Tampereen yliopisto

Large Scale Software Design

Group assignment, facilitation 1

Spring 2024

Group assignment

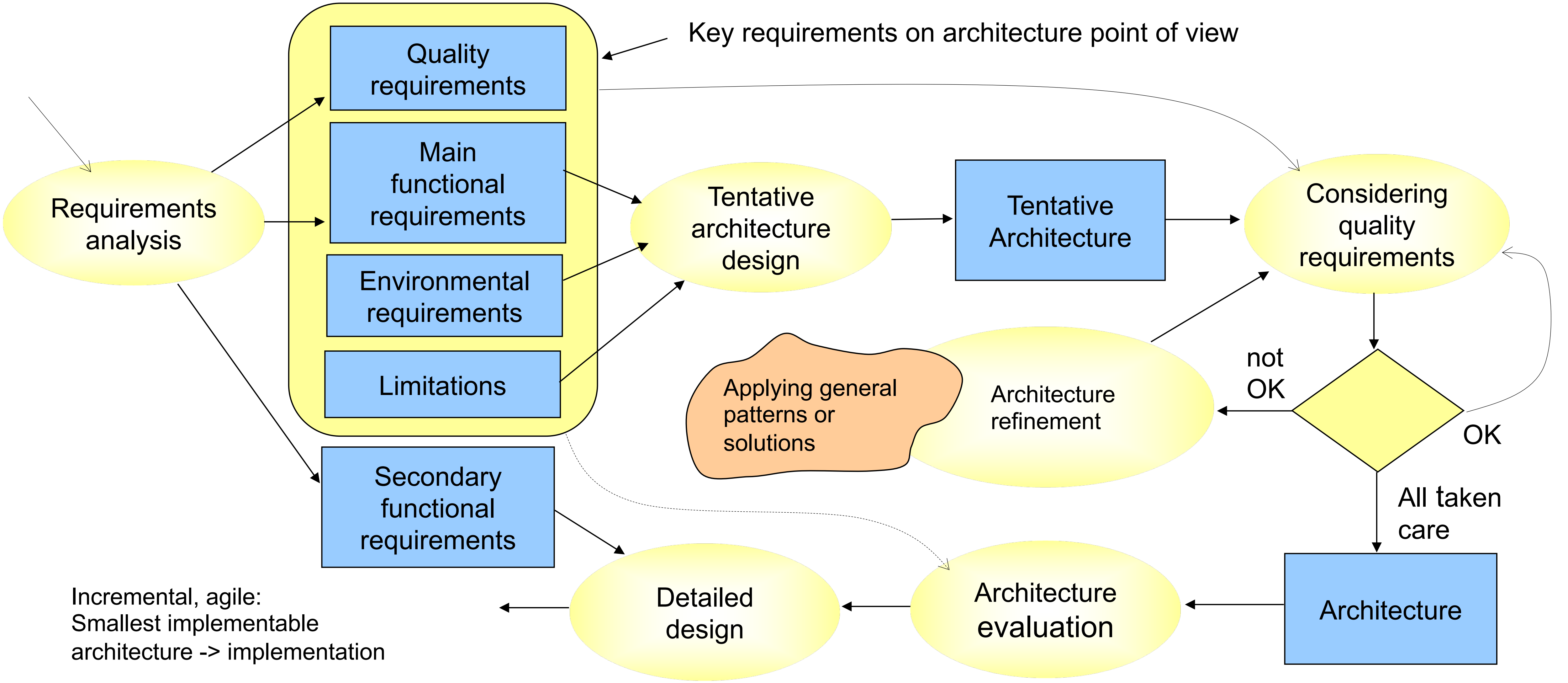
- T-Low food delivery service
  - https://plus.tuni.fi/comp.se.210/spring2024/assignment

1. Intermediate
   - Tuesday, 11th April 2024
   - Only requirements and use cases (one use case diagram is enough)
     - The scenarios are the +1 part of the 4+1 model
2. Final
   - Tuesday, 11th May 2024
   - Includes everything …also intermediate submission corrected after feedback!
   - Max. 4000 words for document
   - Two parts to return:
     - Assignment as PDF with diagrams embedded
     - 4 UML charts drawn also as separate submission (Papyrus files)

Software architecture and quality requirements

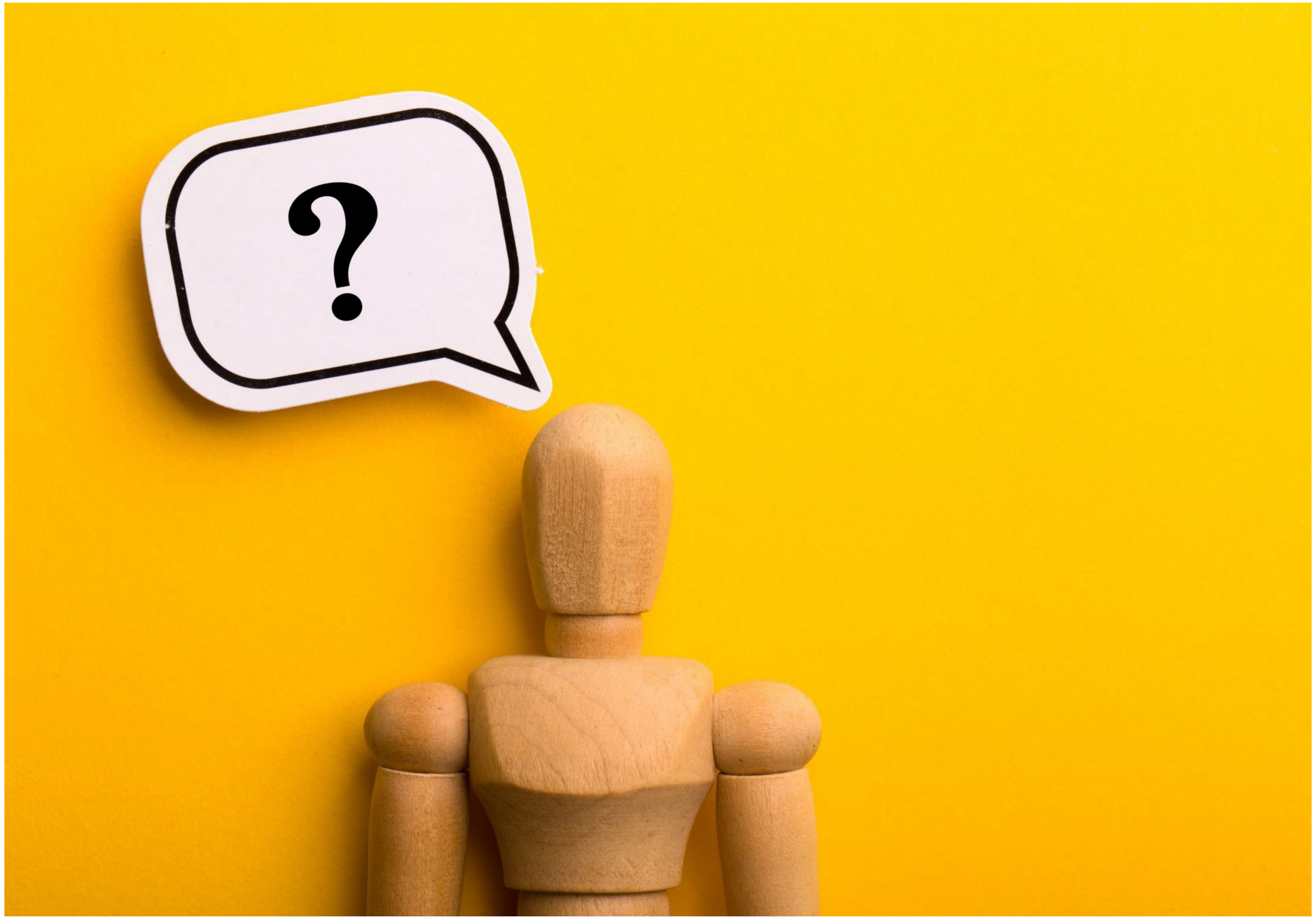
- The architecture is defined *mostly* by its quality requirements, not by functional requirements.
- Architecture is a way to take into account quality requirements in software development process.
- Here quality refers to the quality how the systems performs its logical functions
  - E.g. response time with normal load is 5 ms, components should be exchangeable, …
- Essential requirements are normally prioritised
  - Typically one or two quality properties dominate the architecture.
- Preserving connection between architecture design and requirements is essential
  - How the system fulfils especially quality requirements.

Developing the architecture

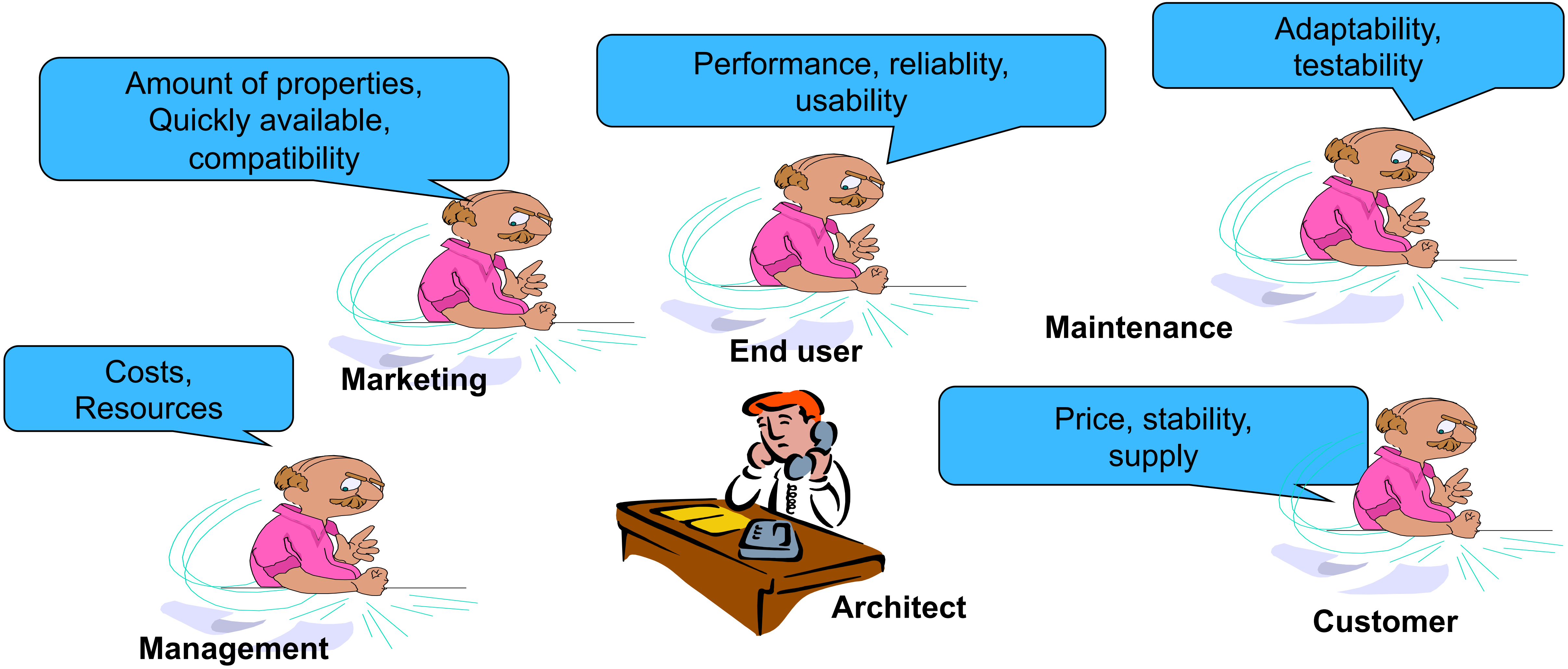

Key requirements on architecture point of view
Quality requirements
Main functional requirements
Environmental requirements
Limitations
Requirements analysis
Tentative architecture design
Tentative Architecture
Considering quality requirements
Applying general patterns or solutions
Architecture refinement
not OK
OK
All taken care
Secondary functional requirements
Detailed design
Architecture evaluation
Architecture

Incremental, agile:
Smallest implementable
architecture -> implementation

What could go wrong?



Remember: Conflicting quality requirements



Amount of properties, Quickly available, compatibility

Marketing

Performance, reliablity, usability

End user

Adaptability, testability

Maintenance

Costs, Resources

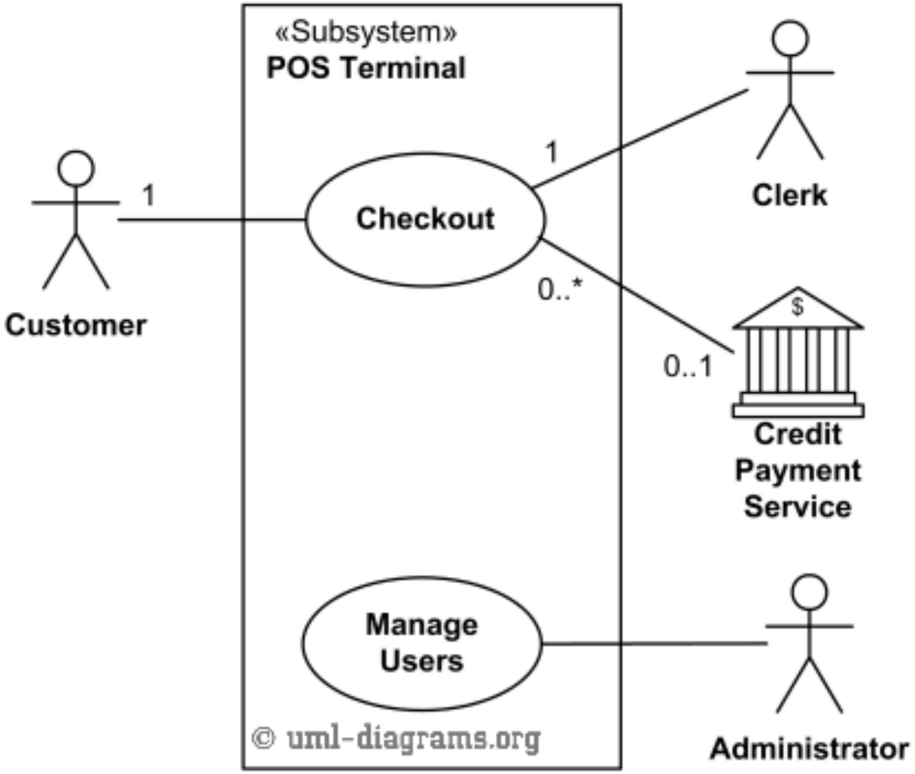
Management

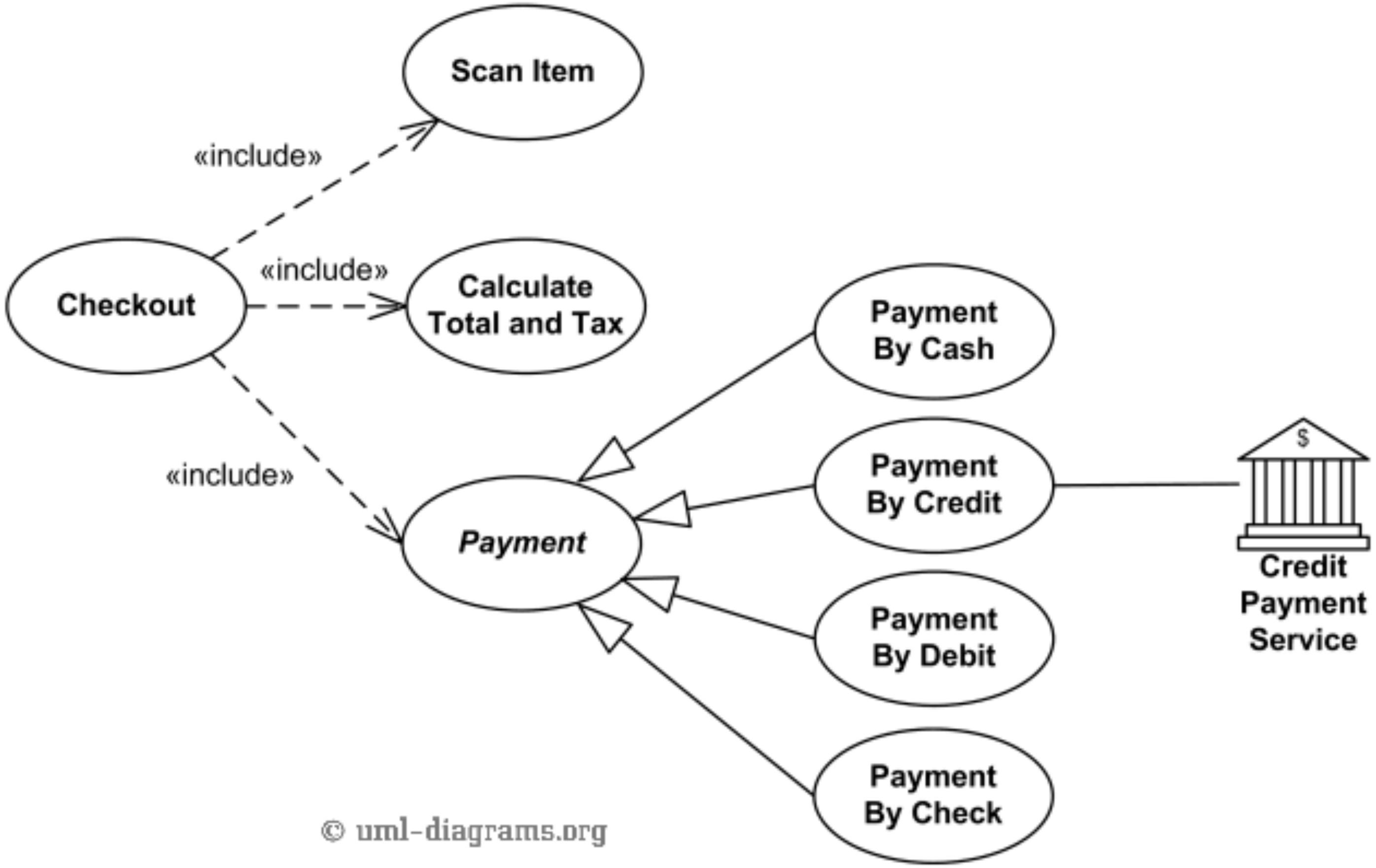
Architect

Price, stability, supply

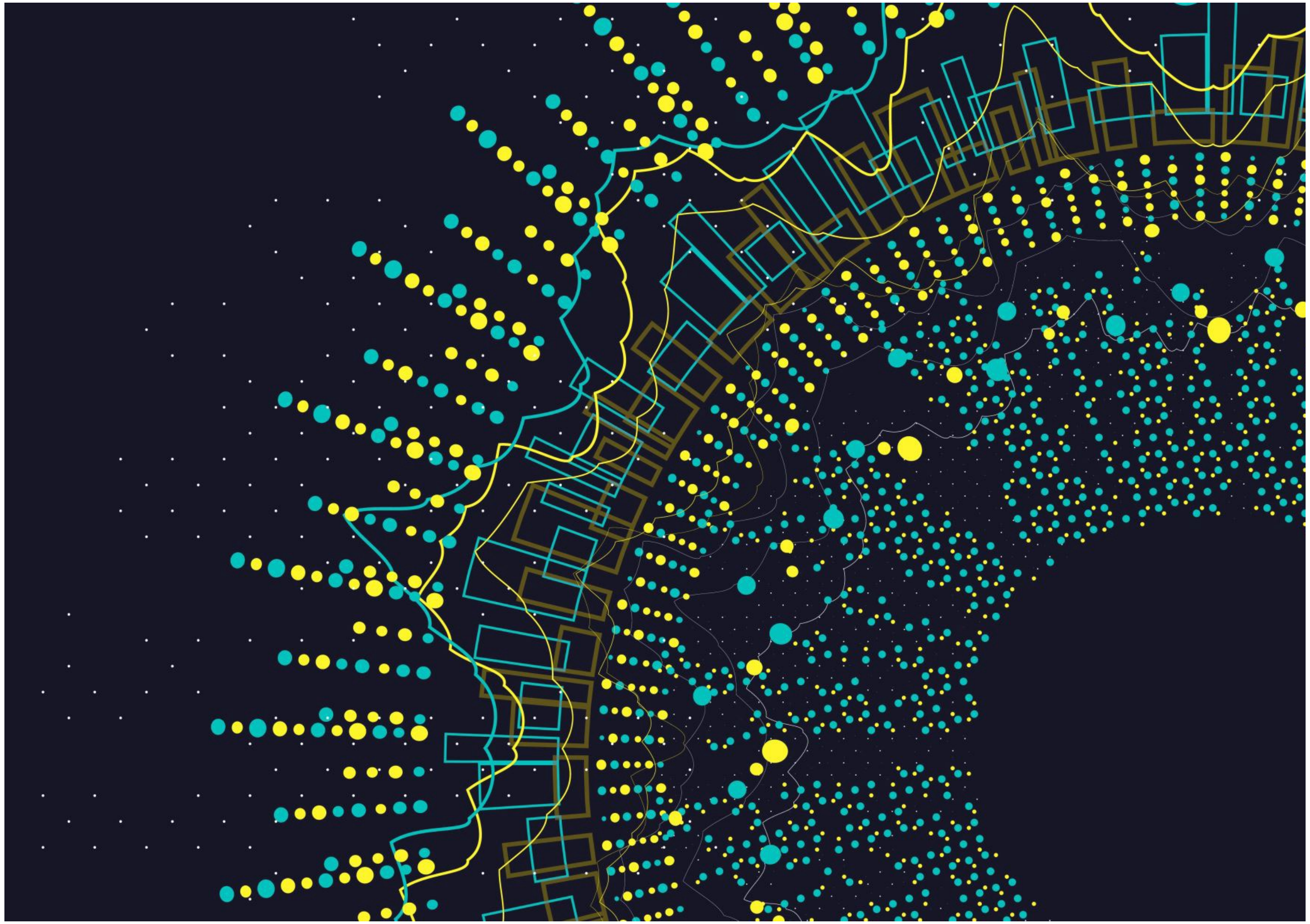
Customer

Use case diagrams

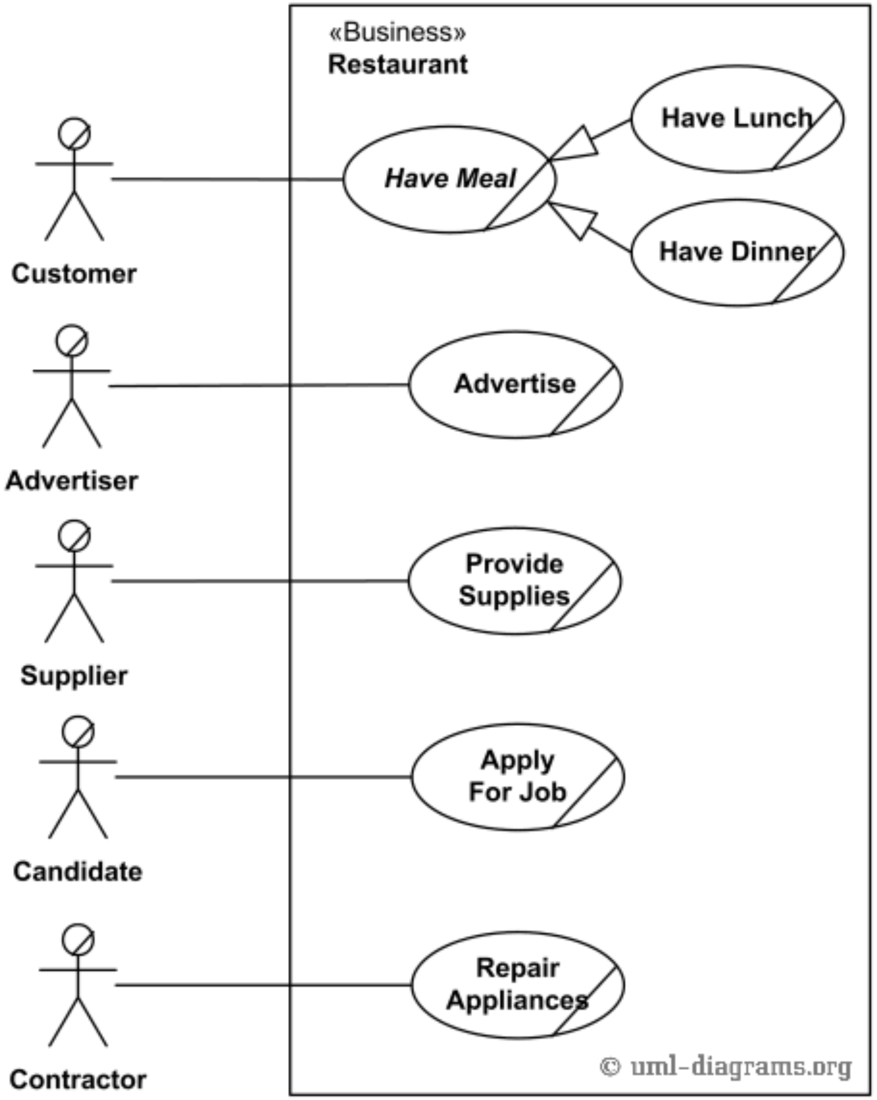


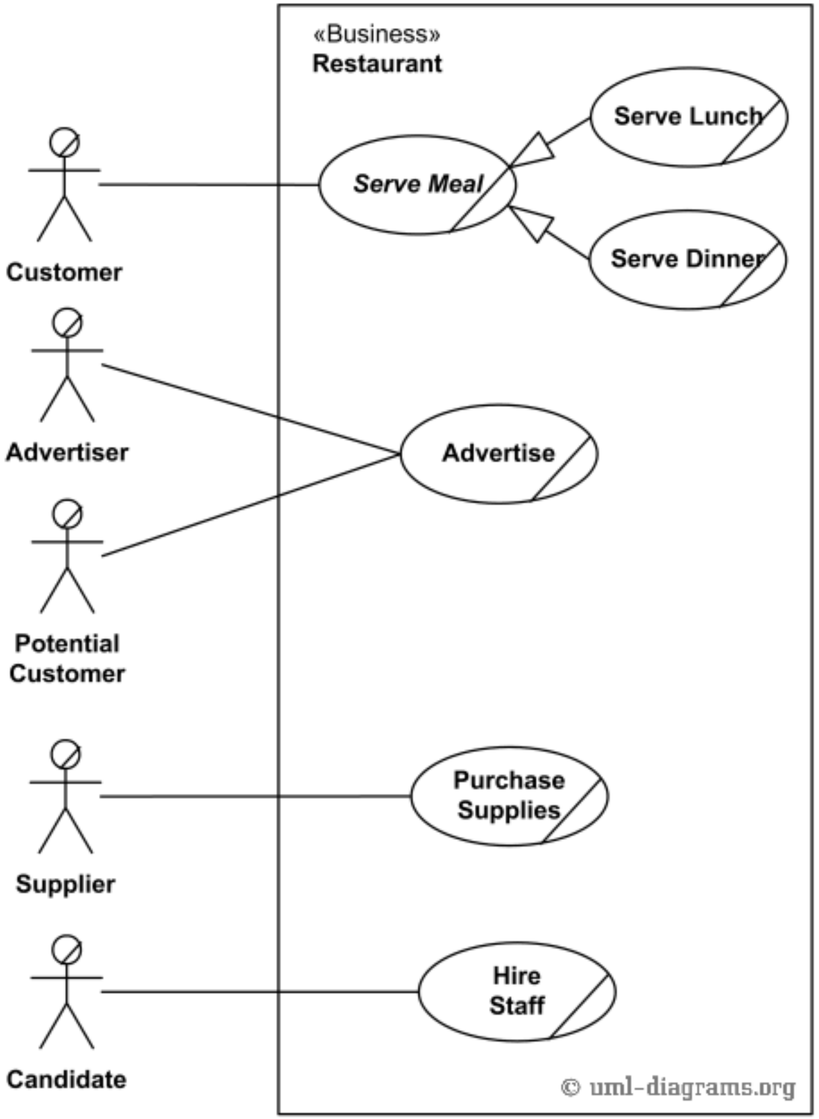


Are use cases always correct and complete?



Use case diagrams: two different views





Documenting a use case

- Documents the user’s interaction with the system, the steps needed to accomplish the task and how the system responds
- Use cases do not contain
  - Implementation specific language
  - Details about the user interfaces or screens

Use case documentation

Identifier and name	UC-1 Some descriptive name
Actors	List here whichever are the external actors that participate in the use case. The actors can be users and/or external systems or software.
Preconditions	Describe the preconditions required for the use case to happen.
Main success scenario	Explain here what happens when the use case succeeds. Often, there is a single main actor that determines the viewpoint to be applied. Usually, the main success scenario is straightforward and contains almost no conditions. Therefore, do not try to give an exhaustive description of all possible combinations of what the main actor could decide to do, but rather give a single success scenario. Still, you should use exceptions for any exceptional situations.
Exceptions	Give a list of exceptions that may affect the main success scenario. Usually, an exception indicates that something is wrong or in an unexpected state. Explain what causes the exception and what happens then. Give each exception an identifier and refer to it in main success scenario to indicate when the exception may occur.
Expected result	This is the result you expect when the main success scenario finishes.

Use case example (modified from: usability.gov)

Identifier and name	UC-23 Do laundry
Actors	Housekeeper
Preconditions	The required appliances are available, the housekeeper is on site, there is laundry, it is Wednesday
Main success scenario	On Wednesdays, the housekeeper *reports* to the laundry room. She *sorts* the laundry that is there. Then she *washes* each load. She *dries* each load. She *folds* the items that need folding. She *irons* and *hangs* the items that are wrinkled. She *throws* away any laundry item that is irrevocably shrunken, soiled or scorched.
Exceptions	1. The laundry is still dirty after washing, … 2. The machines fail, ...
Expected result	Laundry is clean and folded.

Group project work

- Define your use cases (diagram + descriptions)
- Detail (non-functional) requirements (if needed)