Define the requirements of a project

Business Analysis

Analysis Skills : 3

At first glance, the definition of needs and requirements may mean the same thing. And yet, a subtle difference exists. The whole role of the business analyst will be to identify the needs that will become the project requirements.

Published Sunday May 30th 2021, 11:39
Modified Monday August 26th 2024, 10:04

By Olivier Paudex

Introduction

Needs and requirements should not be confused. Although at first glance, the two may mean the same thing. If a need is an unmet goal, a requirement is the decision to do something to achieve that goal.

RADD (Requirement Analysis and Design Definition)

The **RADD** is the section concerning the classification of requirements and their prioritizations as seen by the **BABOK**.

It allows you to have control and overview over all the project's features and to know the progress of the project.

A **PDF** attached to this publication, in the form of a data stream, summarizes all the tasks, their descriptions, tools, stakeholders involved, as well as mnemonic techniques. It has been produced for educational purposes.

Defining a requirement

Bringing together the two definitions above, a need becomes a requirement the moment it is decided that it must be met.

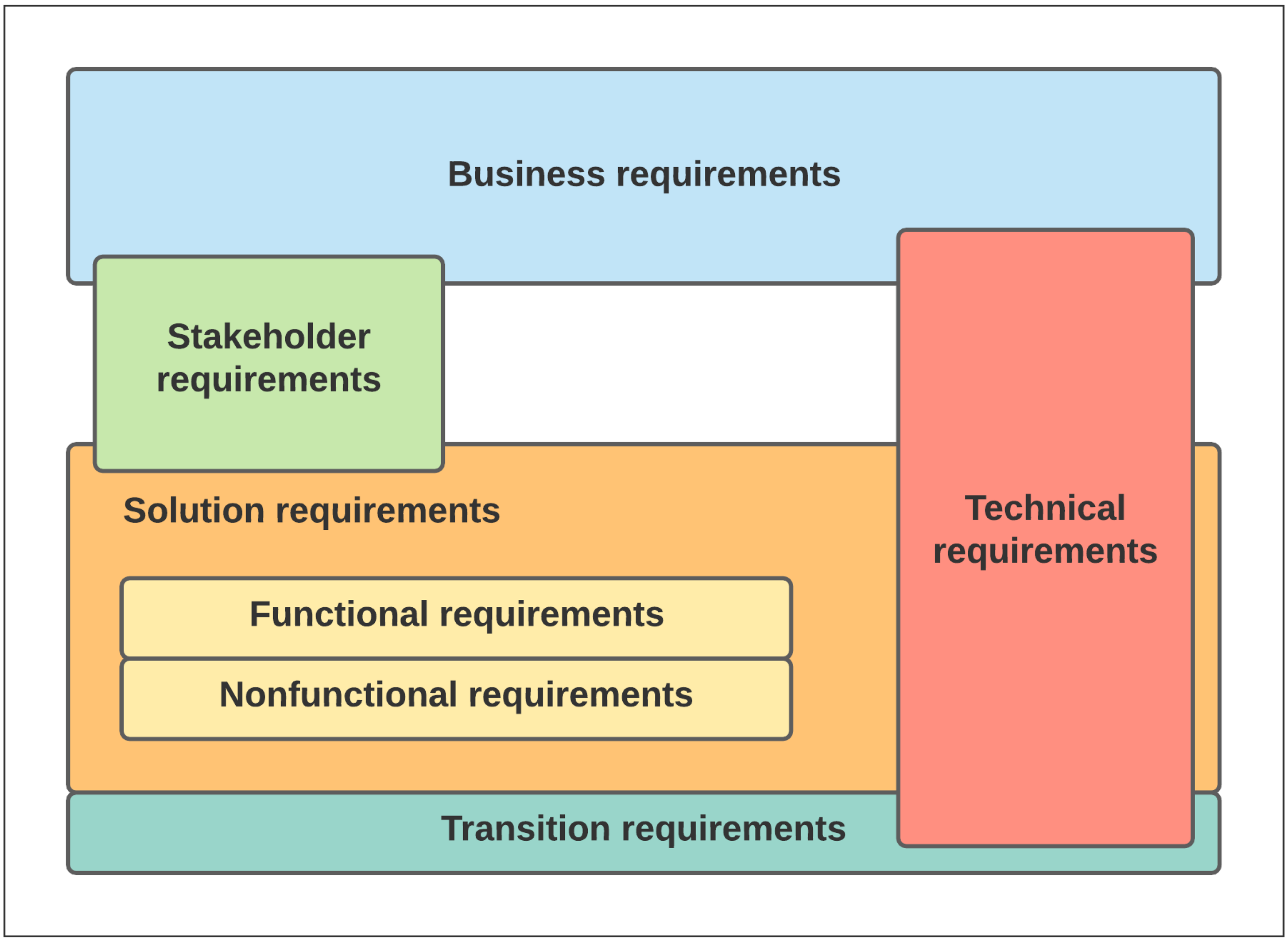
And to take it a step further, it is safe to say that a requirement is :

- a feature or skill that someone needs to solve a problem.
- a feature or skill that a solution must provide to satisfy a standard or specification.
- a representation of said feature or skill.

Categorize the requirements

There are at least five types of requirements, including :

- Business requirements.
- User requirements.
- Solution requirements, which can be divided into two further categories, functional and non-functional requirements.
- Technical requirements.
- Transition requirements.



The different requirements

The business requirements

These are the elements that must be put in place to serve the interests of the business.

User requirements

These are the items a user needs to complete a task.

User requirements also describe the needs of the business, its goals and objectives, but from a user perspective.

When writing them, you should :

- Do not influence users.
- Get consensus to define the requirement, should disagreement arise.
- Take all opinions into account.
- Try not to frustrate users.

Solution requirements

- They should give an overview of the solution but not define it.
- Don't focus on the requirements of a software. This could result in cool features, but not really necessary.
- Keep it simple, without getting carried away with the features.
- Stay focused on what the solution needs to achieve.

The functional requirements

Functional requirements describe the state of a process.

- They describe how the solution should behave.
- What are the expected responses of the solution ?
- What are the operating rules ?
- What are the functions or features ?
- What must the user do to use the solution ?
- Is the expected result correct ?

The non-functional requirements

Type of questions to ask to define non-functional requirements :

- Is it easy to use ?
- Is it reliable ?
- Is it powerful ?
- Is it safe ?
- Is it pretty ?
- Is it a good design ?
- Is it intuitive, easily accessible ?

- What are the technical features? How much memory is needed to enable this feature ?

Transition requirements

- Can we switch from the current state to a new state ?
- Is the system redundant ? Can we afford to take it offline for a while ?
- Do we need to revisit the hardware ?
- Do we need to think about training the staff ?

Technology requirements

Last step to take when solution has been described.

- See or review the design.
- See or review the architecture.
- Choose a programming language and code.
- Know where the data will be stored.
- Know how the data will be displayed.

> Technology should always have a backseat to the data. You can always change the way the data is presented, but not the data itself.

Write requirements

The excellence of a requirement is matched only by its clarity. If you have no questions about a requirement, it’s perfect.

A requirement must :

- **Be complete** : describe your goal and how to achieve that goal.

> **Incomplete example** : *Accounting needs to be able to process employee expenses*.

> **Complete example** : *Accounting supervisors need to be able to validate expenses submitted by employees*.

You always have to know precisely who you’re dealing with. Accounting isn’t specific enough. Just as the verb **“process”** does not describe the action, unlike the verb **“validate”**. Employee expenses don’t mean anything, unlike submitted expenses that define expenses that have been entered by employees, an expense report for example.

- **Be correct** : achieve the goal and meet the user’s desires.

> **Incorrect example** : An employee can change her last name after a marriage.

> **Correct example** : An employee can change her last name after her change has been legally validated.

- **Be unambiguous** : A requirement should be crystal clear.

> **Ambiguous** : Overtime is not allowed.

> **Non-ambiguous** : A work hours sheet that exceeds 40 hours/month will not be accepted and will be returned to the consultant for correction.

- **Be verifiable and testable**.

> **Not verifiable** : Order processing must be able to pack most orders within 5 minutes.

> **Verifiable** : Order processing must be able to package 90% of orders within 5 minutes of receiving the fulfillment form.

- **Be required** : It must describe a purpose, not a user's intent.

> **Not Necessary** : The cashiering system must have its exchange rate updated in real time.

> **Required** : The cashiering system must have its exchange rate updated daily.

- **Be feasible** : To be validated with the technical team.

> **Not feasible** : Security software must capture the reason for each intrusion attempt.

> **Feasible** : Security software must capture the date, time, and IP address of each intrusion attempt as specified in the suspected intrusion criteria.

- **Prioritized** : Based on value, risk level, and frequency of use.

Communicating requirements

There are a thousand and one tools for communicating requirements to other project stakeholders.

- The easiest way of course being to do it in an **"Excel-like"** table with the **"Kanban"** method.

Requirement Description	To do	In progress	To be verified	Done	Date
The system must be able to …	X				
The processing of orders must …		X			
A customer has the possibility to …				X	01-01-2010

A Kanban-type board to record requirements

- There are computer solutions, the most well-known of which are **JIRA** and **Trello** from the company Atlassian.
- These applications are called the **backlog**. They typically allow for prioritization of requirements and collaboration with multiple stakeholders.

Don't forget anything ...

Here are some helpful tips for not missing a point that would be worth writing a requirement.

Cascading the requirements at hand ensures that nothing is missed.

Always ask the question **"What ?"** helps define the business or stakeholder requirements.

- What does the business need describe ?
- What is the main problem ? Do we cover it ?
- What is the objective to be achieved ? Are we achieving it ?
- What information do we have available to write the requirements ?

Always ask the question **"How ?"** to define the requirements of the solution.

- How and where is the data stored ?
- How is the process executed and by whom or what ?

This provides a quick check of the requirements and, more importantly, avoids the temptation to get distracted by the technology.

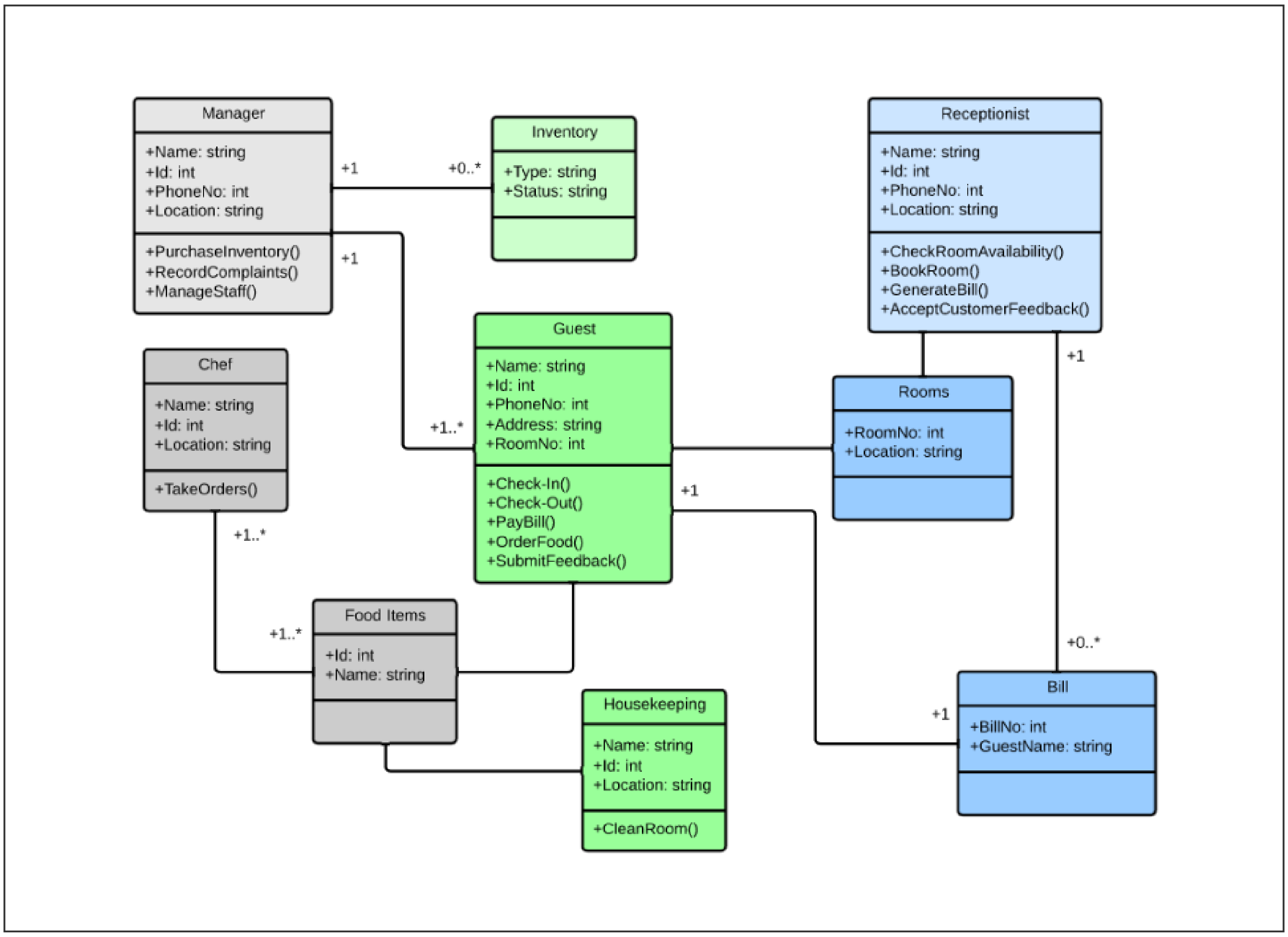
The 4 main components

The requirements are divided into 4 usage groups:

- Data.
- Processes.
- External agents.
- Business rules.

The data

- The definition of the data must be precise. We must specify their origins, their purposes, their functions.
- The data are stored, in most cases, in a database.
- The data are represented in their logical and physical forms in a UML diagram.
- The data are represented in the form of tables, columns. On leur attribue des cardinalités et des relations.



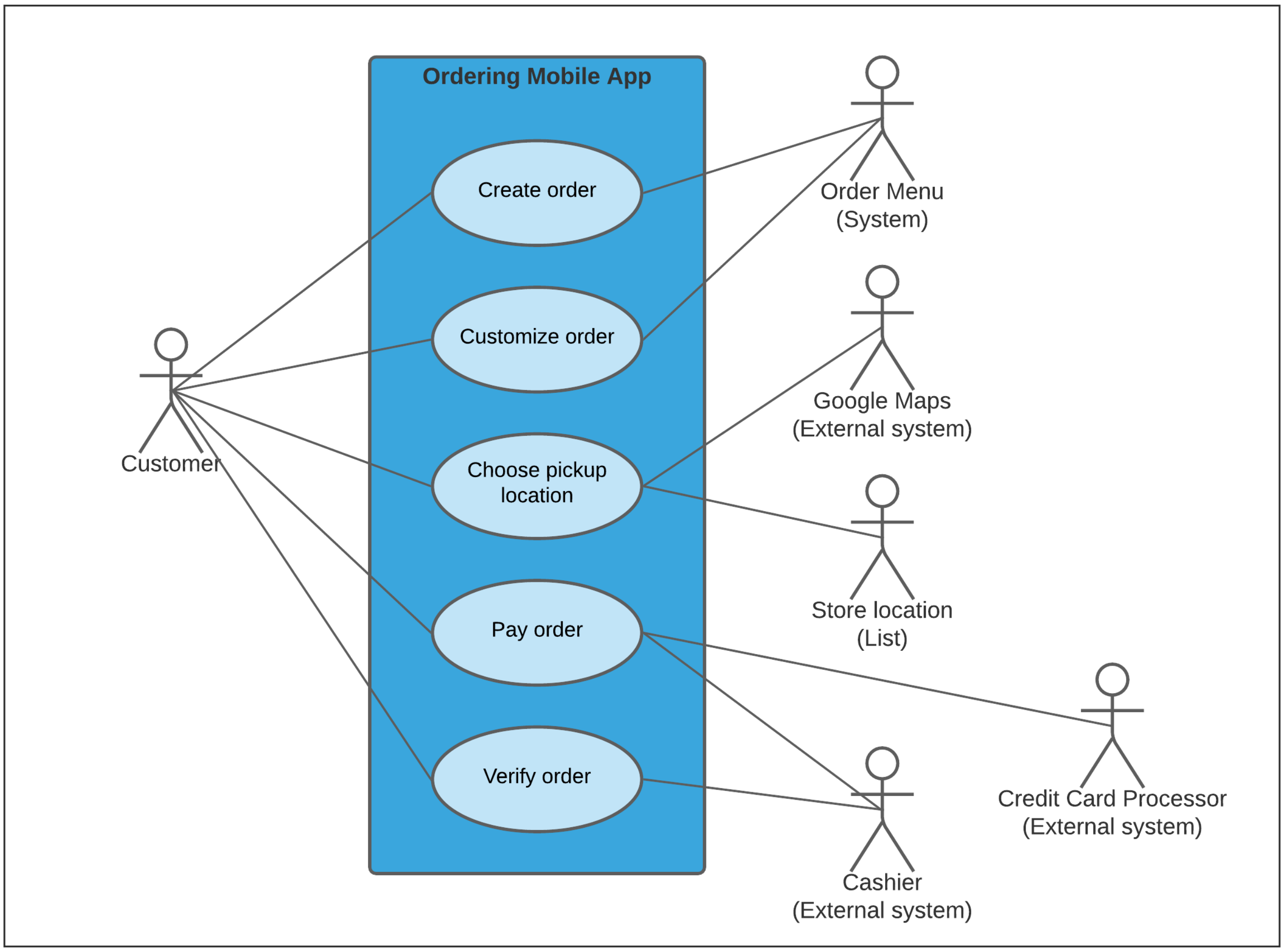
A UML diagram representing data

The processes

A process is an action performed for the purpose of accomplishing a task. It can be automatic or manual.

Example: Pay bill, find regional doctor.

- To define processes, use cases (**use cases**), user stories (**user stories**), flow diagrams (**data flow**) are used.
- Processes are described by the syntax **"noun + verb"**.



Example of use cases for an online ordering system

External agents

These are the first actors or system, identified in a requirement.

Example : A customer must insert his or her bank card into the ATM, enter his or her code, the desired amount, press the OK button, withdraw the money and his or her card.

- Actors intervene on the solution using an interface.
- The requirement must include the data transformed using an interface.

The Business Rules

A business rule defines how a task is performed. The rules define the actor, the process, and the data. They should be written in detail.

Think of exceptions

A customer receives a loyalty gift after 90 days of registration. What happens after two registrations longer than 90 days with the same account.

Think of cardinalities

- Is it possible to have no subscription ?
- Is it possible to have multiple subscriptions with the same account ?

Think of it as a requirement

- The password must be correct or the procedure will go no further.

Conclusion

Defining requirements based on needs can be very time consuming. **The only task for the BA is to write requirements that are clear and unambiguous**. If a requirement contains any doubt as to its interpretation, then the entire project can be compromised. The BA can get help from other stakeholders, primarily the SME. While drawing complex diagrams (**UML, BPMN, etc.**) can be done by the technical people, the BA must understand them and be able to communicate them transparently.

The next chapter will focus on **verification and validation of the solution**.