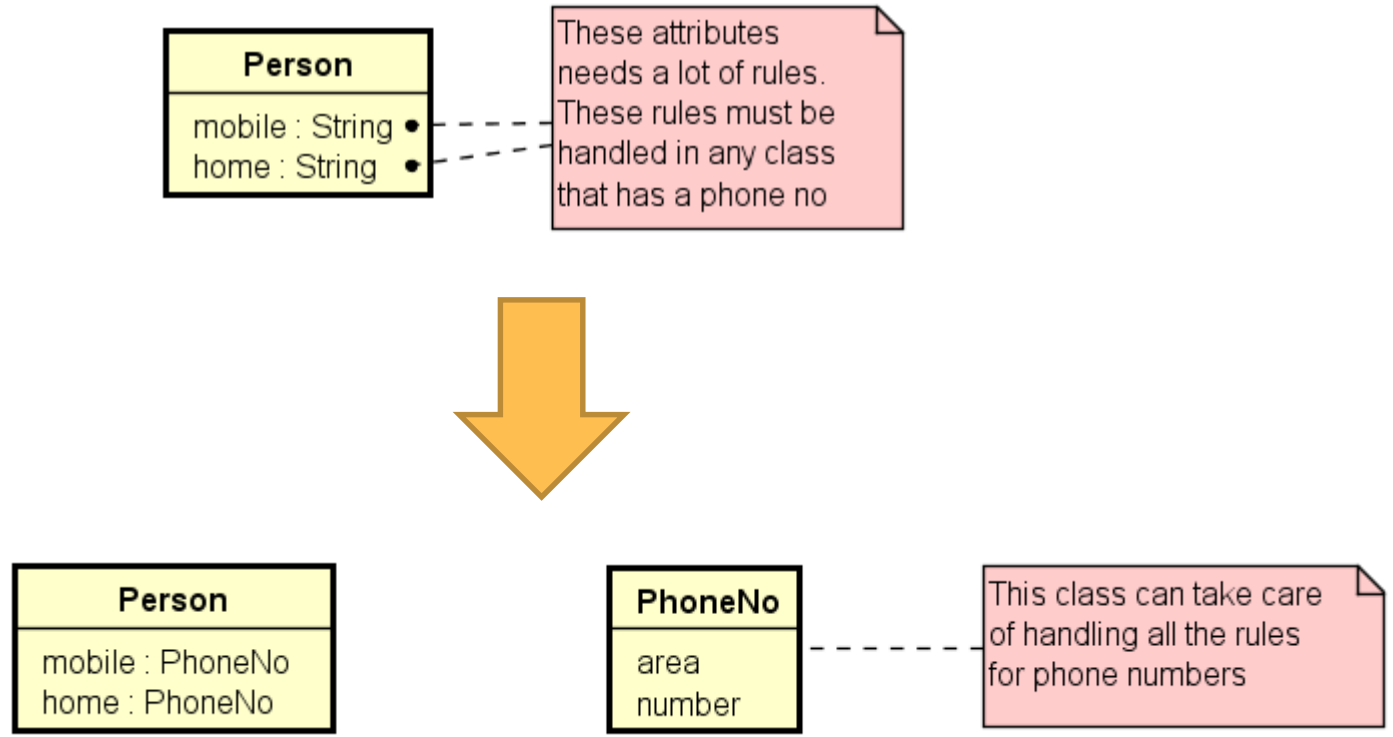
Data type class

Attributes in Domain Models

Painted types

Primitive types that we apply a lot of rules to

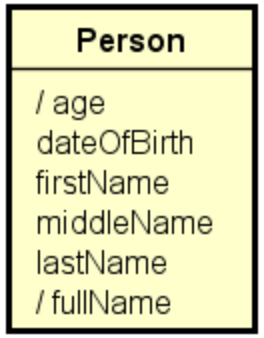
Ex: A phone no



Attributes in Domain Models

Derived Attributes
Attributes that does not exist as attributes, but will be calculated when needed

totalPrice will be calculated by the Receipt class via the association



A *Person* has a date of birth, so we can always find the persons age without having a real attribute for it – same with *fullName*

Derived attributes are show to make the diagram more clear and understandable

Attributes in Domain Models

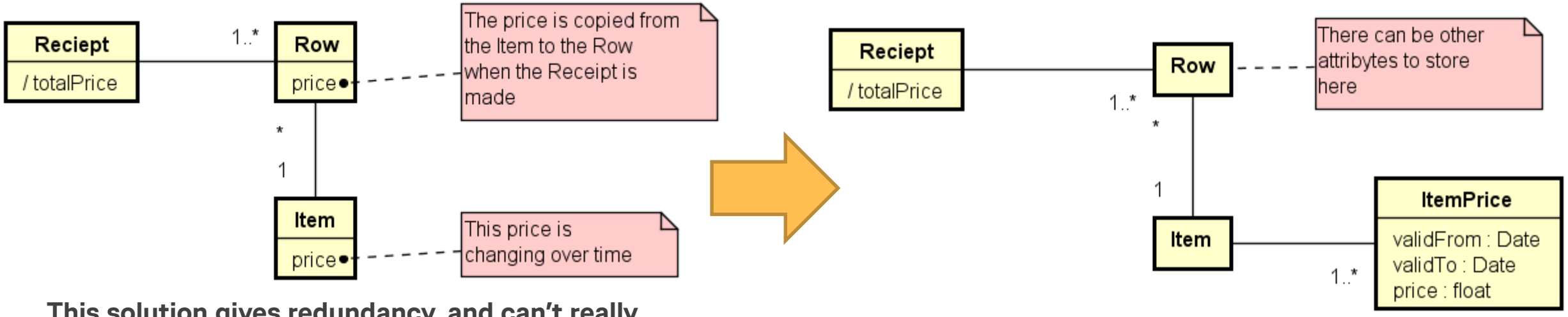
History

Often it is needed to record history data, but is **often forgot by the customer**

Watch for historical data – and remember it makes things more complicated!

The customer wants to show prices over time

Ex: What if Item prices change over time?



This solution gives redundancy, and can't really keep track of prices over time

Exercise

Look at a Dice game that has these simple “requirements”/rules:

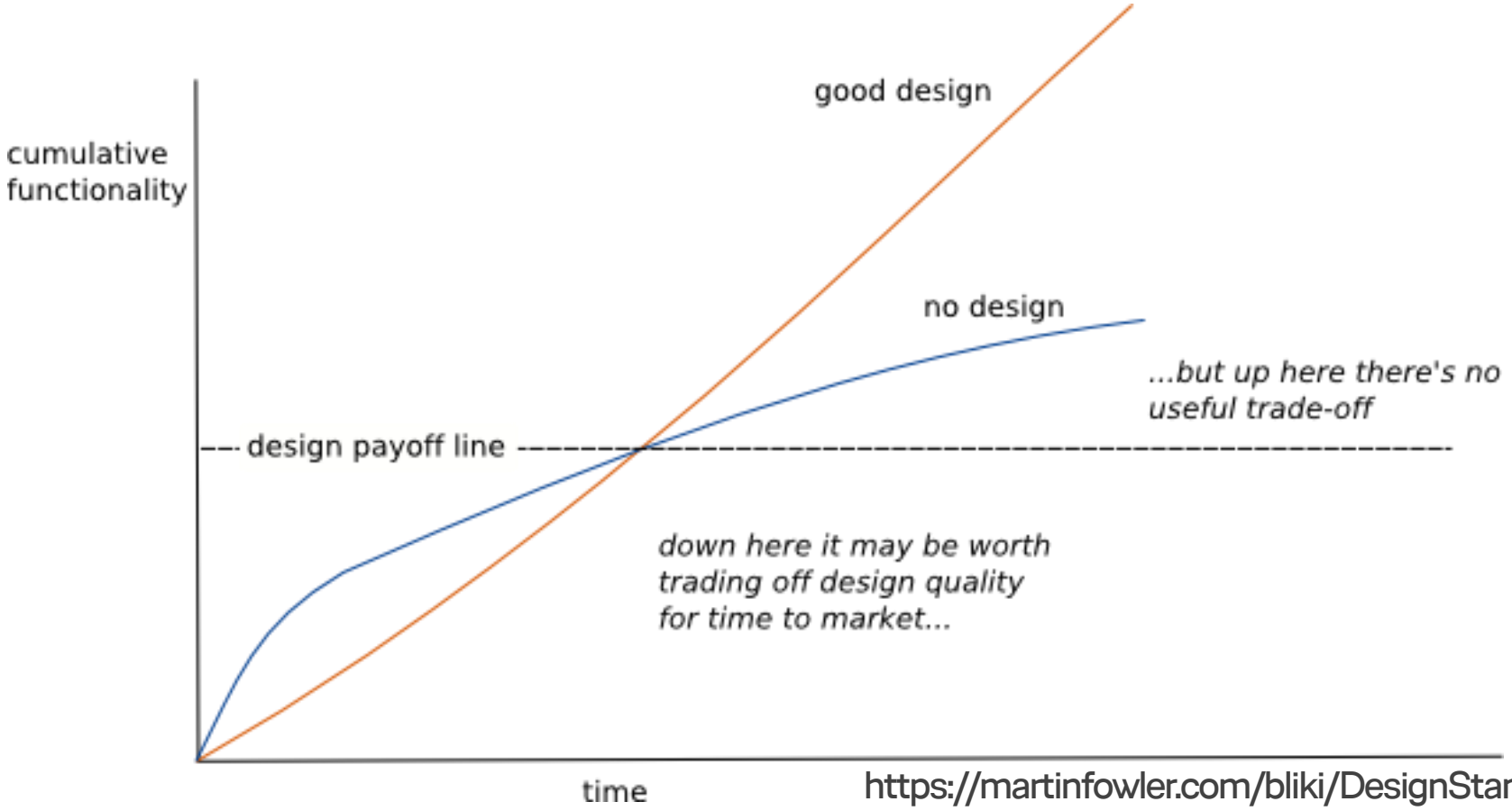
Requirements:
The dice are rolled and the result is presented; if the sum is seven the player wins!

Find the conceptual classes and their relation-ships and create a Domain Model.
First on paper, and then in Astah

Is it worth the effort to design software well?

Ask your-self this simple question:

"Do I want my software to be a success?"



https://martinfowler.com/bliki/DesignStaminaHypothesis.html