MUNI
ECON

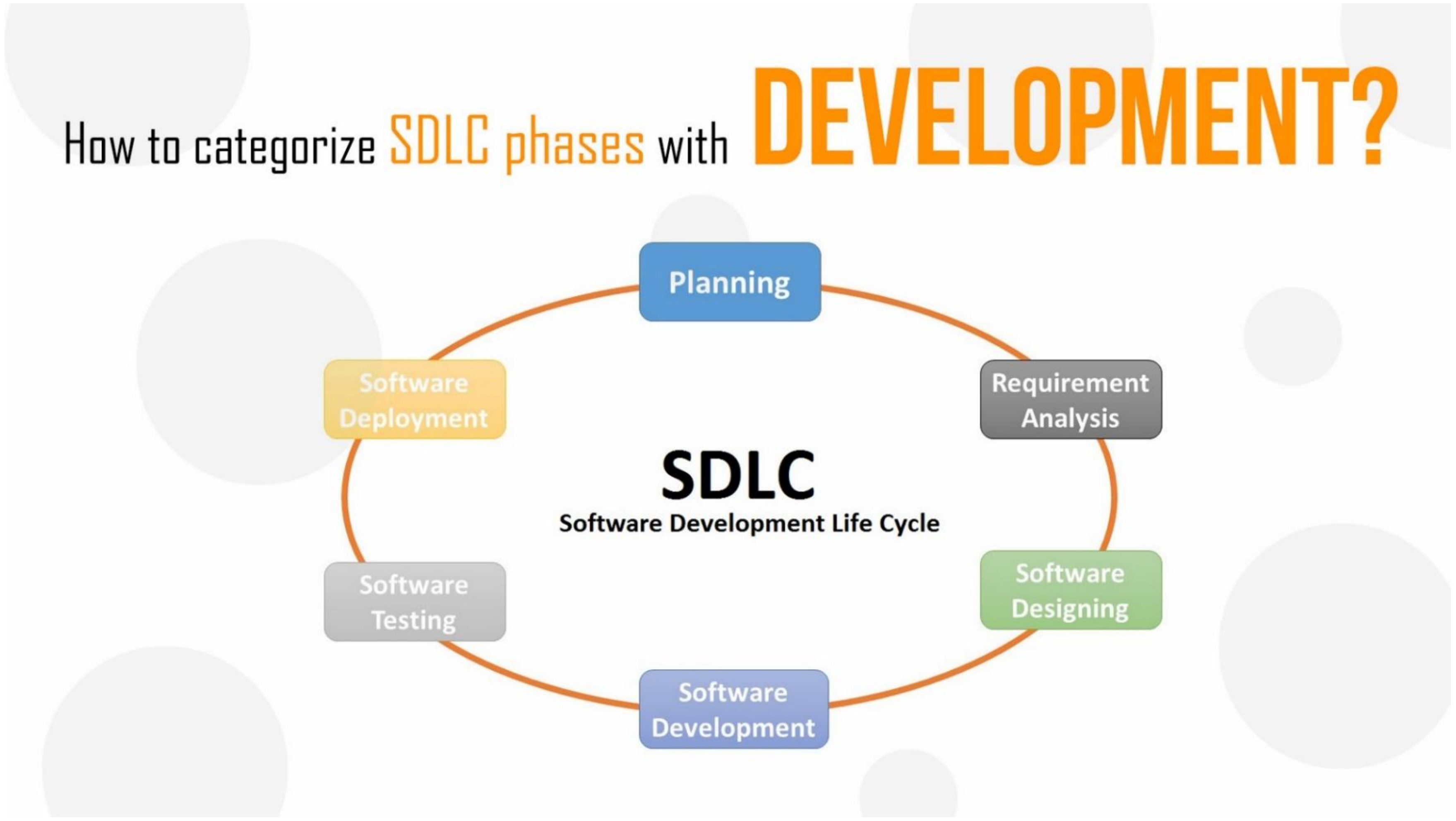
Software Requirements Engineering (SRE)

Ahad Zareravasan, PhD
Masaryk University, Brno, Czech Republic
Email: Zare.Ahad@mail.muni.cz

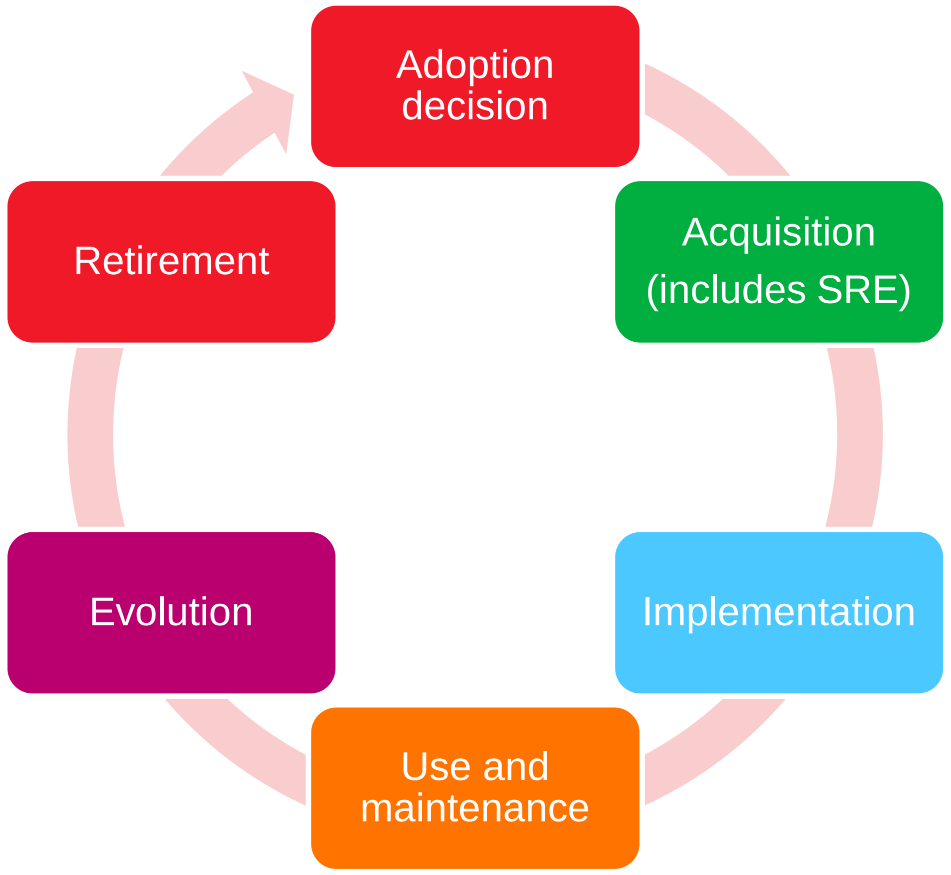
Learning objectives

- Describe the SDLC and Life Cycle Model for System Acquisition
- Explain the necessity of SRE
- Explain the concept of software requirements and different types
- Create a draft of software requirements document for the case study

Software Development life Cycle (SDLC)

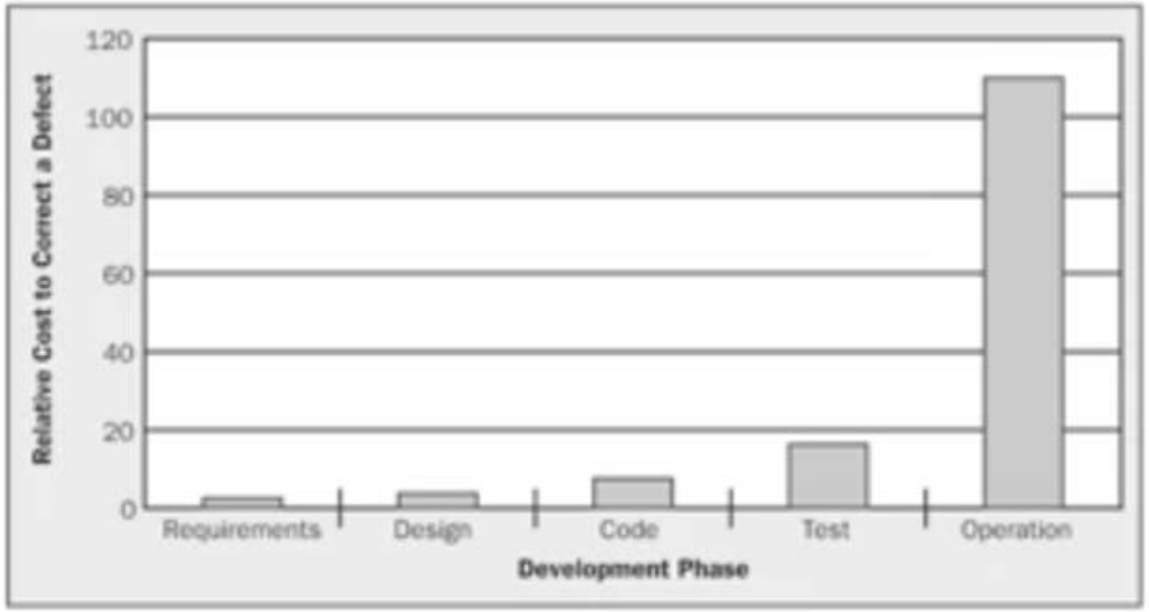


A Life Cycle Model for System Acquisition



MUNI ECON

Cost of Bad Requirements



- Rework: 30 to 50% of total development cost (Boehm and Papaccio, 1988)

- 70 to 85% of rework cost from requirements errors (Leffingwell, 1997)

What are requirements

- *“anything that drives design choices”*
- *Requirements are a specification of* ***what should be implemented****. They are descriptions of* ***how the system should behave****, or of a* ***system property or attribute****. They may be a* ***constraint*** *on the development process of the system.*

MUNI ECON

Types of software requirements

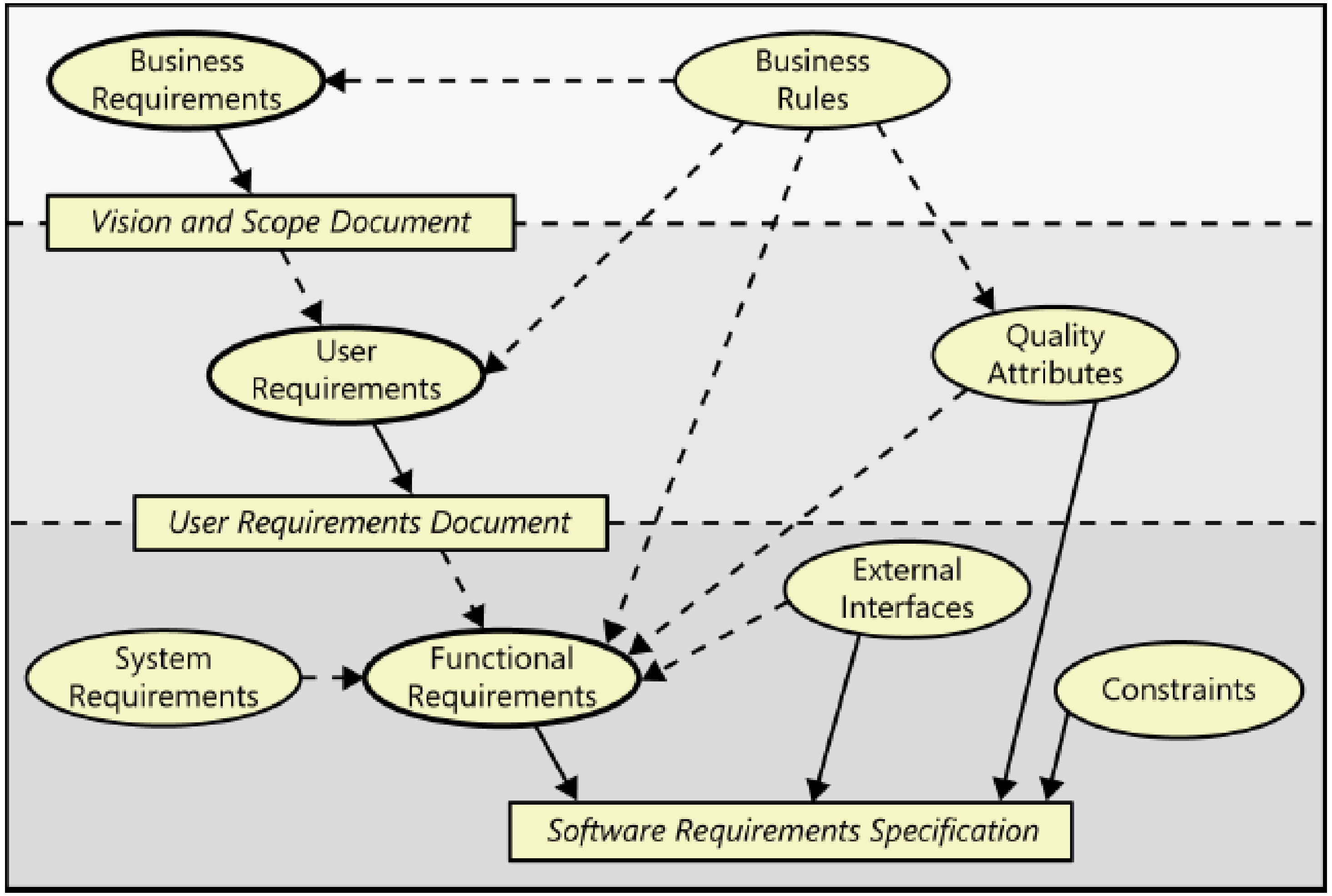
- **Business requirements** are high-level business objective of the organization that develop a product or of a customer who want to pay for a software.
- **Business rule** include corporate policies, government regulations, industry standards, and computational algorithms.
- **Constraint** is a restriction that is imposed on the choices available to the developer for the design and construction of a product.
- **User requirement** is a task or a set of activities that specific users must be able to do using a system.

Types of software requirements

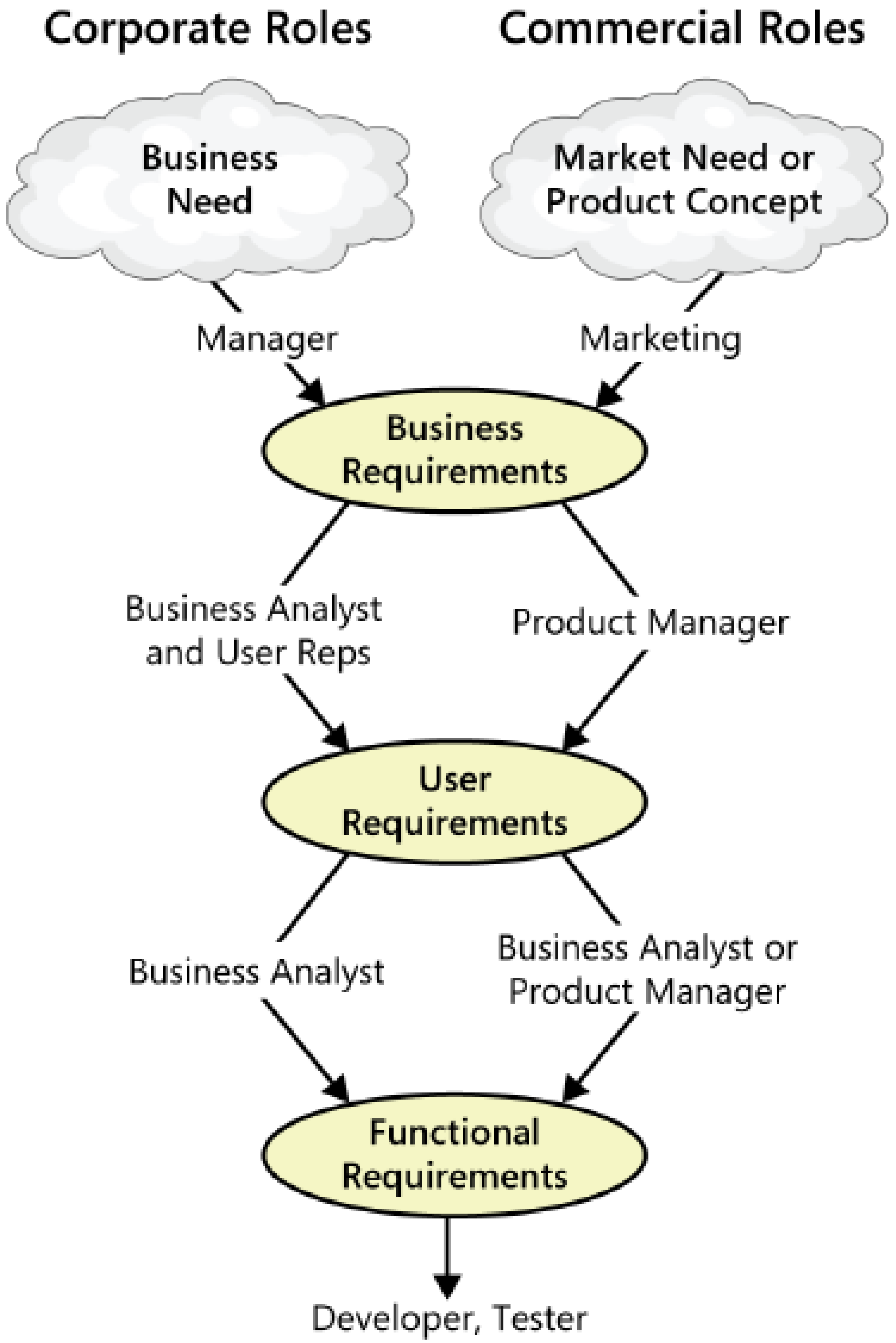
- ❑ **Functional requirement** is a description of system behavior under specific conditions. (**what** the system should do)
- ❑ **Nonfunctional requirement** is a description of a property or a feature or a characteristic of a system, or **how** should the system do or behave?
- ❑ **System requirement** is a top-level requirement for a product that contains multiple subsystems, which could be all software or software and hardware.

MUNI ECON

Relationships among software requirements



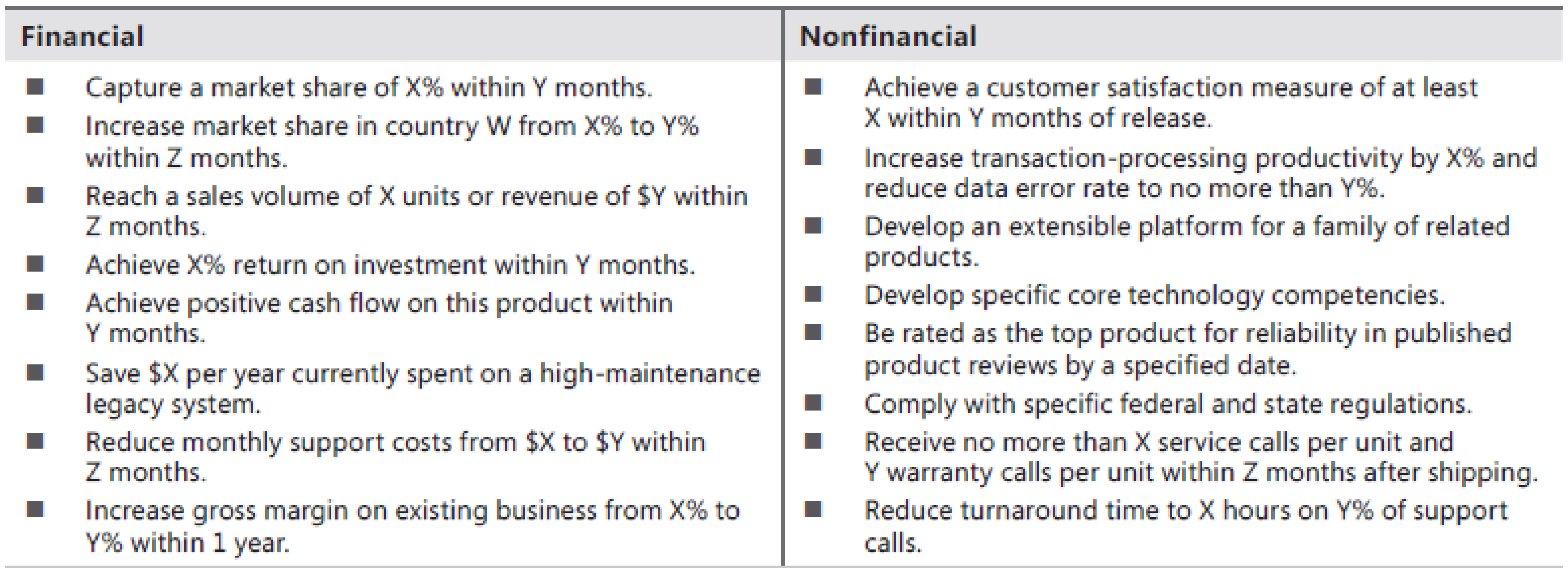
An example of how different stal
requirements development



Business requirements

- **Start with business background**
- **Explore a Business opportunity**
- **List Business objectives (SMART)**
- **Scope**

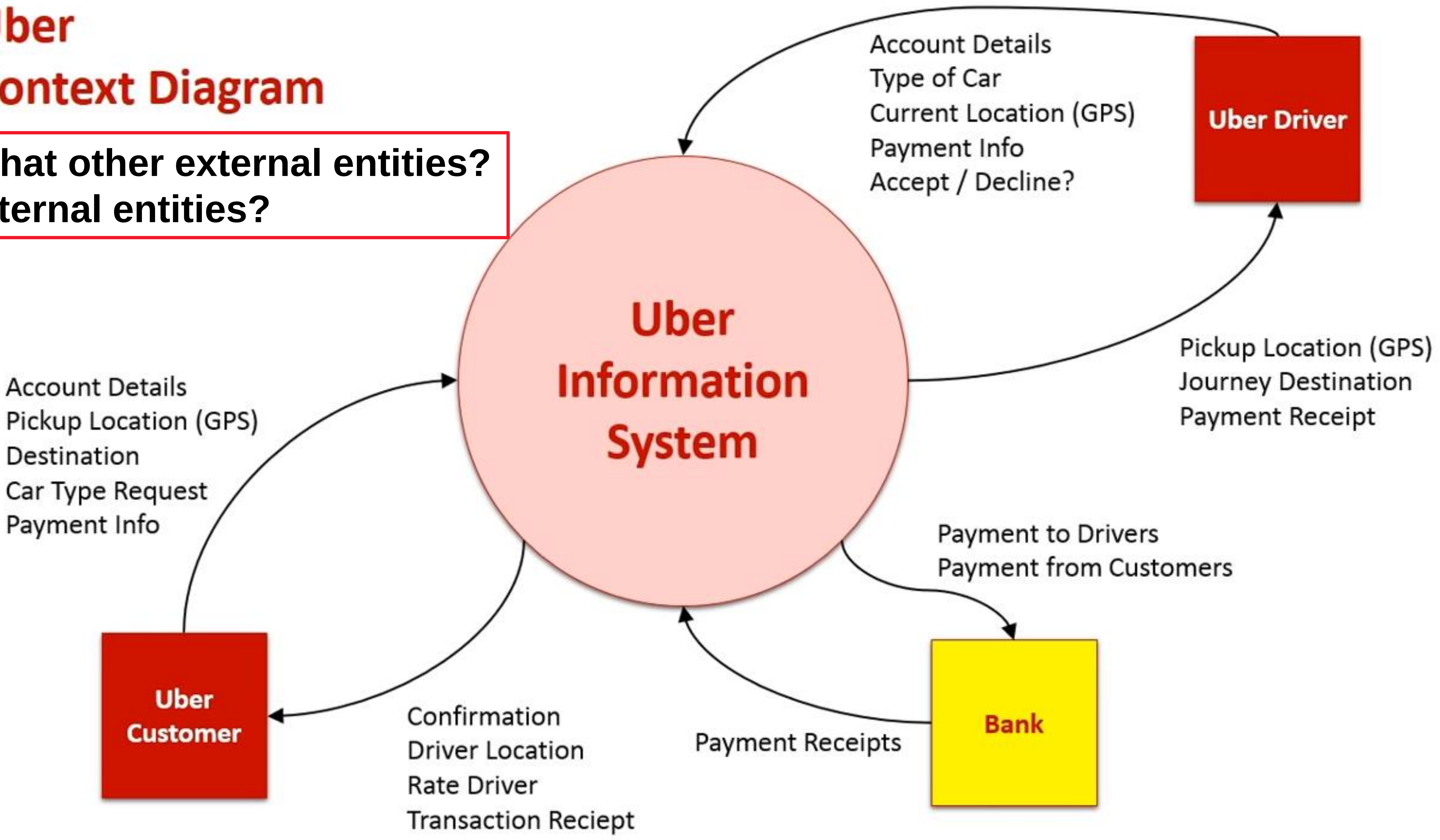
Platitudes ("become recognized as a world-class <whatever>") and vaguely stated improvements ("provide a more rewarding customer experience") are neither helpful nor verifiable.



Financial	Nonfinancial
■ Capture a market share of X% within Y months.	■ Achieve a customer satisfaction measure of at least X within Y months of release.
■ Increase market share in country W from X% to Y% within Z months.	■ Increase transaction-processing productivity by X% and reduce data error rate to no more than Y%.
■ Reach a sales volume of X units or revenue of $Y within Z months.	■ Develop an extensible platform for a family of related products.
■ Achieve X% return on investment within Y months.	■ Develop specific core technology competencies.
■ Achieve positive cash flow on this product within Y months.	■ Be rated as the top product for reliability in published product reviews by a specified date.
■ Save $X per year currently spent on a high-maintenance legacy system.	■ Comply with specific federal and state regulations.
■ Reduce monthly support costs from $X to $Y within Z months.	■ Receive no more than X service calls per unit and Y warranty calls per unit within Z months after shipping.
■ Increase gross margin on existing business from X% to Y% within 1 year.	■ Reduce turnaround time to X hours on Y% of support calls.

Uber
Context Diagram

What other external entities?
Internal entities?



Context diagram building process

- Identify the system and boundaries (the context)
- Identify external entities
- Identify external flows (inputs, outputs)

– **Note**:

- The whole system itself is a process. (it receives input and transform it to output)
- Inside the system is a black box.
- Relationships between external entities are not our concern.

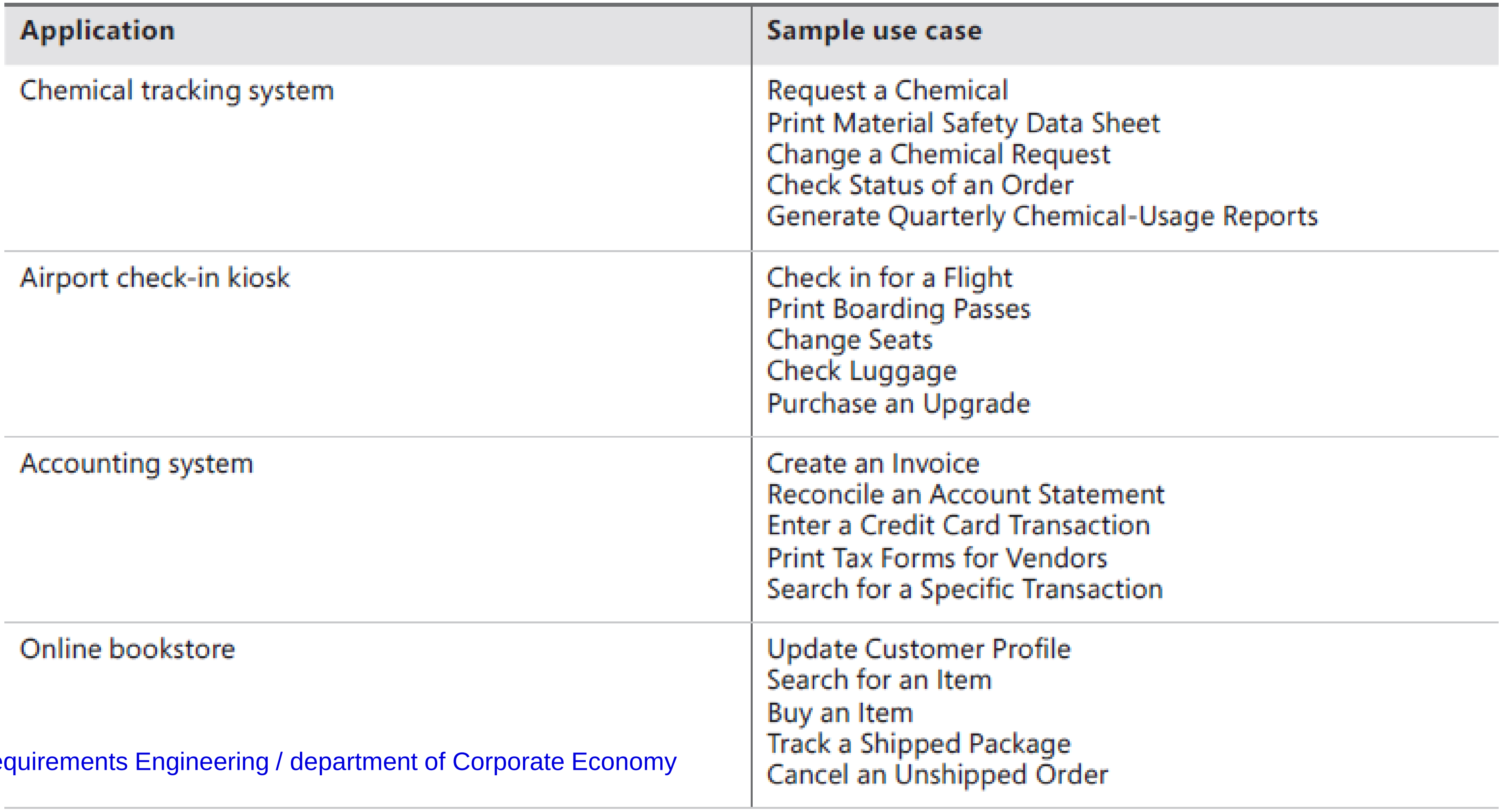
MUNI
ECON

User requirements: use cases

- A **use case** describes a sequence of interactions between a system and a user that results in the actor being able to achieve some outcome of value.
- The names of use cases are always written in the form of a verb followed by an object (use strong, descriptive names).

MUNI
ECON

User requirements: use cases

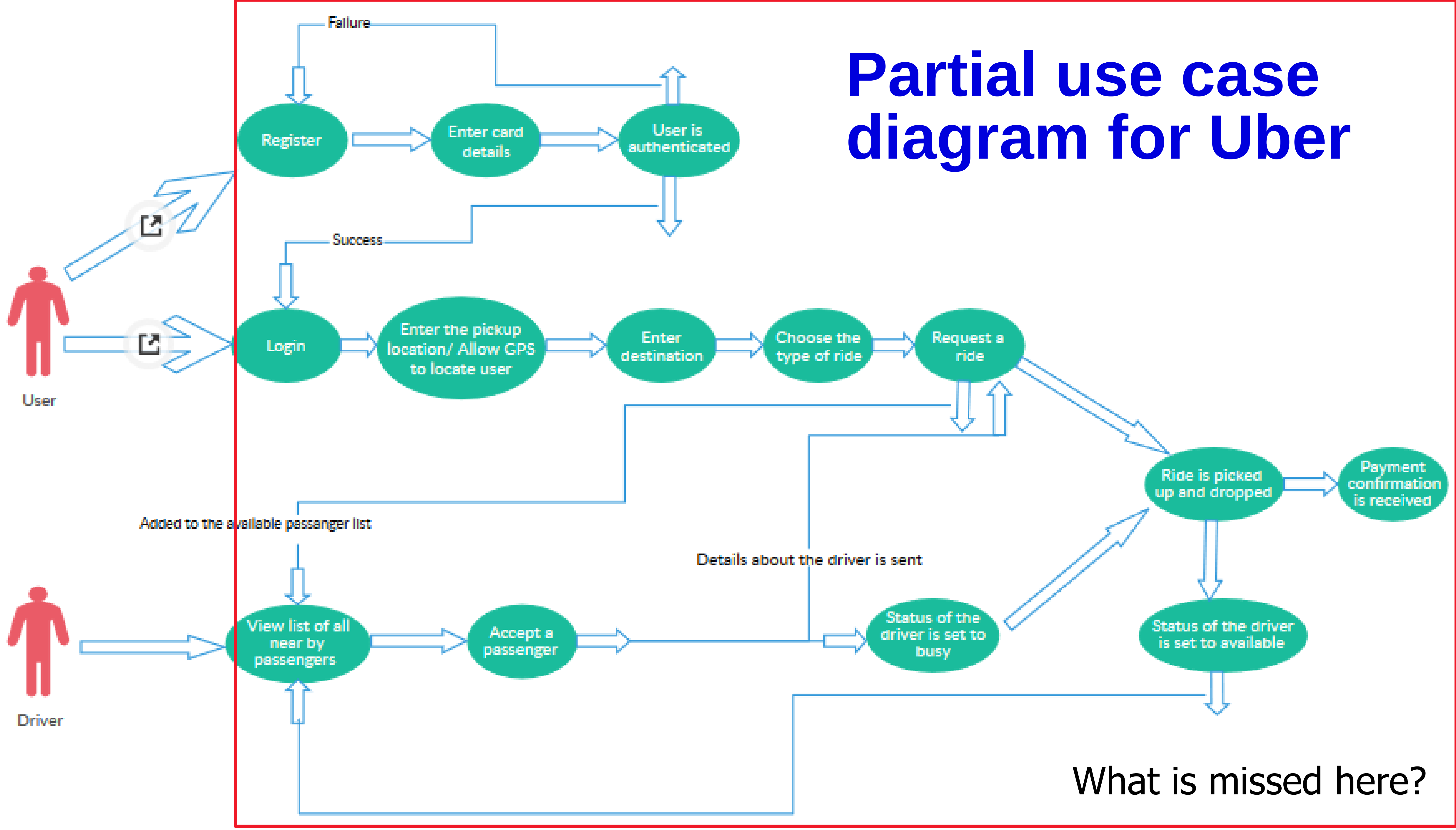


Application	Sample use case
Chemical tracking system	Request a Chemical Print Material Safety Data Sheet Change a Chemical Request Check Status of an Order Generate Quarterly Chemical-Usage Reports
Airport check-in kiosk	Check in for a Flight Print Boarding Passes Change Seats Check Luggage Purchase an Upgrade
Accounting system	Create an Invoice Reconcile an Account Statement Enter a Credit Card Transaction Print Tax Forms for Vendors Search for a Specific Transaction
Online bookstore	Update Customer Profile Search for an Item Buy an Item Track a Shipped Package Cancel an Unshipped Order

Question

– Name some use cases for Uber …..

Partial use case diagram for Uber


Failure
Register
Enter card details
User is authenticated
Success
Login
Enter the pickup location/ Allow GPS to locate user
Enter destination
Choose the type of ride
Request a ride
User
Ride is picked up and dropped
Payment confirmation is received
Added to the available passanger list
Details about the driver is sent
Driver
View list of all near by passengers
Accept a passenger
Status of the driver is set to busy
Status of the driver is set to available

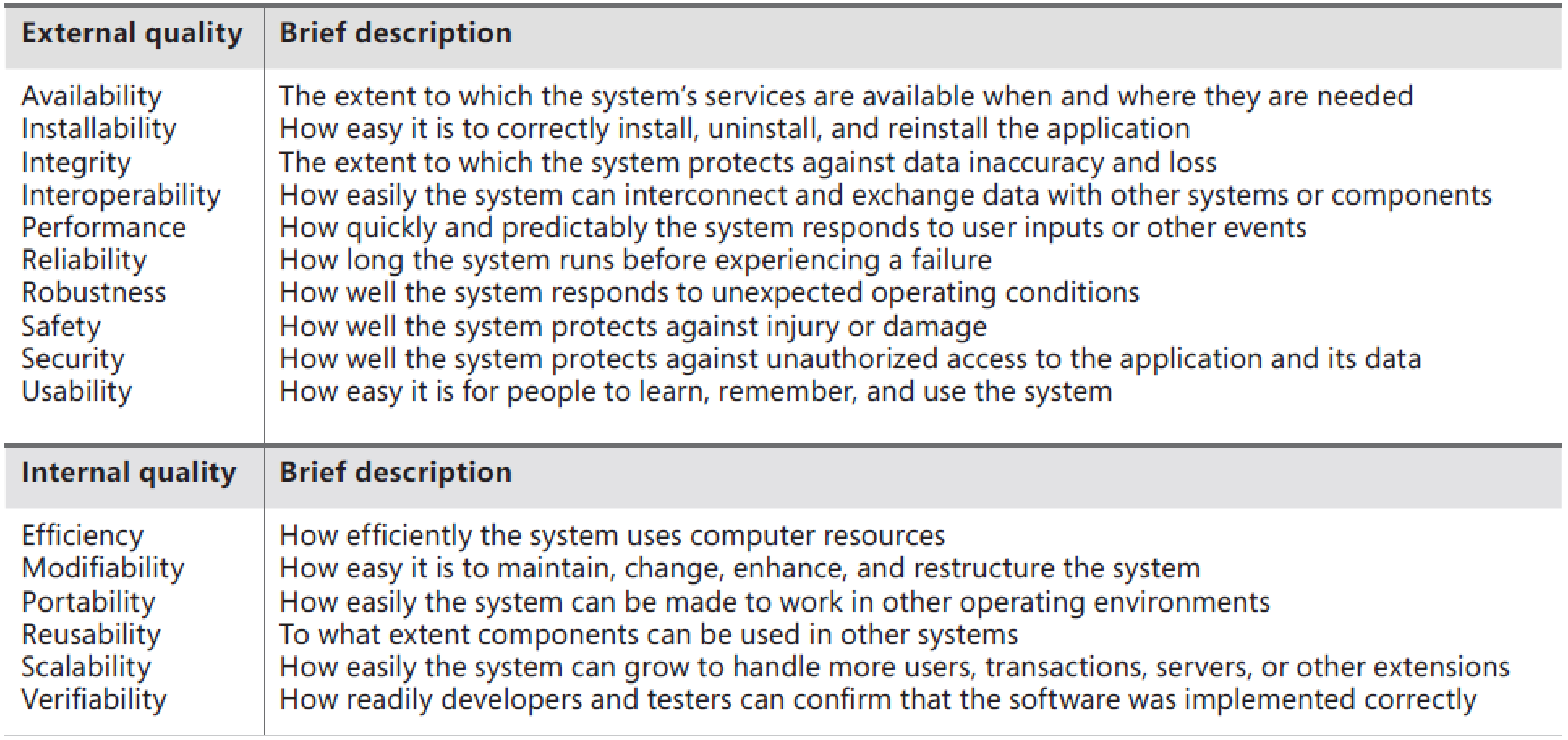
What is missed here?

Functional requirements

- **what** the system should do (so that the use cases could happen)
- Functional requirements are derived mainly from user requirements, and business rules.
- Example: Functional requirements related to Uber registration use case:
- 1. Users (customers) have to register in the system, using their full name, address, email, and mobile number information.
- 2. The registration of users must be validated.
- 3. The password must have at least one special character and be at least 8 characters.
- 4. System must not let users to register twice using a same email or mobile number

Nonfunctional requirements

how the system works



External quality	Brief description
Availability	The extent to which the system's services are available when and where they are needed
Installability	How easy it is to correctly install, uninstall, and reinstall the application
Integrity	The extent to which the system protects against data inaccuracy and loss
Interoperability	How easily the system can interconnect and exchange data with other systems or components
Performance	How quickly and predictably the system responds to user inputs or other events
Reliability	How long the system runs before experiencing a failure
Robustness	How well the system responds to unexpected operating conditions
Safety	How well the system protects against injury or damage
Security	How well the system protects against unauthorized access to the application and its data
Usability	How easy it is for people to learn, remember, and use the system

Internal quality	Brief description
Efficiency	How efficiently the system uses computer resources
Modifiability	How easy it is to maintain, change, enhance, and restructure the system
Portability	How easily the system can be made to work in other operating environments
Reusability	To what extent components can be used in other systems
Scalability	How easily the system can grow to handle more users, transactions, servers, or other extensions
Verifiability	How readily developers and testers can confirm that the software was implemented correctly

Nonfunctional requirements

– Example: Non-functional requirements related to Uber registration use case (how fast, secure, reliable it is):

1. The verification email/text message must be sent within 5 seconds.
2. The verification link/code must be expired in 24 hours.
3. The system should be able to manage 1000 concurrent user registrations.
4. The privacy of user information must be preserved.

Assignment

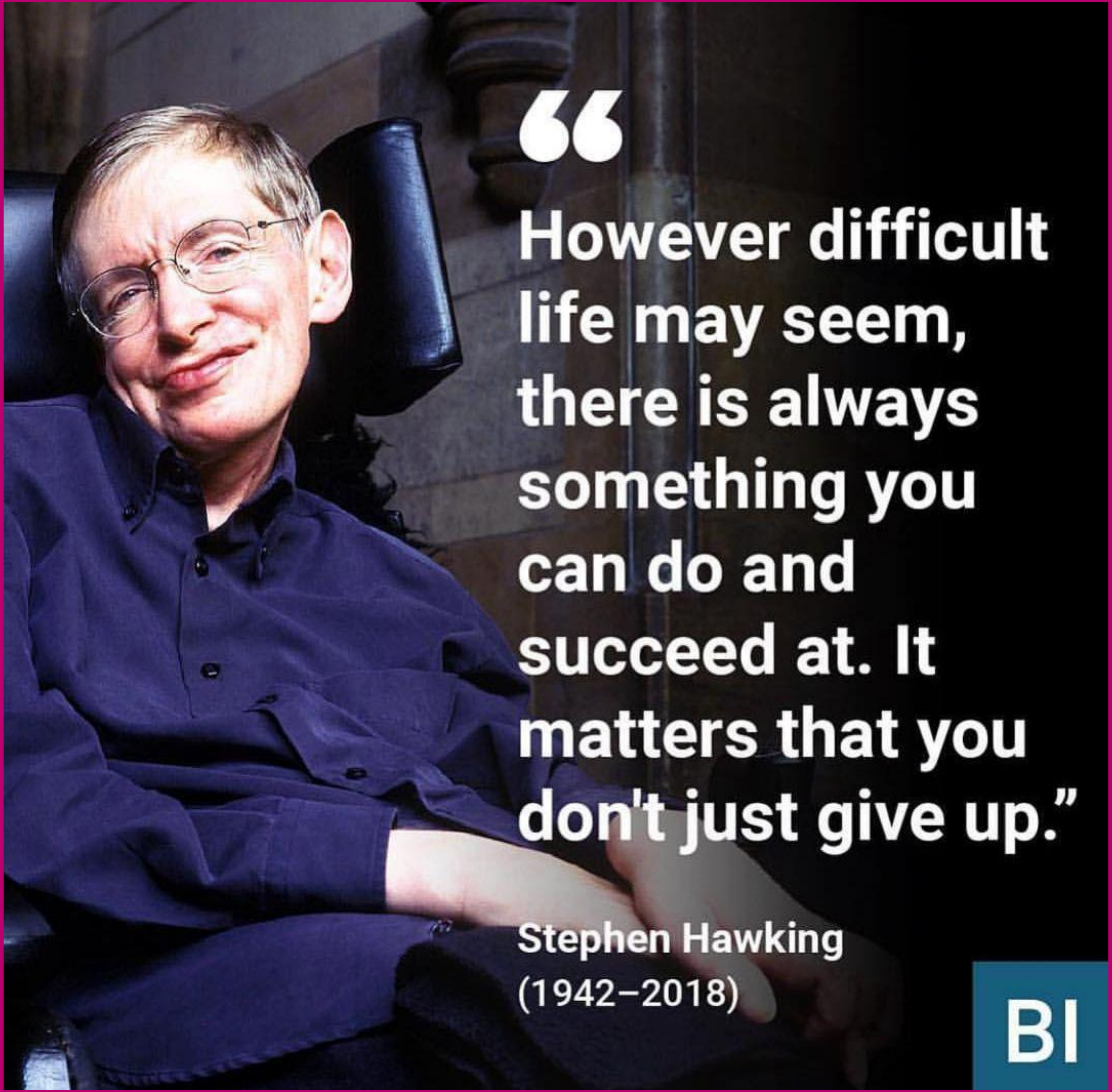
1. Write your understanding of the business problem/opportunity (p 83), and business objectives (p 84) for the proposed system in the case.
2. Draw the system's context diagram, considering the internal and external entities to the system (p 92-93). Describe it and explain why do you label each entity as internal or external.
3. Identify at least four use cases (indicating the relevant actor(s), p 149).
4. Identify and describe at least four functional requirements relevant to two aforementioned use cases and four non-functional requirements (p 147-149).

Follow the instructions for submission

Wrap up

- We learnt about the life cycle of software projects
- We learnt about the concept of software requirements engineering and why do we need it in software projects
- We learnt about different types of software requirements

MUNI ECON


“However difficult life may seem, there is always something you can do and succeed at. It matters that you don't just give up.”
Stephen Hawking
(1942–2018)
BI

Thank you for your attention!

Questions, comments or remarks?

Email: Zare.Ahad@mail.muni.cz

MUNI ECON