Requirements Handbook Companion

Table of Contents

1. Disclaimer

⚠️ | Work in progress!

Here is a list of warnings:

- This work is in progress and subject to constant improvements, so pay attention to dates and versions.

- If you read the PDF version of this material on a browser, the links might not be clickable. Download it instead in your machine and open it with a PDF viewer. The links should be clickable there.

Conventions for this book:

Table 1. Icons signification

Icon	Signification
⊘	A precisely referenced requirement this section is satisfying
🐞	A precisely referenced requirement this section is related to

Icon	Signification
⊗	No corresponding requirement for this section (should be fixed)

2. Context

This repository constitutes the companion of the book: Handbook of requirements and business analysis. It serves as the basis for the future Handbook's site: http://requirements-handbook.org/

3. Overview of PEGS

⚠️ Work in progress!

This chapter is an overview of PEGS. It aims at making this companion book self-content. We highly recommend, for more details and a full description of the subtleties of PEGS, to read the corresponding http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf. Besides, this chapter does not cover the following aspects of PEGS that the http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf fully addresses:

- General principles of requirements (see http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf, chapter 2)

- Quality criteria for requirements (see http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf, chapter 4)

- How to write requirements (see http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf, chapter 6)

- Completeness (see http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf, chapter 11)

- Verification (see http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf, chapter 12)

3.1. Basic principles

3.1.1. Universe of discourse

Figure 1. The PEGS logo

PEGS takes its name as a reminder of the four dimensions of requirements, leading to the organization of requirements in four books (Section 3.2 will detail this books' structure):

Goals

the needs of the target organization, which the **system** will address.

System

a set of related artifacts, devised to help meet certain **goals**.

Project

the set of human processes involved in the planning, construction, revision and operation of the **system**.

Environment

the set of entities external to the **project** and **system** but with the potential to affect the **goals**, **project** or **system** or be affected by them.

> All this document will follow, as much as possible, the color convention illustrated by the logo (see Figure 1).

3.1.2. Kind of requirements

A requirement is a relevant statement about a project, environment, goal, or system. The http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf defines important notions such as property, statements, relevance, or stakeholders.

Icon	Kind	Quick description
🗔	Component	
👤	Responsibility	
	Limit	
	Role	

To describe the kinds of requirements, let us start with the one that may arise in all four PEGS dimensions: component, responsibility, limit and role.
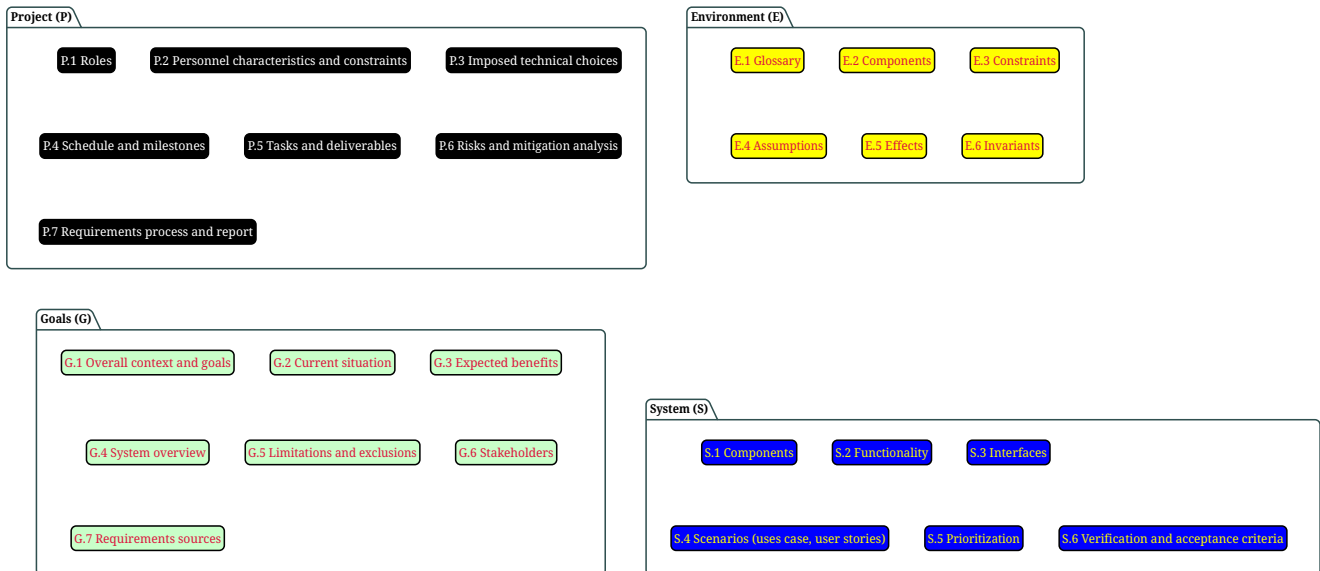
3.2. Overall structure

Figure 2. The four books of requirements

This structure can influence the organization of a document in books. It can also be lead to requirements packages (e.g., in SysML) or folders (e.g., in a spreadsheet).
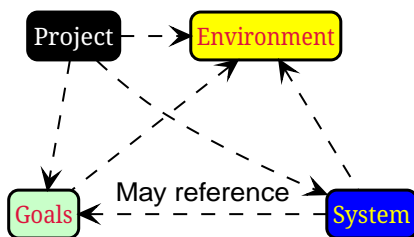
3.3. Links between the four PEGS
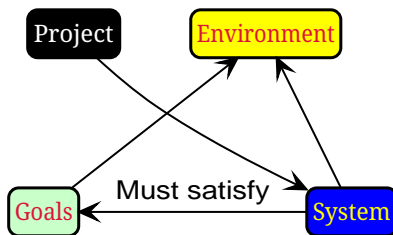

Figure 3. Reference links

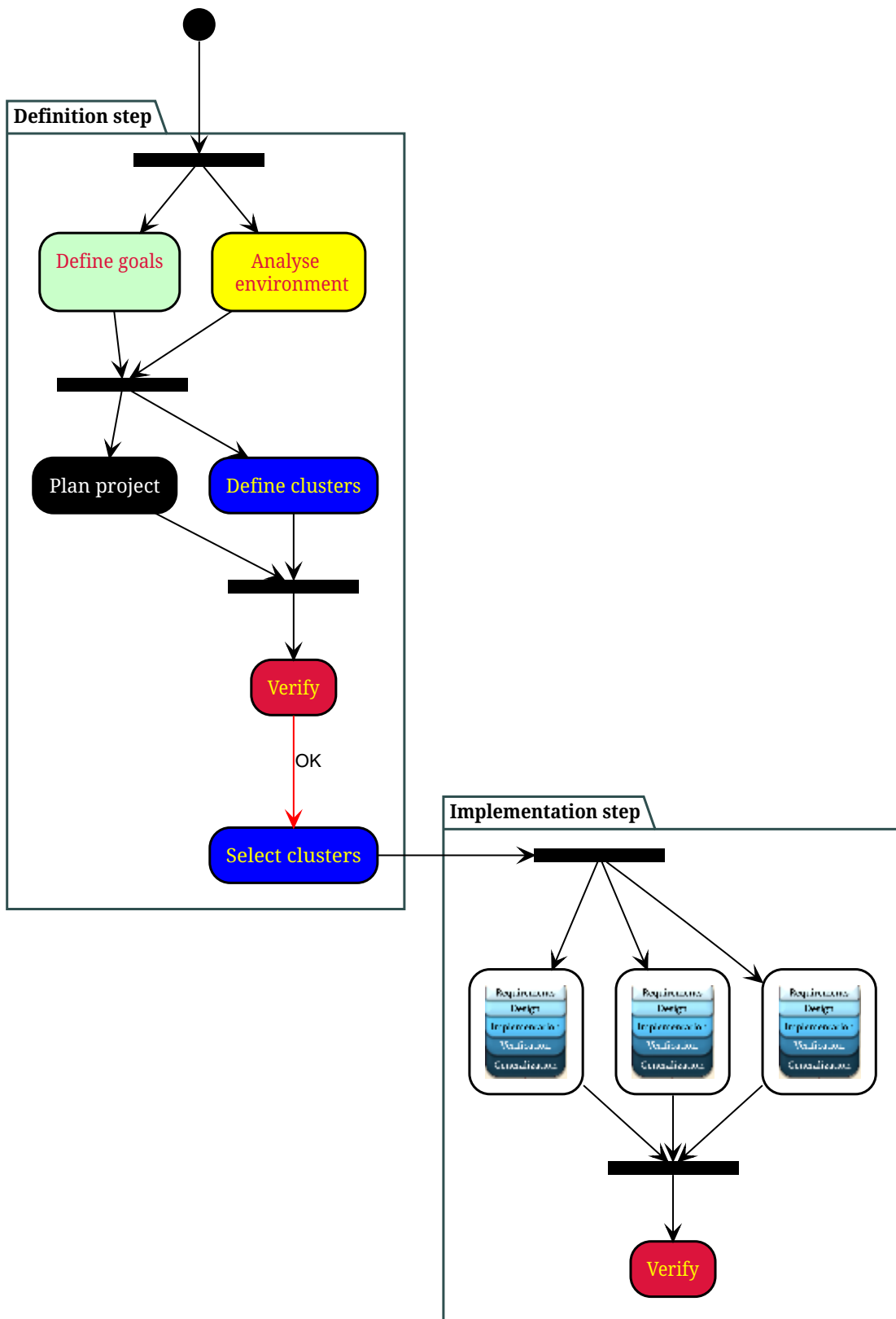*Figure 4. Verification obligations*

3.4. The lifecycle model

Figure 5. The lifecycle model

4. Case studies

We illustrate the use of PEGS through the following case studies:

⚠ Work in progress

1. The Roborace (see the dedicated repo here)

2. A more information-system-oriented example (one option is this library example)

3. This Companion Book' requirements (see the dedicated repo here)

5. Book templates

We provide a set of Book templates in this companion web site to help you apply PEGS and organize your requirements. Here is the list of the available templates (feel free to contribute by submitting additional templates):

⚠️ | Work in progress

1. DOCX

2. Google Doc (see this example for now)

3. LaTeX (see IEEE example)

4. GitHub

5. SysML

⊘ *Corresponding Requirement*

> This section satisfies this requirement.

6. Verification rules

This section provides some implementation examples of the verification rules described in the http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf.

6.1. Books mutual references

One of the requirements for this book, taken from the http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf, says (see here):

> The books may reference each other, but not arbitrarily.

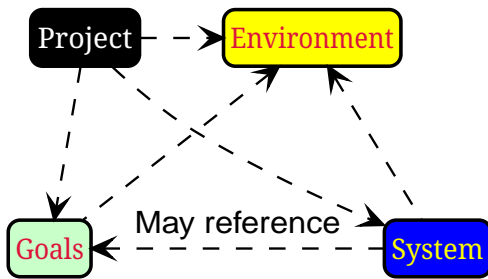The Figure 6 shows which books may refer to which.

Figure 6. Possible references between the books

From this mutual references rule, a concrete implementation, written in Gherkin, enforces its application:

```gherkin
#---------------------------------
# language: en
Feature: Book mutual references
    The books should follow the mutual references rules.

Scenario: The Environment book must not refer to the Goals and Project books
    Given The Environment book
    Then No reference should include the Goals book
    And No reference should include the Project book
    And Only E.5 section can refer to the System book

Scenario: The Goals book must not refer to the Project and System books
    Given The Goals book
    Then No reference should include the Project book
    And No reference should include the System book

Scenario: The System book must not refer to the Project book
    Given The System book
    Then No reference should include the Project book
```

Figure 7. Verification rule as a BDD feature (see this file)

Here is the execution, using `cucmumber`, of the corresponding tests, showing that they all pass (green lines):

```
# language: en
#------------------------------
# Checking Books rules (see {Handbook, v.2021-02-19, p.49})
# JMB - 2021
#------------------------------
Feature: Book mutual references
    The books should follow the mutual references rules.

  Scenario: The Environment book must not refer to the Goals and Project books # feat
    Given The Environment book                                                # feat
    Then No reference should include the Goals book                           # feat
    And No reference should include the Project book                          # feat
    And Only E.5 section can refer to the System book                         # feat

  Scenario: The Goals book must not refer to the Project and System books # features/
    Given The Goals book                                                  # features/
    Then No reference should include the Project book                     # features/
    And No reference should include the System book                       # features/

  Scenario: The System book must not refer to the Project book # features/book.featur
    Given The System book                                      # features/step_defini
    Then No reference should include the Project book          # features/step_defini
```

Figure 8. tests execution

References

- Sommerville & Sawyer

- Axel van Lamsweerde

- Klaus Pohl

Appendices

Appendix A: Useful links

- The draft of the Method Book: PDF

- The draft of the Companion Book: Google Doc

Appendix B: Specific implementations

Here is a list of potential mappings between the PEGS:

SysML

- Each PEGS could be a package

- Requirements could be stereotyped (e.g., [Goals] or [System])

FORM-L

- M. Thuy could be interested in providing a template for the PEGS in Form-L

RSML

- Florian could integrate the PEGS in the editor

Appendix C: Checks results and issues

URLs

Checking URLs result (last update: 2021-03-05):

```
Ven 5 mar 2021 15:00:37 CET

FILE: README.adoc
[] http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf
[/] http://requirements-handbook.org/
[]
https://docs.google.com/document/d/1HrWCRzyW_iTf1QXFFzEoDvvc66IzMCDb3uXGS5GRWz8/edit?u
sp=sharing
[] https://github.com/FormalRequirements/requirements-handbook-companion
[] https://formalrequirements.github.io/requirements-handbook-companion
[] https://formalrequirements.github.io/companionRequirements
[] https://github.com/formalrequirements/companionRequirements
[] https://gist.github.com/rxaviers/7360908
[] https://github.com/FormalRequirements/requirements-handbook-
companion/workflows/Check%20URLs/badge.svg
[] https://github.com/FormalRequirements/requirements-handbook-companion/actions
[] https://img.shields.io/badge/License-MIT-yellow.svg
[] https://opensource.org/licenses/MIT
[] https://img.shields.io/badge/Gitpod-ready--to--code-blue?logo=gitpod
[] https://gitpod.io/#https://github.com/FormalRequirements/requirements-handbook-
companion
[] https://img.shields.io/badge/PDF-Download-blue
[] https://github.com/FormalRequirements/requirements-handbook-
companion/blob/main/README.pdf
[] https://github.com/FormalRequirements/roboraceRequirements
[] https://github.com/ddd-by-examples/library?ref=hackernoon.com#domain-description
[] https://github.com/FormalRequirements/companionRequirements
[] https://github.com/jpeisenbarth/SRS-Tex
[] https://raw.githubusercontent.com/FormalRequirements/requirements-handbook-
companion/9f100f121c15772b07cc2bdc3d25afee587784fe/images/links.svg
[] https://github.com/FormalRequirements/requirements-handbook-
companion/blob/main/check-results.txt

22 links checked.

FILE: githubImpl.adoc
[] http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf
[/] http://requirements-handbook.org/
[]
https://docs.google.com/document/d/1HrWCRzyW_iTf1QXFFzEoDvvc66IzMCDb3uXGS5GRWz8/edit?u
sp=sharing
```

```
[] https://github.com

4 links checked.

FILE: overview.adoc
[] http://se.ethz.ch/~meyer/down/requirements_handbook/REQUIREMENTS.pdf
[/] http://requirements-handbook.org/
[]
https://docs.google.com/document/d/1HrWCRzyW_iTf1QXFFzEoDvvc66IzMCDb3uXGS5GRWz8/edit?u
sp=sharing

3 links checked.
```