



Requirements Engineering: Specification & Validation

Software Requirements and Design

CITS4401

Lecture 8 part 1

Key ideas (this week)

1. Testing requirements

Convince me!

2. Prototyping requirements

Show me!

3. Requirements Specification Document

Tell me!

4. Managing requirements

Maintain me!



The Requirements Specification...

- In planned methodologies
- is the **official statement** of what is required of the system developers
- forms the basis of a system **contract** between client and developer
- is a **reference tool** used by clients, designers, programmers, testers and maintenance engineers.

The Problems of Requirements

- *What goal(s) are we trying to satisfy?*
- *How do we identify the scope and properties of the solution space ?*
- Software Engineers differ in their views about the best way to answer these questions.
- We will look at both planned and agile approaches and something in between.

Visit learning.acm.org



The four PEGS of requirements engineering

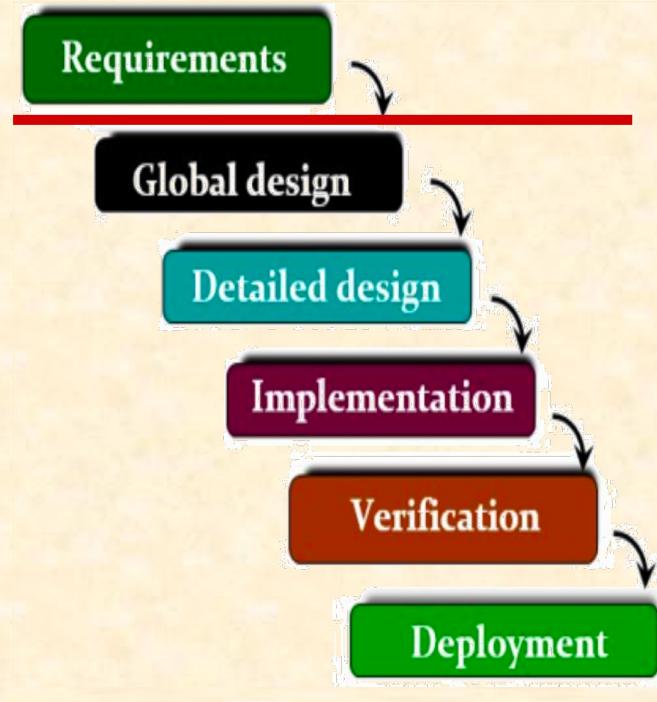
Bertrand Meyer

Schaffhausen Institute of Technology & Eiffel Software

ACM Tech Talk, 4 March 2021

Waterfall vs Agile

Chasm: traditional vs agile



Agile rejection of “big upfront anything”

*And those things called requirements?
They are really candidate solutions;
separating requirements from
implementation is just another form of
handover*

Mary Poppendieck,
in “Lean Software”





IEEE Standard

Still the standard...

IEEE Recommended Practice for Software Requirements Specifications

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Approved 25 June 1998

IEEE-SA Standards Board

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, acronyms & abbreviations
 - 1.4 References
 - 1.5 Overview
- 2. Overall description
 - 2.1 Product perspective
 - 2.2 Product functions
 - 2.3 User characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and dependencies
- 3. Specific requirements
- Appendices
- Index

Abstract: The content and qualities of a good software requirements specification (SRS) are described and several sample SRS outlines are presented. This recommended practice is aimed at specifying requirements of software to be developed but also can be applied to assist in the selection of in-house and commercial software products. Guidelines for compliance with IEEE/EIA 12207.1-1997 are also provided.

Keywords: contract, customer, prototyping, software requirements specification, supplier, system requirements specifications

13

Requirements Specification Document Template (IEEE)



1. Introduction

- 1.1 Purpose of the system
- 1.2 Scope of the system
- 1.3 Objectives and success criteria of the project
- 1.4 Definitions, acronyms, and abbreviations
- 1.5 References
- 1.6 Overview

2. Current system

3. Proposed system

- 3.1 Overview
- 3.2 Functional requirements

- 3.3 Non-functional requirements
 - 3.3.1 Usability
 - 3.3.2 Reliability
 - 3.3.3 Performance
 - 3.3.4 Supportability
 - 3.3.5 Implementation
 - 3.3.6 Interface
 - 3.3.7 Packaging
 - 3.3.8 Legal
- 3.4 System models
 - 3.4.1 Scenarios
 - 3.4.2 UML use case models
 - 3.4.3 Object model
 - 3.4.4 Dynamic models
 - 3.4.5 User interface—navigation paths and screen mock-ups

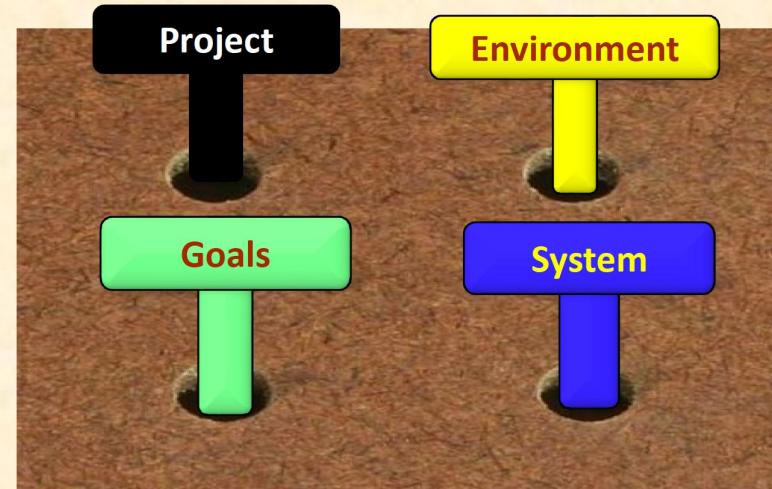
4. Glossary

PEGS (Meyer)

Defining requirements properly: the four PEGS

The aim is to execute:

- a **project**
- in a certain **environment**
- to achieve certain **goals**
- by developing a **system**



15

System vs Environment

System versus environment

Compare:

- “The gate shall close in at most 3 seconds”
- “Trains shall be assumed to travel at no more than 300 Km/Hr”



Pamela Zave



Michael Jackson



Project vs Goals

- Project Requirements concern how the SW development will be carried out
- Example: The system will be implemented using React and Express
- Example: The system will be delivered in 3 sprints of 1 month each with final deliverable on 31 May 2021
- Goals Requirements capture the business goals of the system – why we are building it
- Example: The system will enable admin staff to process 4 student applications per hour instead of 2 hours per application now

Requirements Sources

Sources of requirements

“Requirements document”

Meeting minutes

PowerPoint presentations

Emails

Regulatory documents

Documentation on previous projects

Code

Competing products

Anthony Finkelstein, 1994
“Pre-Requirement Specification



PEGS requirements repository

In a nutshell (2): Four books of requirements



Project Book

P

- P.1 Roles
- P.2 Personnel characteristics and constraints
- P.3 Imposed technical choices
- P.4 Schedule and milestones
- P.5 Tasks and deliverables
- P.6 Risks and mitigation analysis
- P.7 Requirements process and report

Goals Book

G

- G.1 Overall context & goals
- G.2 Current situation
- G.3 Expected benefits
- G.4 System overview
- G.5 Limitations and exclusions
- G.6 Stakeholders
- G.7 Requirements sources

Environment Book

E

- E.1 Glossary
- E.2 Components
- E.3 Constraints
- E.4 Assumptions
- E.5 Effects
- E.6 Invariants

System book

S

- S.1 Components
- S.2 Functionality
- S.3 Interfaces
- S.4 Scenarios (use cases, user stories)
- S.5 Prioritization
- S.6 Verification and acceptance criteria

PEGS books

Notes on the plan

- Does not assume a linear document
- Elements can be anywhere but should be recorded in the repository
- Tools can produce linear version
- Templates (Word etc.) will be available
- We are writing a companion book applying these ideas to a large practical example



31

References from Meyer's talk

References



On Formalism in Specifications (B. Meyer), in *IEEE Computer*, 1985

<http://se.ethz.ch/~meyer/publications/computer/formalism.html>

Multirequirements (B. Meyer), in *Modelling and Quality in Requirements Engineering*, eds. Seyff and Koziolek, MV Wissenschaft, 2013

<http://se.ethz.ch/~meyer/publications/methodology/multirequirements.pdf>

The Anatomy of Requirements (J-M Bruel, S. Ebersold, F. Galinier, B. Meyer, A. Naumchev), *TOOLS* 2019

<https://arxiv.org/abs/1906.06614>

The Formal Picnic Approach to Requirements (B. Meyer), blog article, 2020

<https://cacm.acm.org/blogs/blog-cacm/232677-the-formal-picnic-approach-to-requirements/fulltext>

The Role of Formalism in Software Requirements (J-M Bruel, S. Ebersold, F. Galinier, B. Meyer, A. Naumchev), accepted for publication in 2021 in *ACM Computing Surveys*
<http://se.ethz.ch/~meyer/publications/requirements/formalism.pdf>

Handbook of Requirements and Business Analysis (B. Meyer), to appear in 2021

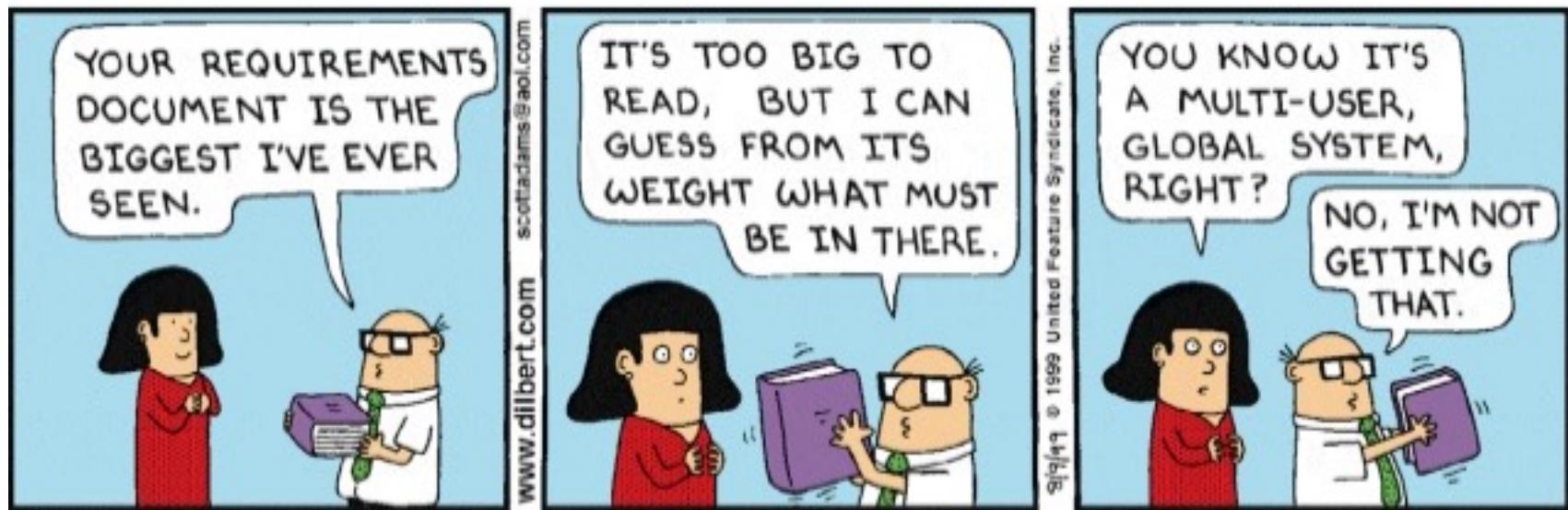
50

Attributes of a Good Requirements Specification

- Concise
- Complete
- Unambiguous
- Testable
- Consistent
- Feasible
- Modifiable
- Traceable
- Specifies external system behaviour only
- Specifies constraints on the implementation
- Easy to change
- Reference Tool for maintainers
- Records forethought about the life cycle
- Characterises acceptable responses to undesired events

More is not usually better

Aim to be concise!



Dilbert

© Scott Adams

At <http://dilbert.com/strip/1999-08-09/>

(Educational/Classroom usage permission is granted by Universal Uclick. All Rights Reserved)

Summary

- **Requirements Specification**
 - Document or Repository
 - Basis for client-developer contract
- Desirable attributes
 - Concise, complete, consistent, unambiguous, testable, modifiable, traceable, feasible
- Choose an appropriate method for your project
 - Agile: User stories
 - In between: PEGS
 - Waterfall: IEEE Standard