

Requirements and Business Analysis

Chapter 1 - Requirements: Basic concepts & definitions

v.2024.04.19

<https://requirements.university>

OFFICIAL
REFERENCE:

<https://requirements.university>

HOW TO CITE:

“Jean-Michel Bruel, Handbook of Requirements and Business Analysis Teaching Materials.
<https://requirements.university>.”



If you have any content that I did not reference well or that should be removed, please do not hesitate to contact me so that I can correct this presentation.



About the slides' author

- Professor at Toulouse University
 - Teaching [modeling](#), [requirements](#) and [DevOps](#)
- Member of the CNRS-IRIT Laboratory
 - Model-Based [Systems Engineering](#)
- Leader of the [companion book](#)

<https://bit.ly/jmbruel>



Get the slides (pdf)

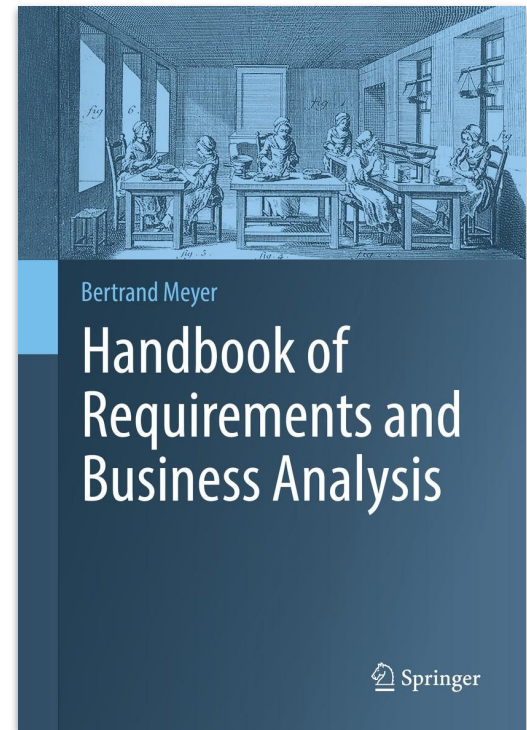
Disclaimer



This material is based on this book, by Bertrand Meyer.

But it only reflects the point of view of its author, especially in these slides about the graphical representation of the concepts.

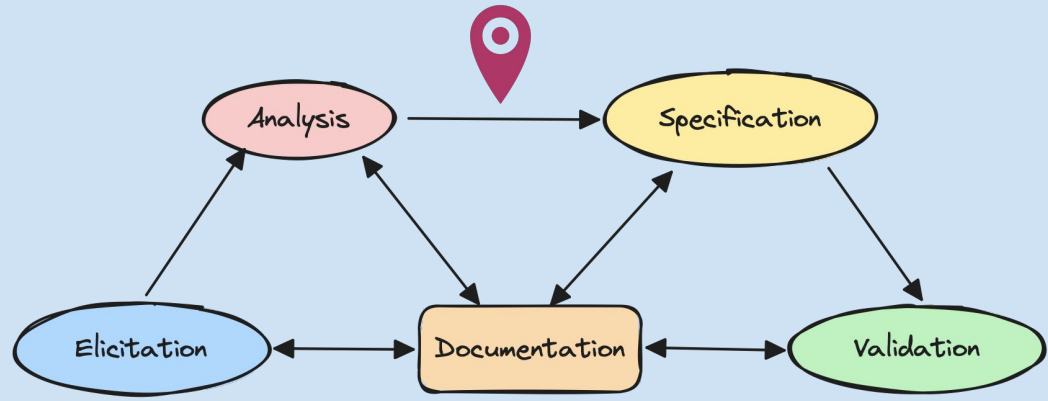
It is part of additional materials developed and available at <https://requirements.university>



<https://se.inf.ethz.ch/requirements/>

Outline

1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements



Outline

1. Dimensions of RE

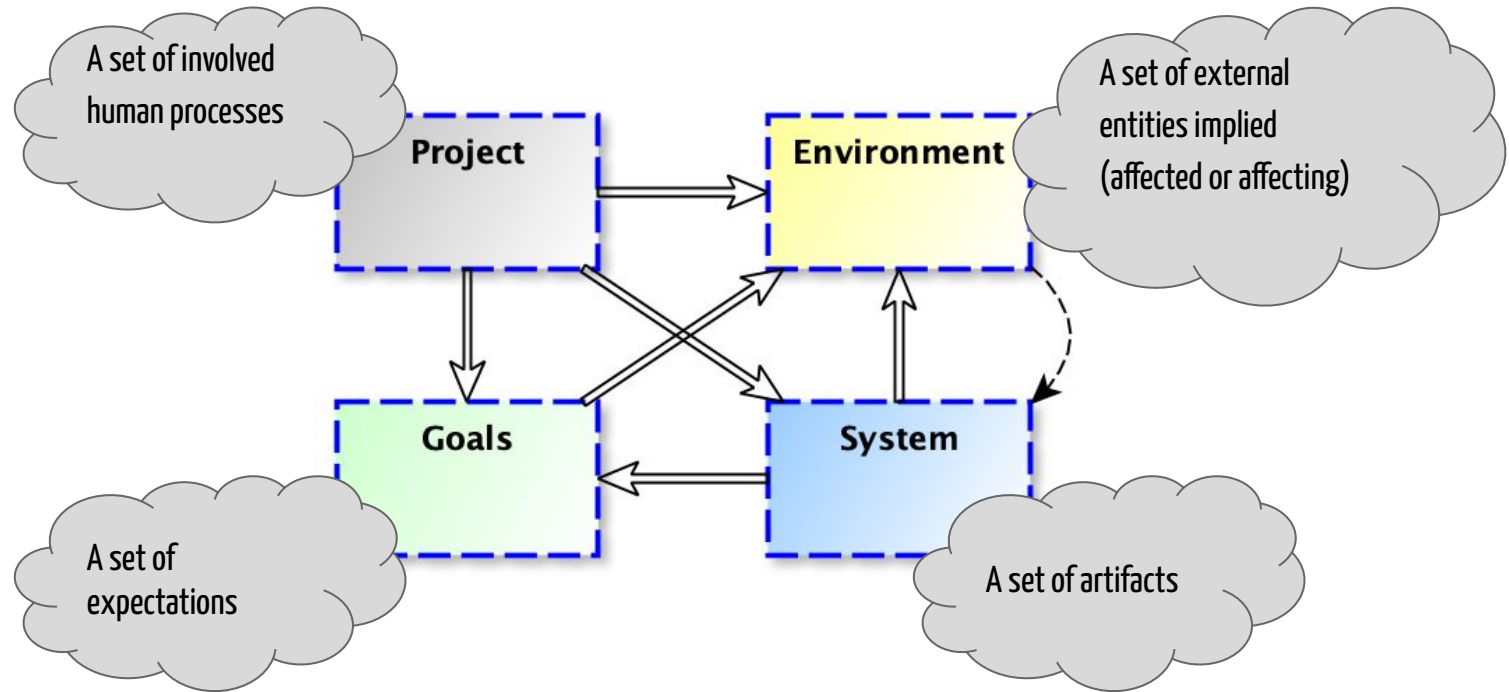
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements





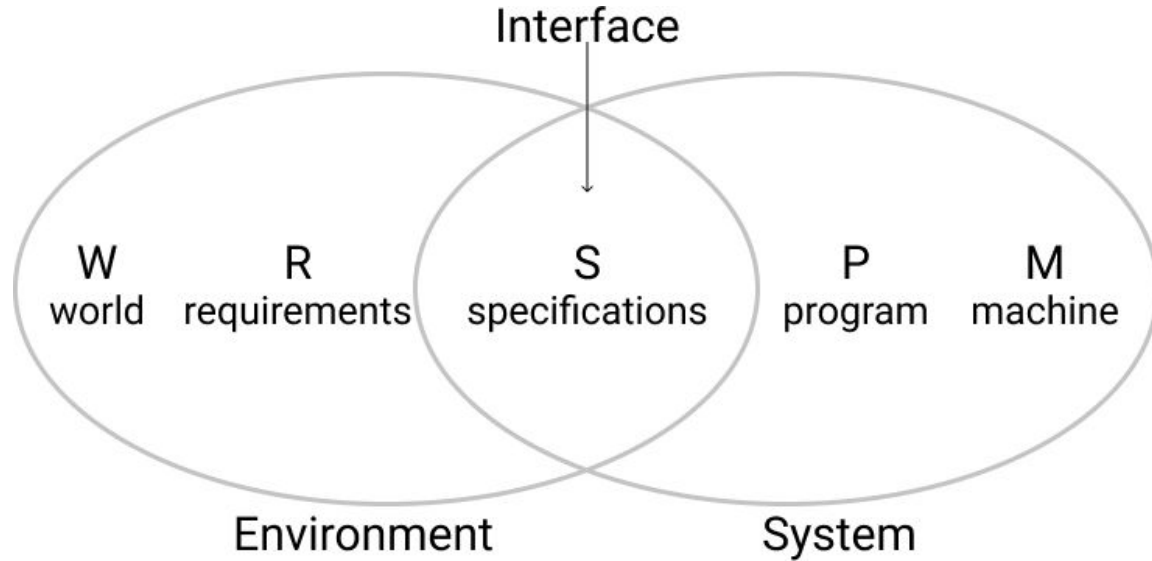
Context (universe of discourse)

“a project to develop a system, in a certain environment, to satisfy a set of goals”



System vs. Environment

Importance of interfaces



[M. Jackson and P. Zave, 1995]

Target organization & Production organization

Customer/Developer

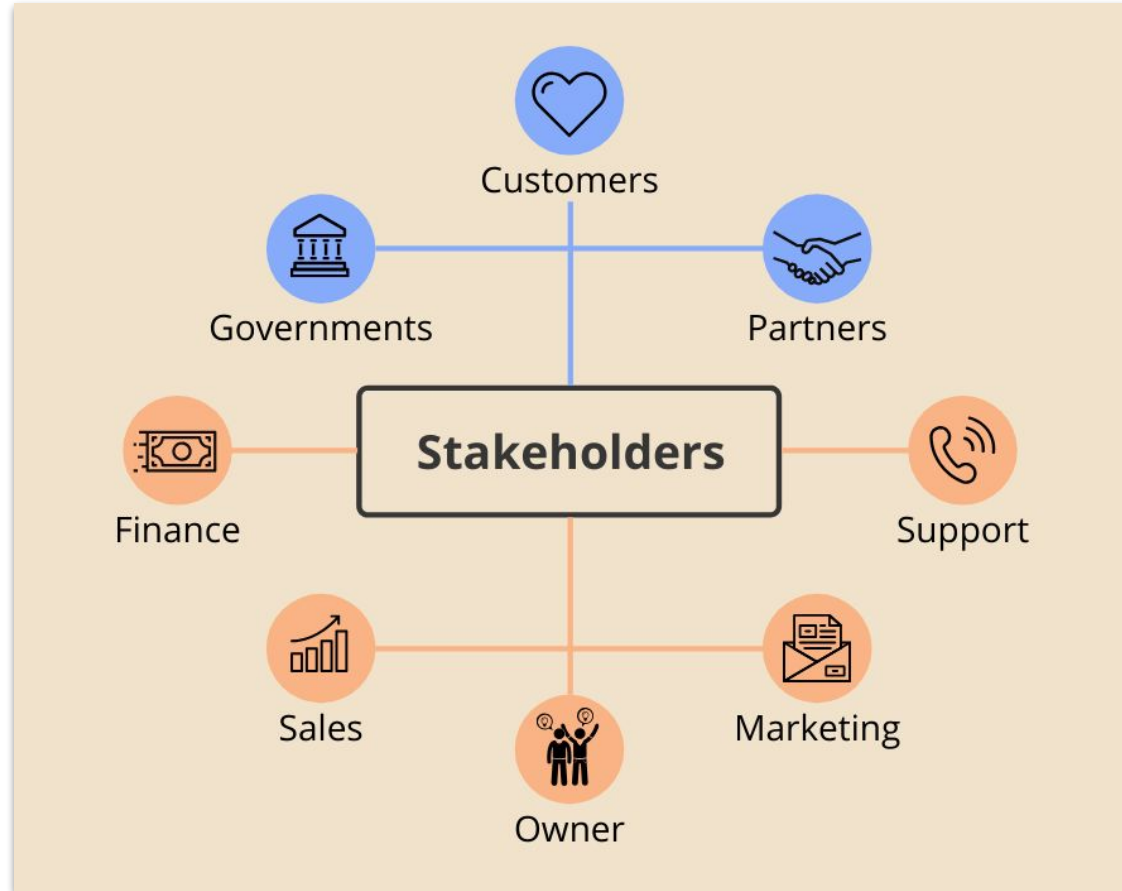
MOA/MOE

...



Stakeholders

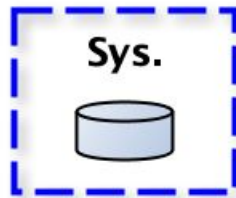
“For Who”



Additional concepts

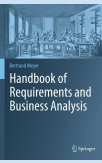
We distinguish the different **stages** of a **System**:

- The **system** itself (mainly to talk about its components)
- The **running** system (mainly to talk about its behavior)
- The system in **development** (mainly to talk about phases and artifacts)



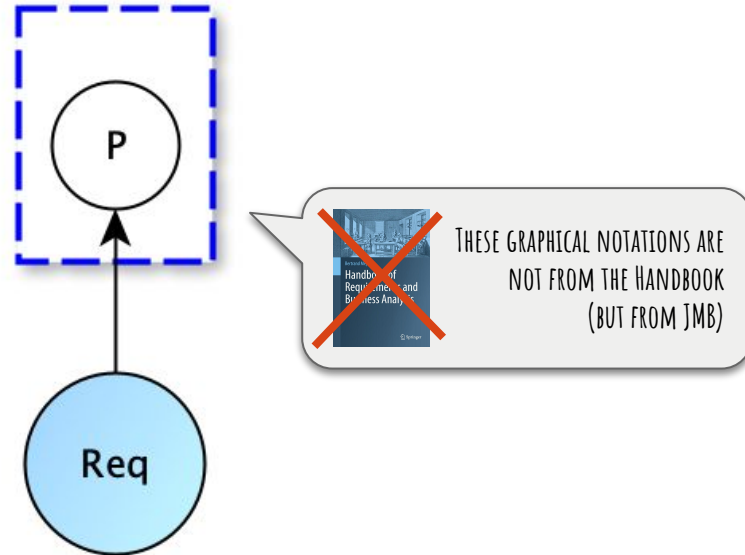
Outline

1. Dimensions of RE
- 2. Defining requirements**
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements



General definition of a Requirement

“A requirement is a (relevant) **statement** about a **property**”



Examples

“The project shall produce a first release by 31 October 2023.”

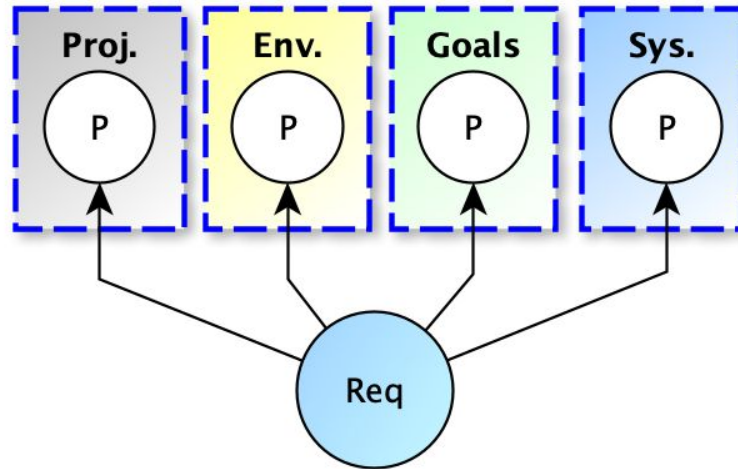
“All Web sites shall conform to GDPR (EU privacy rules).”

“The Bridge Maintenance System shall limit bridge closures to no more than one night a month.”

“After 5 failed login attempts, access shall be blocked for 30 minutes.”

General definition of a Requirement

“A requirement is a (relevant) **statement** about a **project**, **environment**, **goals** or **system property**”



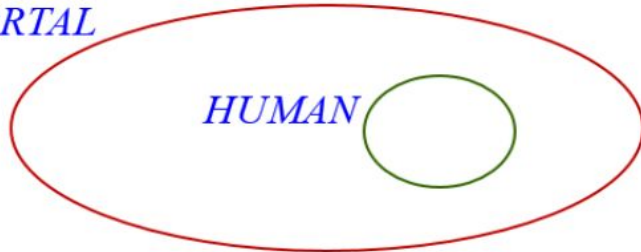
Some basic concepts

Property: boolean predicate (on a project, system or environment)

Statement: human-readable expression of a property

$\forall x: HUMAN \mid x.is_mortal$

MORTAL



Relevance

Goals: always (by definition)

Environment: if it can affect or be affected

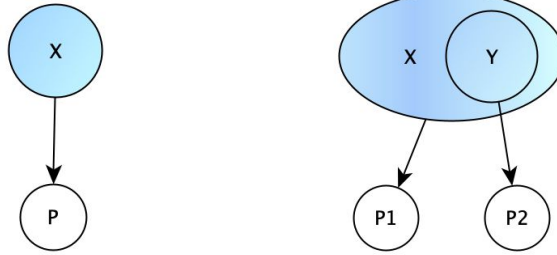
System: if it can affect or be affected by a stakeholder

Project: if it can affect or be affected by a stakeholder

A **statement** of a property is **relevant** if the **property is relevant**

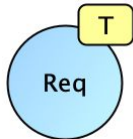
Elements of graphical representation

A requirement can be **Atomic** or **Composite**



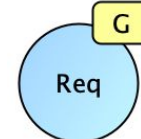
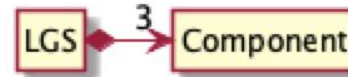
~~Handbook of Requirements and Business Analysis~~
 THESE GRAPHICAL NOTATIONS ARE NOT FROM THE HANDBOOK (BUT FROM JMB)

The **notation** of a requirement is the concrete syntax in which it is expressed (Text, Tabular, Graphical, formal)



“The LGS has three components.”

$\forall x: HUMAN \mid x.is_mortal$

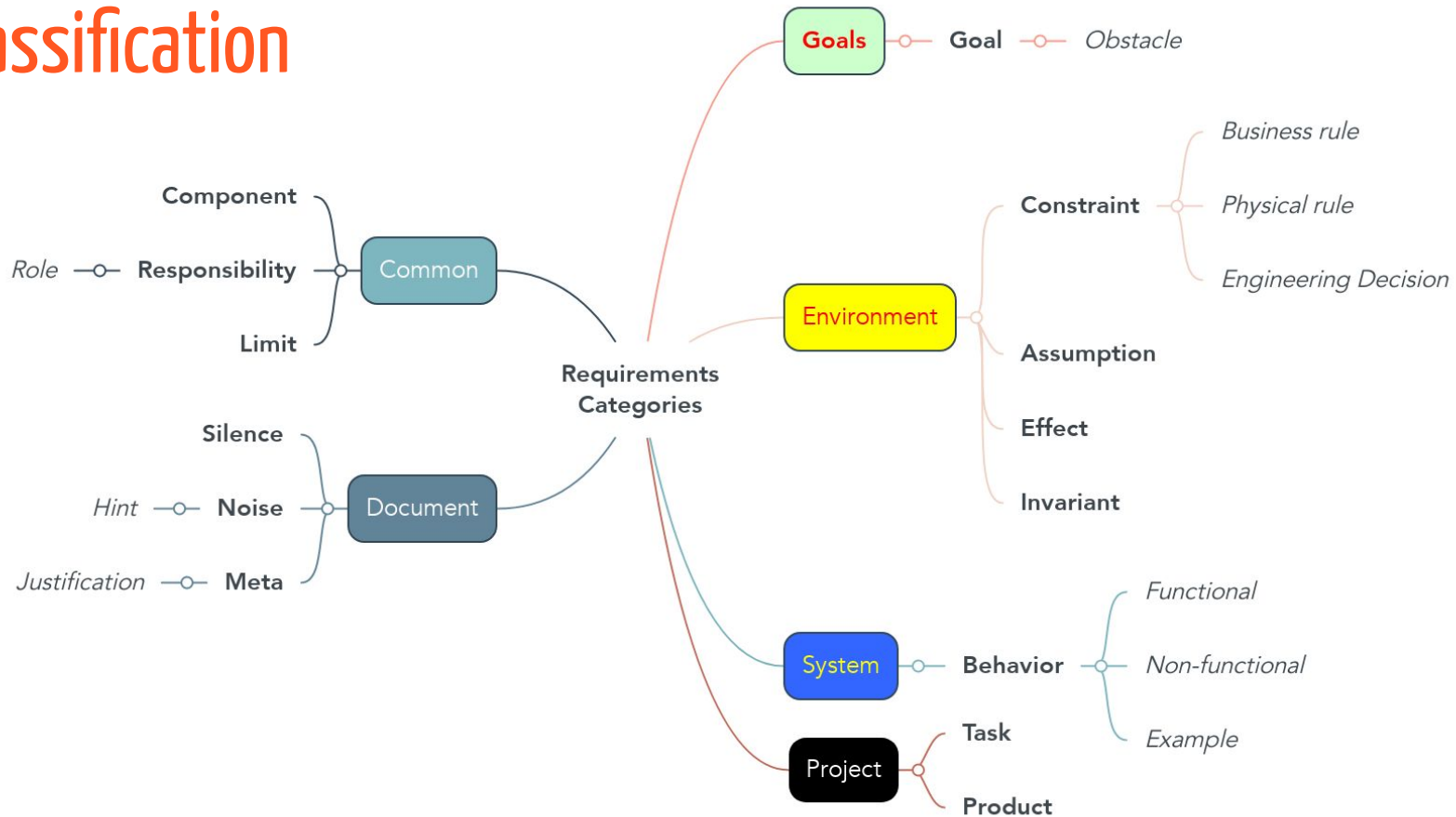


Outline

1. Dimensions of RE
2. Defining requirements
- 3. Kinds of requirements**
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements

Kinds of requirements (overview)

Classification



Kind of requirements (common to all PEGS)

- Component
- Responsibility
 - *Role*
- Limit

Kind of requirements (Goals)

- Goal
 - *Obstacle*

Kind of requirements (Projects)

- Task
- Product

Kind of requirements (System)

- Behavior
 - *Functional*
 - *Non-functional*
 - *Example*

Kind of requirements (Environment)

- Constraint
 - *Business rule*
 - *Physical rule*
 - *Engineering decision*
- Assumption
- Effect
- Invariant

Kind of requirements (Document description)

- Silence
- Noise
 - *Hint*
- Meta-requirement
 - *Justification*

Outline

1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
- 4. Requirements affecting Goals**
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements

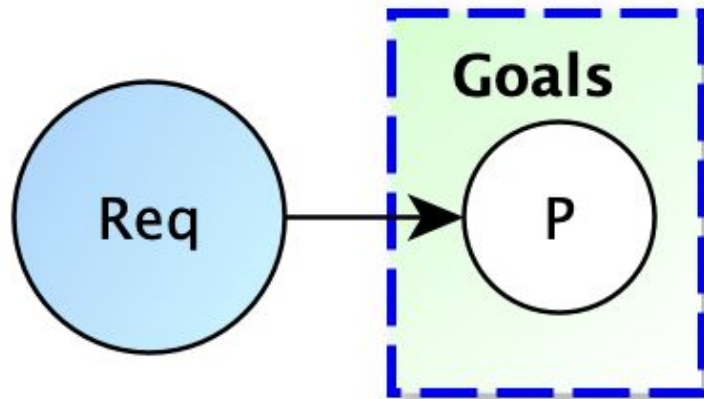
Goals Requirements

- Goal
 - *Obstacle*



Goal

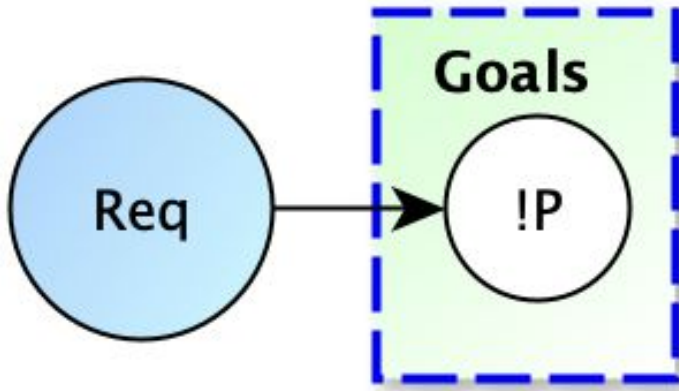
(Desired result for the target organization)



“The goal of the system is to make sure as many users as possible book a flight through our app instead of the concurrents’ ones.”

Obstacle (kind of goal)

(Goal describing a property to be overcome)



“The current manual operation makes it impossible to meet the expected growth of traffic over the next 10 years.”

Outline

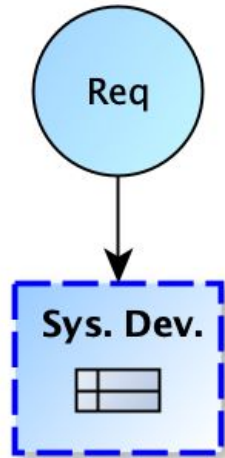
1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
- 5. Requirements on the Project**
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements

Projects requirements

- Task
- Product

Task

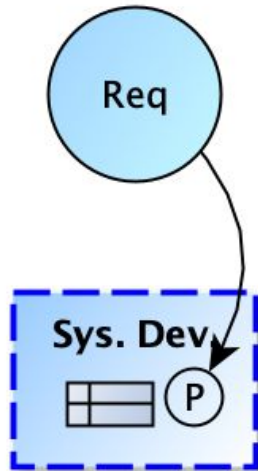
(The property that the project includes a certain activity)



“The team should meet in a daily basis, called daily meeting.”

Product

(Artifact produced or needed by a task)



“The following test plan is provided...”

Outline

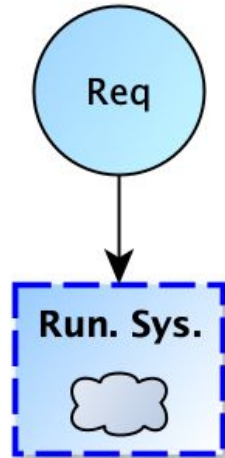
1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
- 6. Requirements on the System**
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements

System requirements

- Behavior
 - *Functional*
 - *Non-functional*
 - *Example*

Behavior

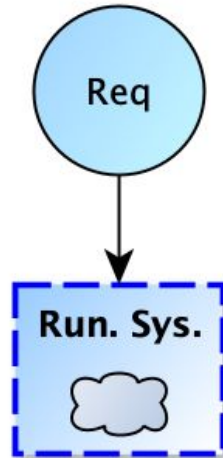
(A property of the effects of the operation of the system or some of its components)



“The system should allow to open and close the door safely.”

Functional requirement (kind of behavior)

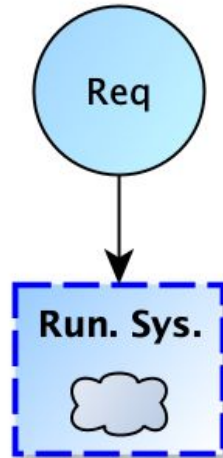
(What the system must do)



“The system should allow to open and close the door safely.”

Non-functional requirement (kind of behavior)

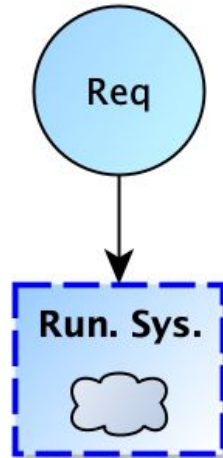
(How the system must perform)



“The identification process should be secure.”

Example (kind of behavior)

(Illustrative/representative scenario)



“Here is the description of the use case *cancel a previous order...*”

Outline

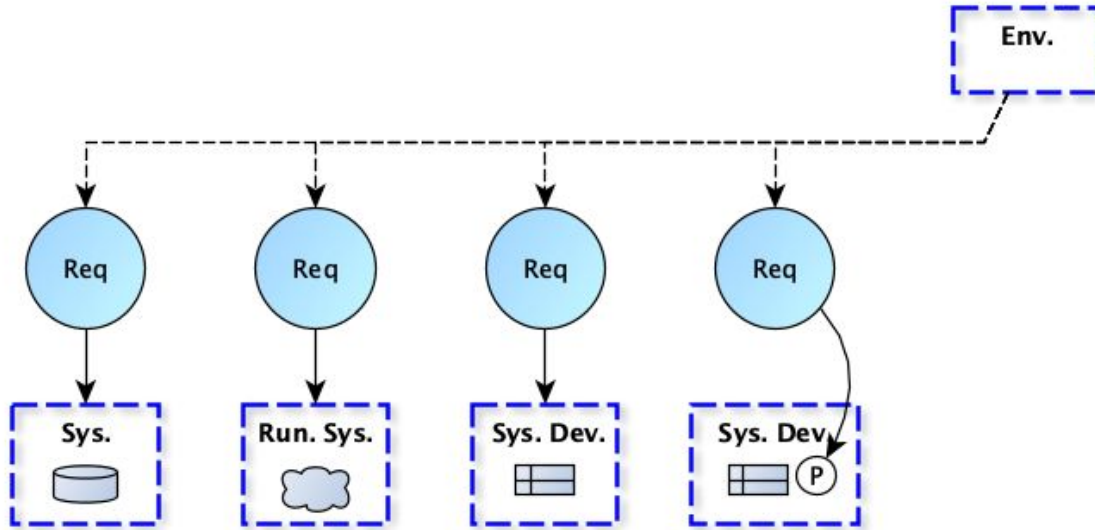
1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
- 7. Requirements on the Environment**
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements

Environment requirements

- Constraint
 - *Business rule*
 - *Physical rule*
 - *Engineering decision*
- Assumption
- Effect
- Invariant

Constraint

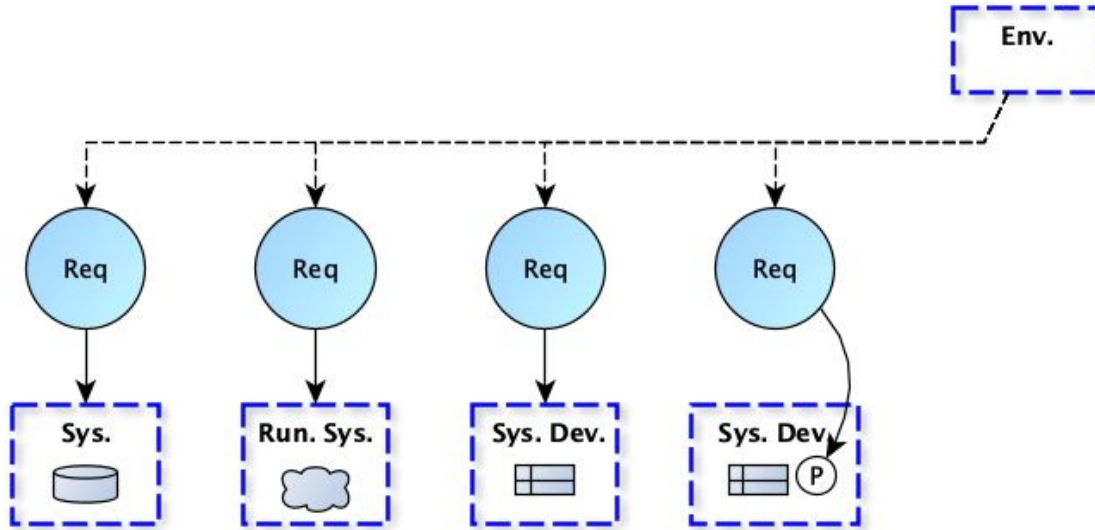
(A property imposed by the environment)



“Every transfer over 10.000€ requires an authorization.”

Business rules (kind of Constraint)

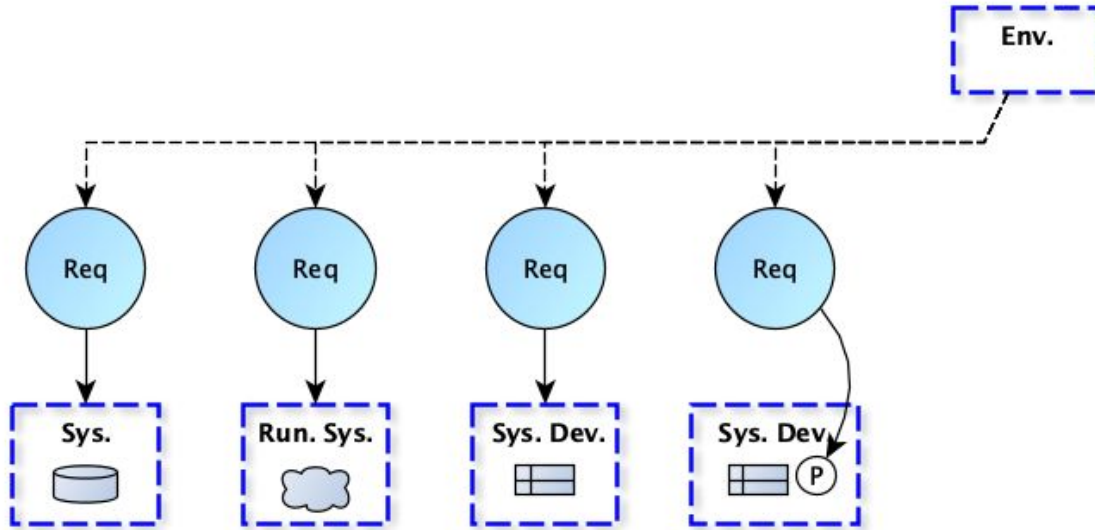
(A constraint imposed by an organization or standard)



“According to the regulation rule X.45F53, the amount of the engine CO2 emission must be less than...”

Physical rules (kind of Constraint)

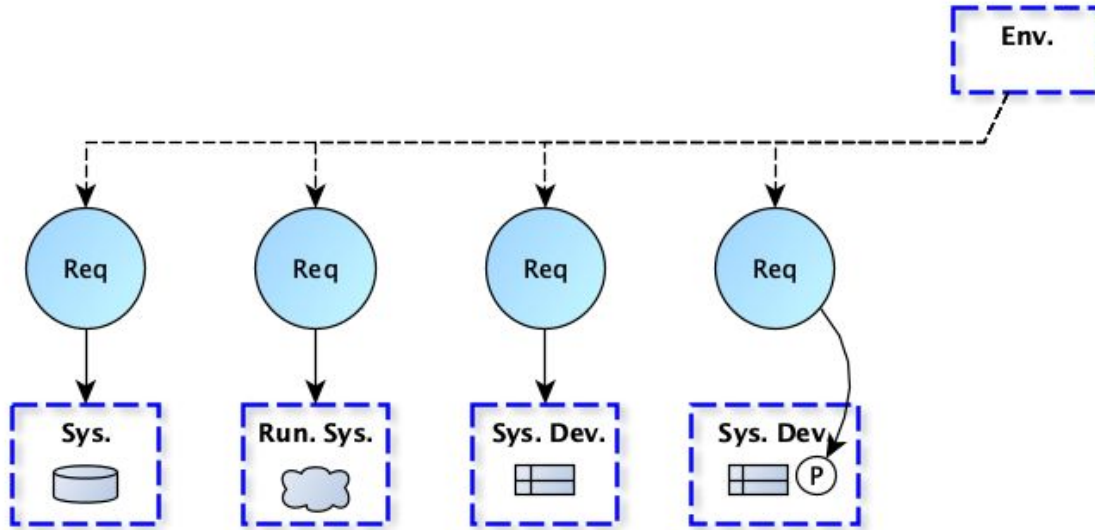
(A constraint imposed by nature)



“The volume of the tank needs to be twice the amount of ...”

Engineering decisions (kind of Constraint)

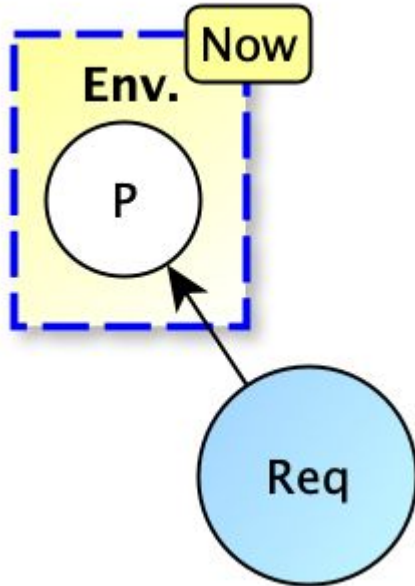
(A constraint imposed by **design**)



“According to our maximum reuse policy, our proprietary secured login system XYZ should be reused in this project.”

Assumption

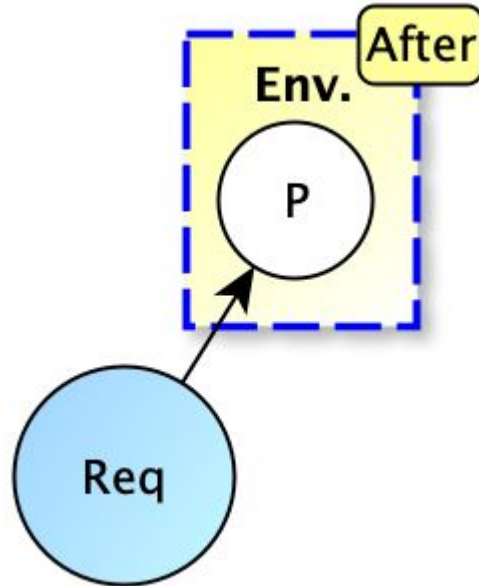
(Expected property of the environment)



“The available bandwidth will be 1 Mbit/s or more.”

Effect

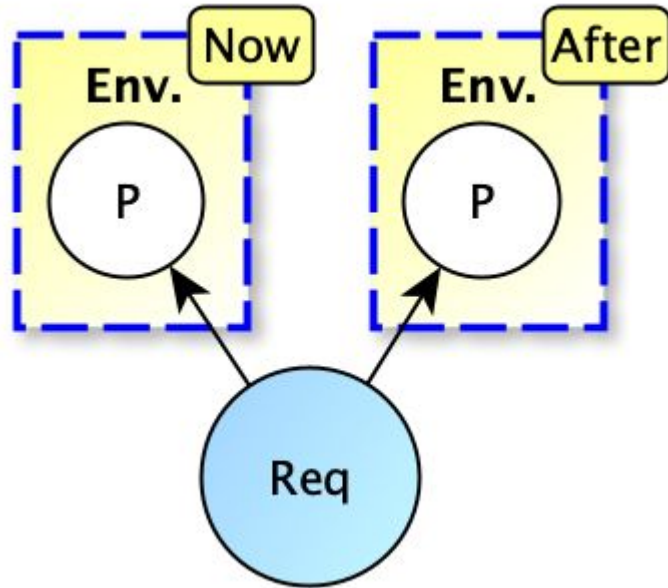
(Property of the environment affected by the system)



“When the system is put into operation, employees will be paid on the last working day of the month.”

Invariant

(Environment property that must be maintained)



“The system expects a temperature between 18 to 25 degrees Celsius (precondition) and maintains it in that range.”

Outline

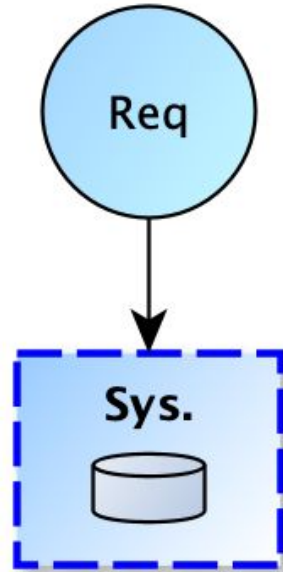
1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
- 8. Requirements applying to all dimensions**
9. Special requirements elements
10. The people behind requirements
11. Why perform requirements

Common to all PEGS

- Component
- Responsibility
 - *Role*
- Limit

Component

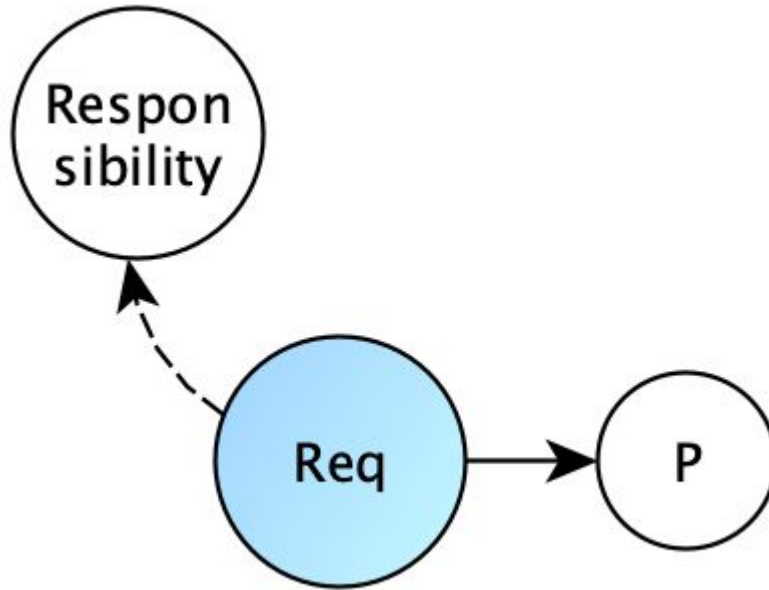
(Identification of a part of a whole)



“The Landing Gear System is composed of three parts.”

Responsibility

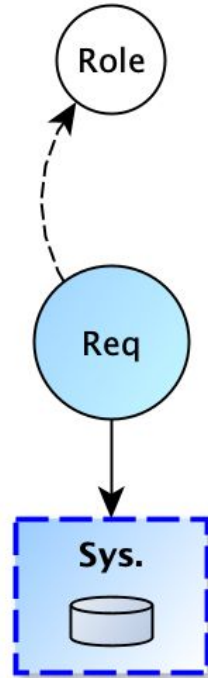
(Assignment of behavior or task to component)



“The control system is in charge of the opening/closing of the door.”

Role (kind of responsibility)

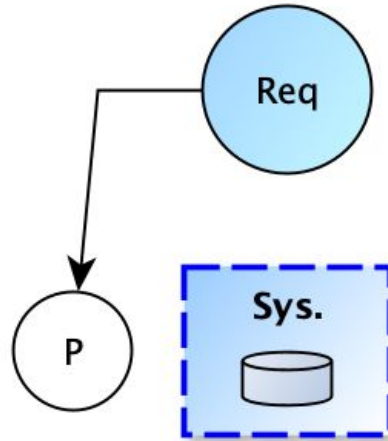
(A human or organizational responsibility)



“Authorizations are provided by the head of the audit department.”

Limit

(the property that the project, system or environment does *not* include a requirement of any of the preceding kinds)



“Integration testing will be performed in a follow-up project.”

Outline

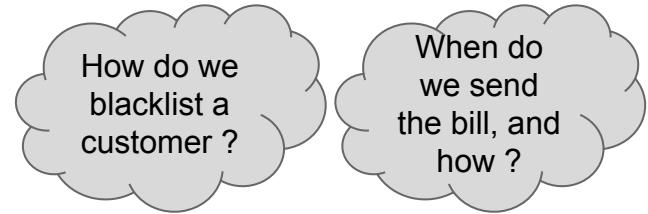
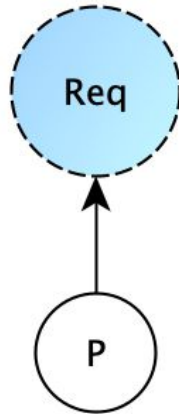
1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
- 9. Special requirements elements**
10. The people behind requirements
11. Why perform requirements

“Document-related” requirements

- Silence
- Noise
 - *Hint*
- Meta-requirement
 - *Justification*

Silence

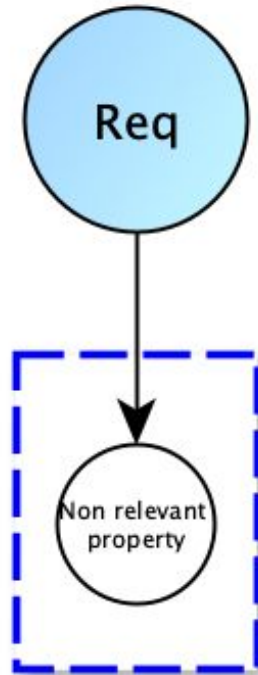
(a property that should have a requirement, but does not)



“The system should send the bill to the non blacklisted customers.”

Noise

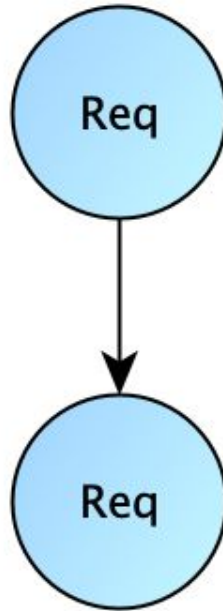
(something that should not be in the requirement document but is there)



“The director is not consistent in his decision making.”

Meta-requirement

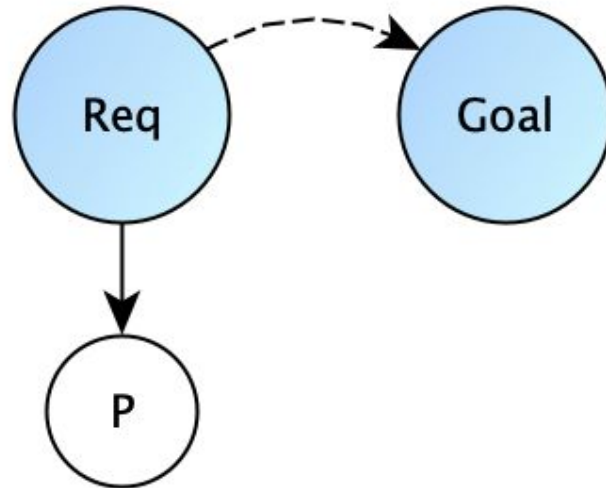
(a property of requirements themselves)



“The details are provided in Fig. 7.”

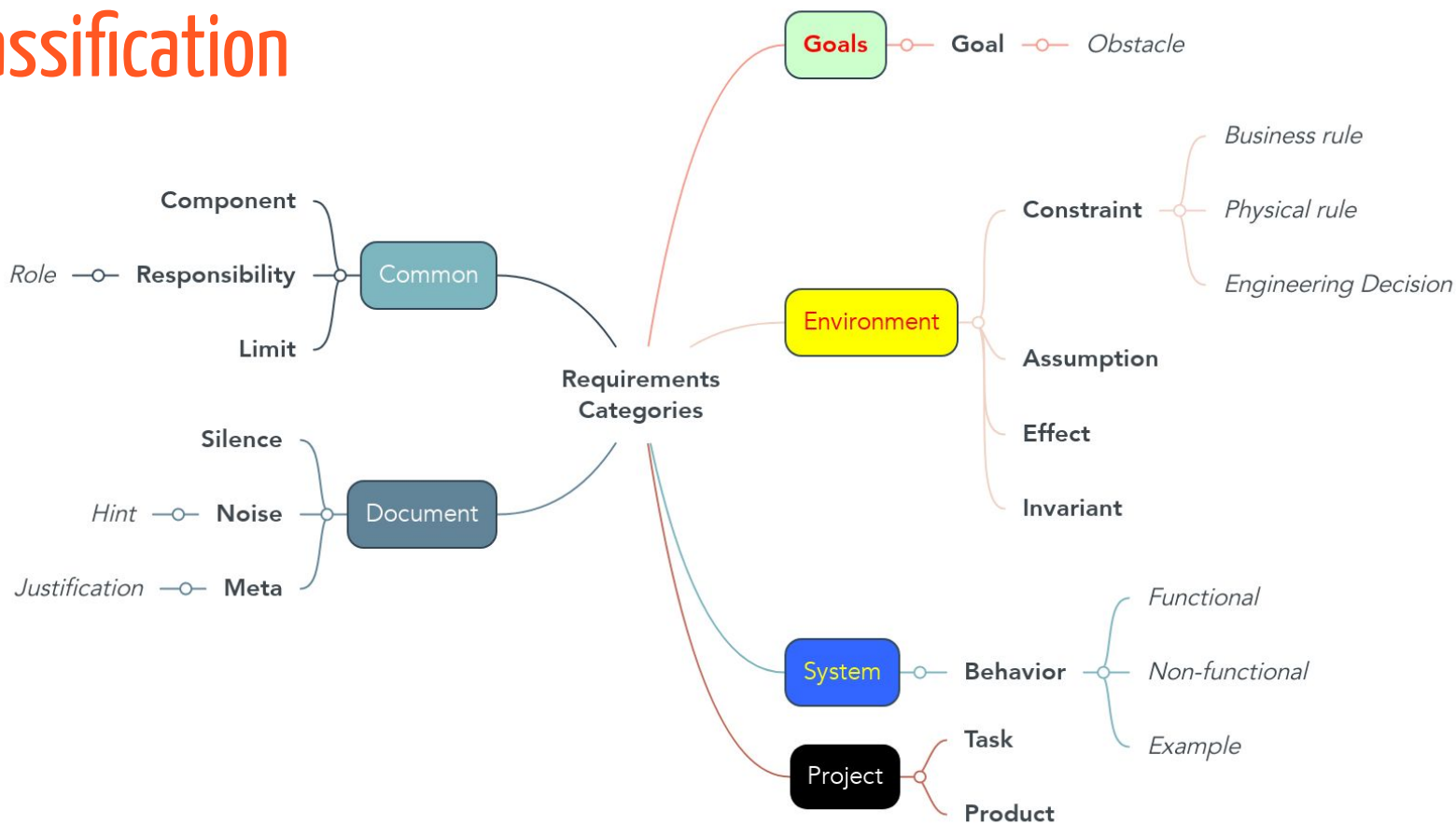
Justification (kind of Meta-requirement)

(Explanation of a project or system property, in reference to a goal or environment property)



“The presence of two signature fields follows from the rule on purchases higher than € 5000 (section E.3.X).”

Classification



Categories of requirements (derived)

- **Justification** (from Meta)
- **Role** (from Responsibility)
- **Business rule** (from Constraint)
- **Physical rule** (from Constraint)
- **Engineering decision** (from Constraint)
- **Hint** (from Noise)
- **Obstacle** (from Goal)
- **Functional** (from Behavior)
- **Non-Functional** (from Behavior)
- **Example** (from Behavior)

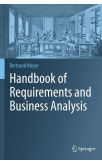
Guideline for category identification

1. Which PEGS (this shortens the possibilities)
2. Check if specific (not component/resp/limit or document)
3. Pick the best among the remaining ones

Outline

1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
- 10. The people behind requirements**
11. Why perform requirements

Categories of stakeholders



Chapter #1

Target groups (customer side)

- ❑ Primary Users of the future system (end users who directly interact with the product or system)
- ❑ Secondary Users of the future system (users who may not directly interact with the product but are affected by its use or outcomes)
- ❑ Potential Users (test groups when no real users are available)
- ❑ Users of the existing system (that will be replaced)
- ❑ Client Programmers (when the system will provide an API)
- ❑ Subject-Matter Experts (experts of the domain area covered by the system)
- ❑ Marketing and Sales (teams responsible for promoting and selling the product)
- ❑ Legal and Compliance Teams (ensuring the project complies with legal requirements and contracts)
- ❑ Regulators/Compliance Authorities (organizations or government bodies responsible for industry regulations)
- ❑ Customers (individuals or organizations that purchase or finance the product or service)
- ❑ Customers' customers (when the target system is for a client's customers usage)
- ❑ Labor unions (if the system will affect people's work conditions)
- ❑ Product Managers (those responsible for defining the product vision, strategy, and roadmap)
- ❑ Decision-makers (on target side)
- ❑ Executives/Leadership (senior management and executives with strategic interests)
- ❑ Sponsors (entities or individuals providing funding or resources for the project)
- ❑ Business Analysts (professionals who analyze business needs and translate them into requirements)
- ❑ Operations/IT Support (teams responsible for maintaining and supporting the product in the long term)
- ❑ Competitors (rival companies or products in the market)

Target groups (customer side) (ctd.)

- ❑ Suppliers/Vendors (those providing essential components or services for the project)
- ❑ Community and Public Interest Groups (organizations or individuals representing community interests)
- ❑ Financial Analysts/Investors (individuals or entities with a financial stake in the project's success)
- ❑ Environmental/Sustainability Groups (organizations concerned with ecological and sustainability aspects)
- ❑ Ethical and Social Responsibility Advocates (stakeholders who focus on ethical and social aspects of the project)
- ❑ Internal Users/Employees (persons who may use the system for internal processes)
- ❑ Support and Helpdesk (teams responsible for addressing user inquiries and issues)
- ❑ Security Experts (professionals responsible for ensuring the security of the product)
- ❑ Human Resources (who may have requirements related to employee roles and responsibilities)
- ❑ Government/Regulatory Bodies (agencies overseeing industry-specific regulations)
- ❑ Non-Governmental Organizations (external organizations with a vested interest in the project's impact)
- ❑ Academic Institutions (researchers or educators with an interest in the project's outcomes)
- ❑ Media and Press (entities that may cover or report on the project)
- ❑ Alliances and Partners (organizations collaborating on the project)
- ❑ ...
- ❑
- ❑
- ❑

Production groups (development side)

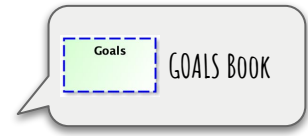
- ❑ Project Managers (the persons directing the development)
- ❑ Testers/Quality Assurance (individuals responsible for testing the product to ensure it meets requirements)
- ❑ Testers (to ascertain conformance of any future implementation to the requirements)
- ❑ Documenters (to ease the production of user manuals for example)
- ❑ Trainers (if the future system requires training)
- ❑ Designers/UX Experts (those responsible for the user interface and overall user experience)
- ❑ HCI Experts (or ergonomics experts, for Human-Computer Interfaces)
- ❑ Developers/Engineering Team (the technical team responsible for building and implementing the solution, to avoid "pie-in-the-sky" systems requirements)
- ❑ Open-source Community (for open-source projects)
- ❑ GDPR Experts (to assess conformance with the regulation)
- ❑ Legal Experts (for any Intellectual property or licensing issues)
- ❑ Ethical Experts or Corporate Social Responsibility (for concerns such as gender balance)
- ❑ Health and Safety Experts (for hazard control)

Outline

1. Dimensions of RE
2. Defining requirements
3. Kinds of requirements
4. Requirements affecting Goals
5. Requirements on the Project
6. Requirements on the System
7. Requirements on the Environment
8. Requirements applying to all dimensions
9. Special requirements elements
10. The people behind requirements
- 11. Why perform requirements**

12 good reasons (goals) to perform requirements

- Understand the organization's goals
- Understand the problem to be addressed by the system
- Help plan the project
- Prompt relevant questions about the problem, environment, goals and system
- Provide basis for answering questions about the problem, environment, goals and system
- Decide what the system should do
- Decide what the system should not do
- Ascertain that the system will satisfy the needs of its stakeholders
- Understand fundamental environment properties
- Provide a basis for the development
- Provide a basis for verification
- Provide a basis for evolution

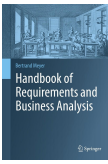


Understand the organization's goals

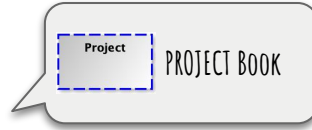
All actors in the project should keep in mind what the business objectives are, underlying all project decisions.

Understand the problem to be addressed by the system

What actual **issues** is the project meant to address?



Help plan the project



One of the outcomes of requirements is to help **organize** the project.

Prompt relevant questions

about the problem, environment, goals and system

One benefit of studying requirements is to make sure that **important questions** about the future system (and the goals, environment and project) **do not remain overlooked**.

Provide basis for answering questions

about the problem, environment, goals and system

Asking questions (the previous point) is good; answering them is better!

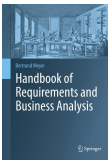


Chapter #1

Decide what the system should do



This task corresponds to the **traditional view** of requirements.



Chapter #1

Decide what the system should not do

Just as important is the **delimitation** of the system's functions and scope.

Good requirements make sure to define where the **scope** of the future system stops, and what problems it is not expected to address.

Ascertain that the system will satisfy the needs of its stakeholders

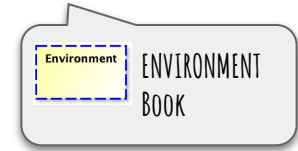
Every system is ultimately built **for people**.

With a process leading to explicit requirements, stakeholders' representatives will be able to examine them and **detect any mismatch with actual needs**.

Understand fundamental environment properties

What are the **constraints** on the system?

How will the system **affect** the business or physical environment?



Provide a basis for the development

The requirements will provide the **key reference** for the design and implementation tasks.

Provide a basis for verification

Being expressed in terms of stakeholder needs and environment properties, rather than implementation, requirements are the ideal reference for assessing the **fitness of the system** (once it exists) to its objectives.

This observation applies both to a traditional “Waterfall” model (performing verification at the end) and to a more modern continuous verification process.

Provide a basis for verification (V vs. V)

Does the right **thing**

- Validation
- « Building the right system »



<https://www.canon.co.nz/software-solutions/iw-sam>

Does them right

- Verification
- « Building the system right »



<https://www.techopedia.com>

Provide a basis for evolution

No released system remains **unchanged forever**.

While inevitable, **evolution** should proceed in a smooth and orderly way.

Requirements provide the right basis for **controlled evolution**.

Discussions time



Get the slides

 <https://requirements.university>

