

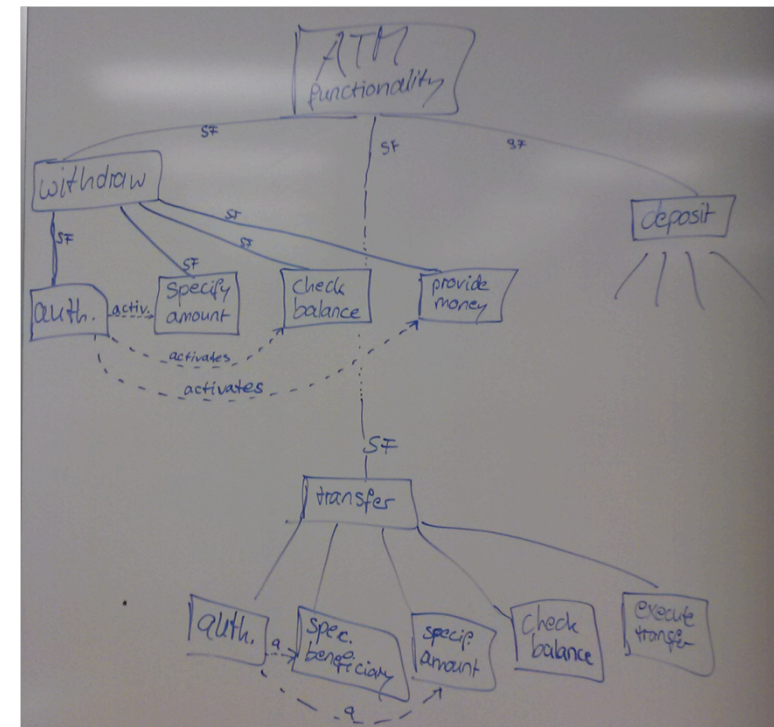
Requirements Engineering: Quality Assurance

CECS 590

Recap time!



- Functional hierarchies
 - What is a functional hierarchy?
 - Why are they helpful? What for?
 - How do we develop one?



Requirements Engineering – Outline

- WHY do we need Requirements Engineering and what is it?
- Principles: Definitions, process, roles, problem/solution view, artifact orientation
- System Models: Decomposition and abstraction, system views
- Frameworks: What reference structures can I use for requirements?
- Business Case Analysis: Why are we building this system?
- Stakeholders: Who are the people to talk to about requirements?
- Goals and Constraints: What are the major objectives for the system?
- System Vision: What exactly do we want to achieve?
- Domain Models: What are the surrounding systems ours interacts with?
- Usage Models: How will the system interact with the user?
- Software quality models: How to determine the quality characteristics?
- Quality requirements: How to specify which qualities need to be met?
- Process requirements: How to specify constraints for development?
- Towards a system specification: How to hand over to design?
- **Quality assurance: How to ensure that RE is done in a good way?**
- Change management: How to evolve requirements?

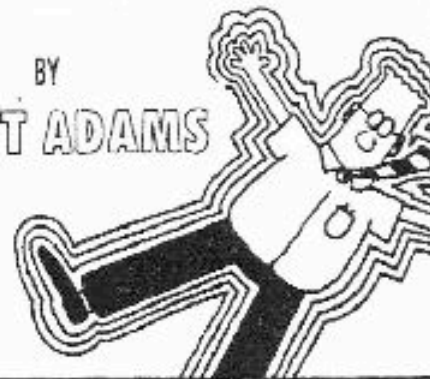
RATBERT, MY COMPANY
IS HIRING FOR OUR
QUALITY ASSURANCE
GROUP. YOU'D BE PERFECT.



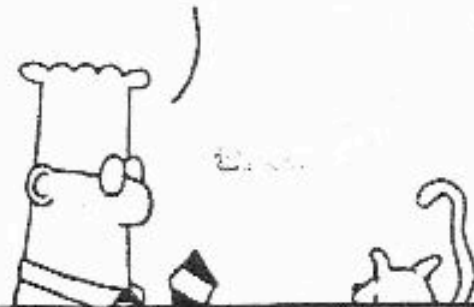
WHAT WOULD
I HAVE TO DO?

DILBERT[®]

BY
SCOTT ADAMS



YOU WOULD FIND FLAWS
IN OUR NEW PRODUCT,
THUS MAKING YOURSELF
AN OBJECT OF INTENSE
HATRED AND RIDICULE.



BUT THEN YOU'D FIX
THOSE FLAWS... AND
YOUR RESPECT FOR ME
WOULD GROW INTO
A SPECIAL BOND OF
FRIENDSHIP,
RIGHT?!



NO, THEN
WE SHIP.

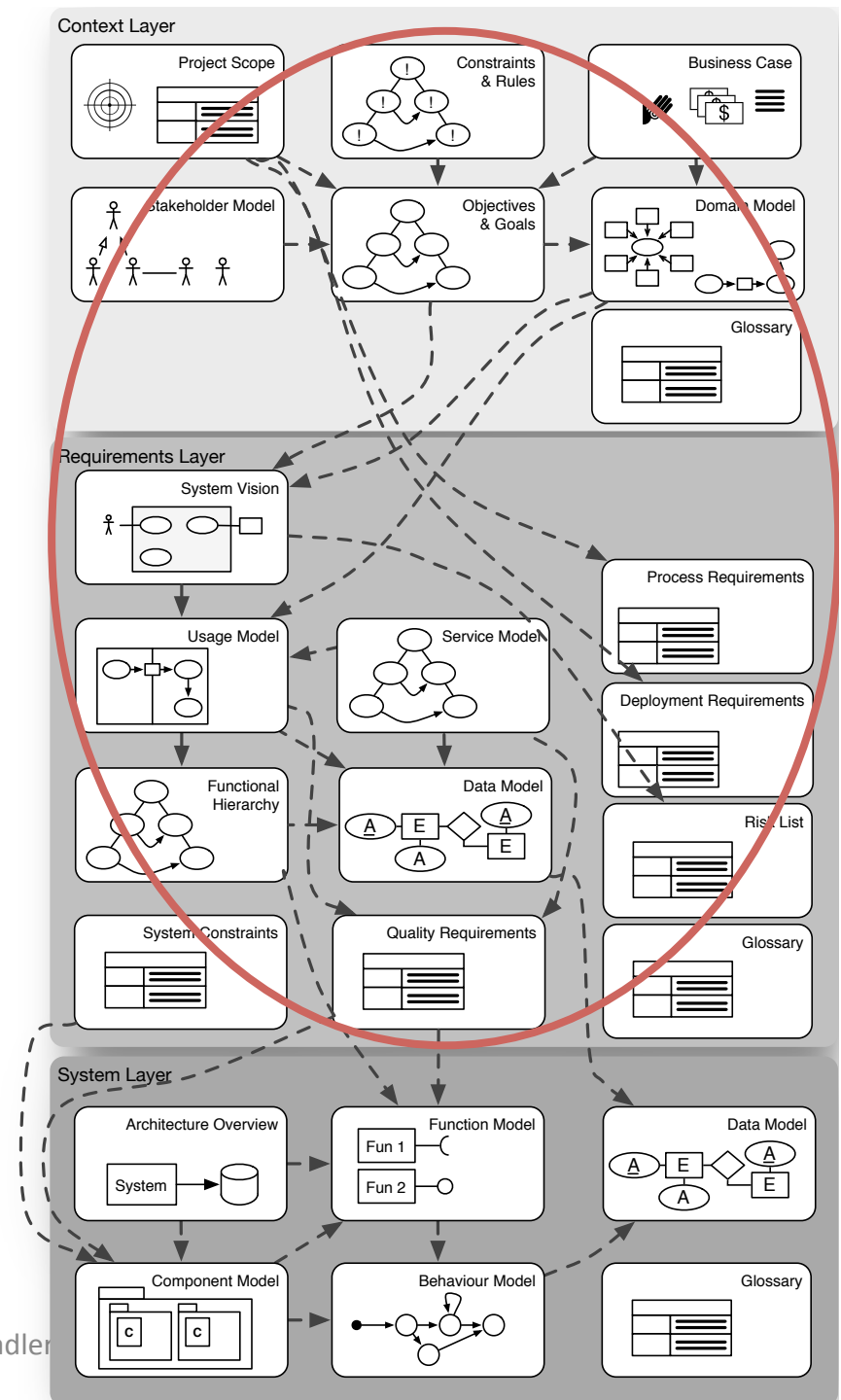


S. Adams E-mail: SCOTTADAMS@AOL.COM

7/1/96 © 1996 United Feature Syndicate, Inc. (NYC)

Learning Goals

- Foundations of quality assurance
 - Quality criteria for RE
 - Constructive and analytical Quality Assurance
- QA for Artefacts
- Techniques for Quality Assurance

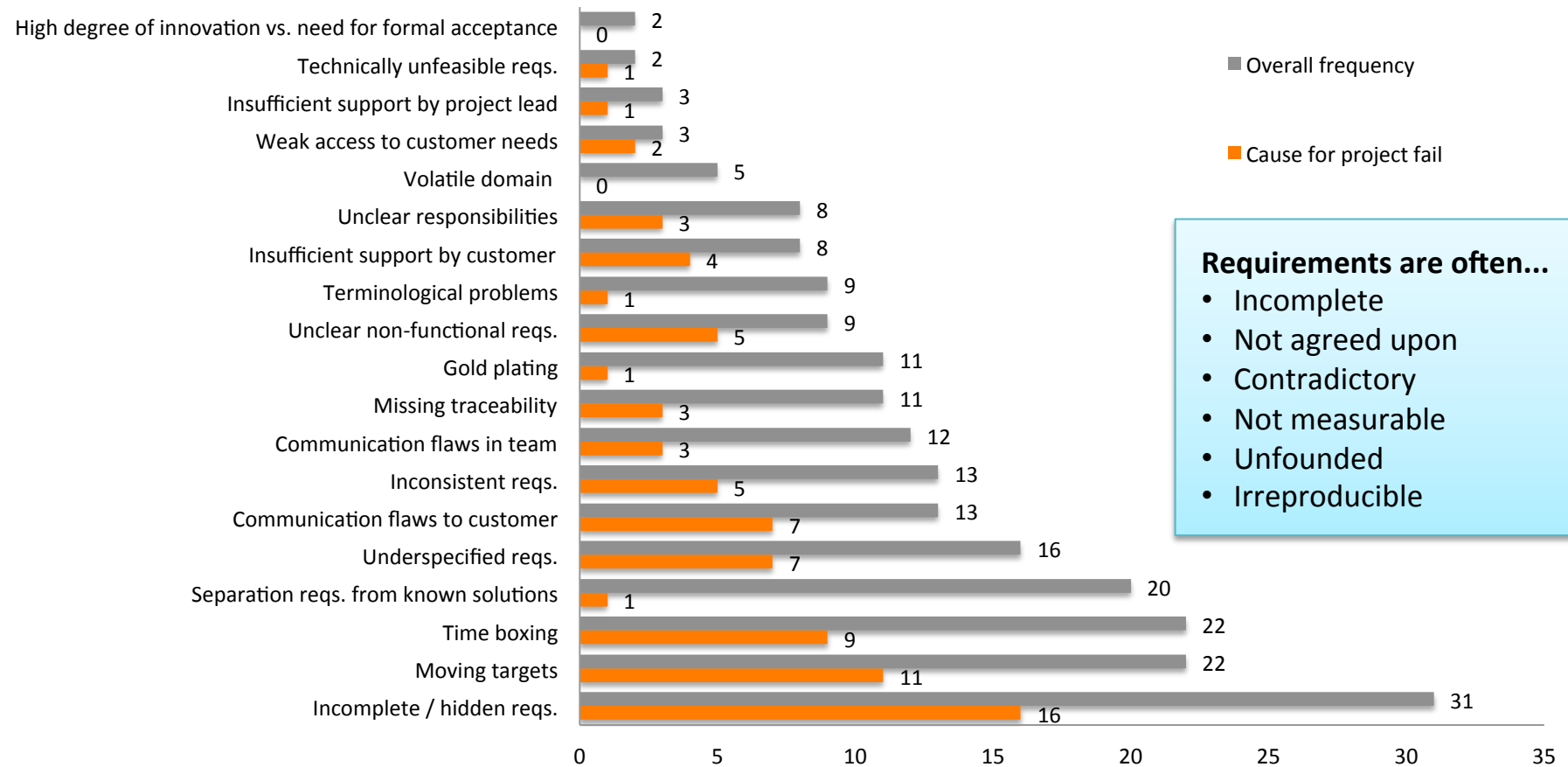


Overview: Quality Assurance

- Motivation and Terminology
- Quality of Requirements Documents
- Principles of Quality Assurance
- Techniques for constructive QA
- Techniques for analytical QA

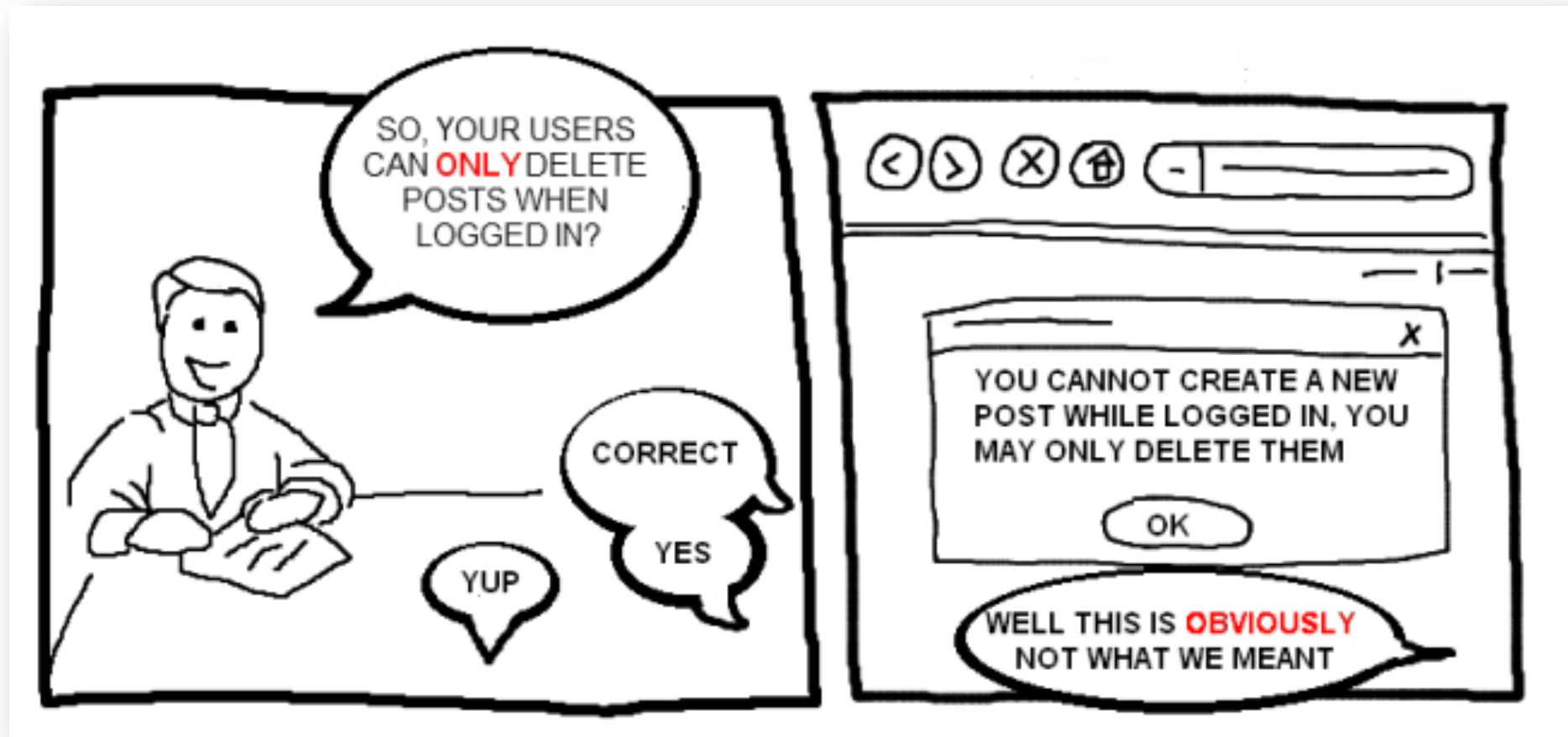
K Rayker, stock.xchng

Recap: Challenges in practice



Mendez et al. Naming the Pain in Requirements Engineering – The NaPIRE-Report

One of the many challenges...



Terminology in the context of quality assurance in RE

Quality defect

- **Incorrect (invalid) requirement:** Requirement that does not reflect the intention of the stakeholder (in the sense of „validity“)
- **Quality defect:** Requirement that can be valid, but has qualitative defects, e.g. missing measurability, low understandability, contradictory, ...
- **Interrelation of those two:**
 - Incorrect requirements are often hidden due to quality defects
 - Correctness of requirements often viewed as quality criterium

Validation and Verification

- **Validation:** Check of requirement w.r.t. correctness
- **Verification:** Check of system w.r.t. fulfillment of requirements
- Both are part of QA

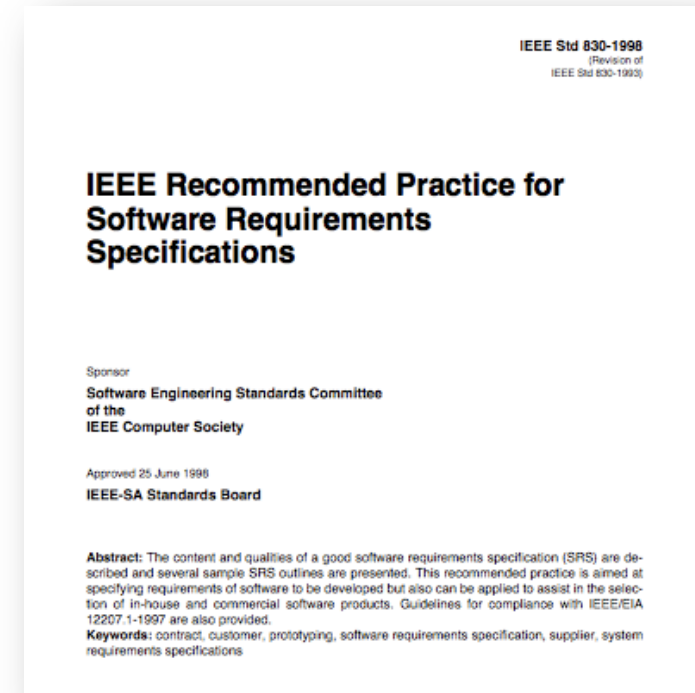
Quality Assurance in Requirements Engineering

- Def. QA in RE: Application of systematic measures for identifying quality defects and assuring the quality of the requirements specifications.

→ Check of quality criteria, e.g.:

- Correctness
- Completeness
- Consistency
- Traceability
- ... (see following slides)

→ The examination can be conducted constructive or analytical using a formal procedure.



Overview: Quality Assurance

- Motivation and Terminology
- Quality of Requirements Documents
- Principles of Quality Assurance
- Techniques for constructive QA
- Techniques for analytical QA

K Rayker, stock.xchng

Recap: Requirements, documents, artefacts, repositories

- A **requirement** is a demanded characteristic of a system or process.
- We distinguish
 - Syntactic representation: Text, table, diagram, formula
 - Semantic representation: Content – informal or formal
- A **requirements document or artefact**
 - Contains a number of requirements
 - Has a structure
- Delimitation: A **requirements repository** (database for requirements) serves for storing large sets of requirements (and requirements artefacts)

QA of requirements documents

- Quality of requirements documents is crucial for project success. → Why is that? What is based on them?
- We need specific procedures for QA
- Relevance of the quality criteria needs to be determined by the further use of the documents.



- We distinguish the quality of
 - Requirements documents / artefacts
 - Sets of requirements / statements
 - Individual requirements
 - Systems

The diagram illustrates the relationships between various models in the Requirements Engineering (RE) domain, organized into four layers: Content Layer, Requirements Layer, Systems Layer, and Systems Layer. The diagram shows how different models are interconnected and how they relate to each other across these layers.

Content Layer:

- Project Scope:** Represented by a globe and a document icon.
- Constraints & Values:** Represented by a cloud-like shape with three nodes.
- Business Case:** Represented by a document icon with a dollar sign.
- Stakeholder Model:** Represented by three stick figures.
- Objectives & Goals:** Represented by a cloud-like shape with three nodes.
- Domain Model:** Represented by a hierarchical tree structure.
- Domain:** Represented by a document icon with a grid.

Requirements Layer:

- System Vision:** Represented by a document icon with a cloud and a box.
- Usage Model:** Represented by a diagram showing a person interacting with a system.
- Service Model:** Represented by a cloud-like shape with three nodes.
- Formal Hierarchy:** Represented by a hierarchical tree structure.
- Use Model:** Represented by a diagram showing a person interacting with a system.
- Private Requirements:** Represented by a document icon with a grid.
- Deployment Requirements:** Represented by a document icon with a grid.
- Risk List:** Represented by a document icon with a grid.
- System Constraints:** Represented by a document icon with a grid.
- Quality Requirements:** Represented by a document icon with a grid.
- Domain:** Represented by a document icon with a grid.

Systems Layer:

- Architecture Overview:** Represented by a diagram showing a system and its components.
- Function Model:** Represented by a diagram showing a system and its functions.
- Component Model:** Represented by a diagram showing a system and its components.
- Behaviour Model:** Represented by a diagram showing a system and its behavior.
- Data Model:** Represented by a diagram showing a system and its data.
- Inventory:** Represented by a document icon with a grid.

The diagram shows how these models are interconnected and how they relate to each other across these layers. Dashed lines indicate relationships and dependencies between the models.

Asset	ID	Parent	Category	Initiative	Description	Notes	Status	Version	Is a	Is a	Is a	Is a	Is a	Is a
System	SYSD0001	External	External	External	Original data performance and security enhancement and new hardware	The hardware is to be replaced. Data migration and security are separate activities of the project and are not the focus of this project.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0002	Internal	Internal	Internal	IT infrastructure and process enhancement	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0003	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0004	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0005	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0006	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0007	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0008	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0009	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x
System	SYSD0010	Internal	Internal	Internal	Internal	The system is to be replaced. The system is to be replaced. The system is to be replaced.	Not implemented	1.0.0	x	x	x	x	x	x

Usability (NFR)	A system can be tested so that developers edit when ever possible.
Performance (NFR)	The perceived response time should not be too high.



Overview: Quality Assurance

- Motivation and Terminology
- Quality of Requirements Documents
- Principles of Quality Assurance
- Techniques for constructive QA
- Techniques for analytical QA

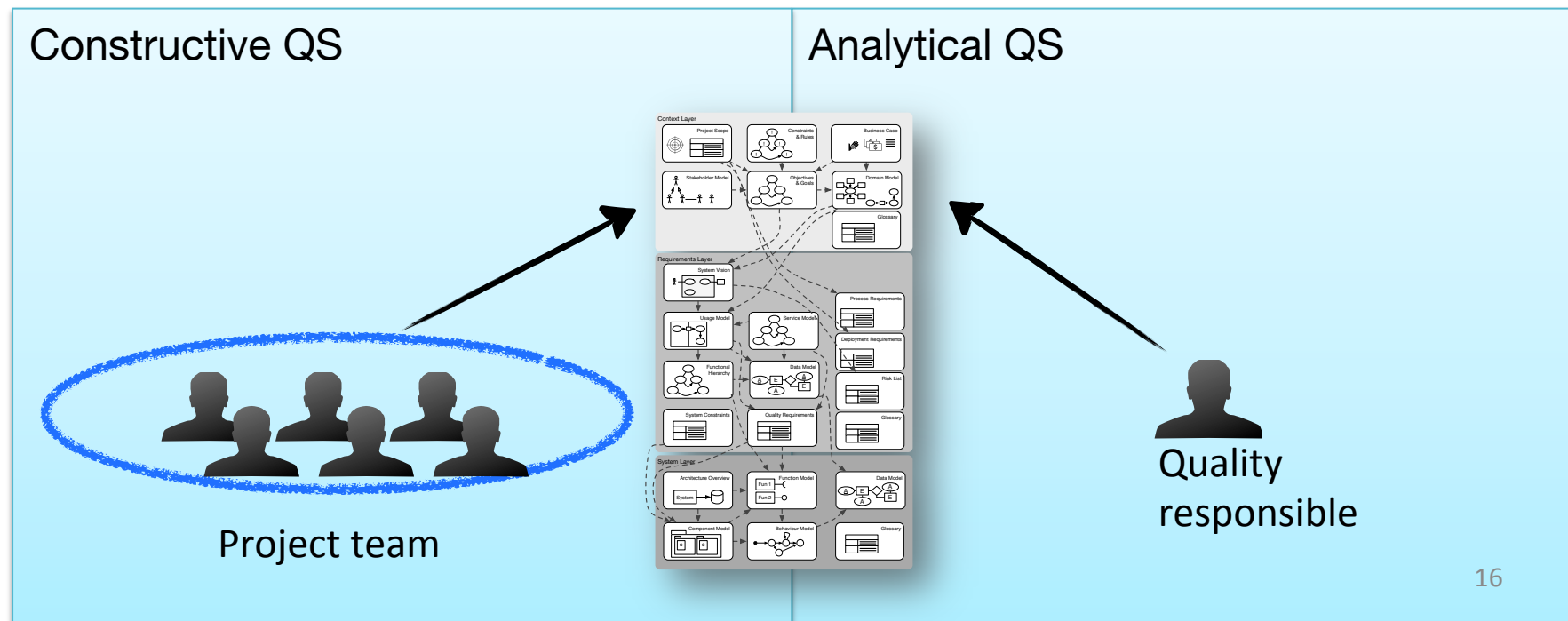
K Rayker, stock.xchng

Principle of constructive and analytical QA

Depending on the quality criteria, responsible for checking are:

- Project team members with domain knowledge during elaboration of the requirements, e.g. „correctness“ → this is called **constructive QA**
- External/neutral quality responsables who perform checks, e.g. „traceability“ and „understandability“ → this is called **analytical QA**

→ Which measures can you think of for performing either of these?



Classification of QA



Note: this is a general classification of QA, and not all of it applies to QA within RE.

Overview: Quality Assurance

- Motivation and Terminology
- Quality of Requirements Documents
- Principles of Quality Assurance
- Techniques for constructive QA
 - Reference Models
 - Quality criteria according to IEEE 830
 - Linguistics
 - Guidelines
- Techniques for analytical QA

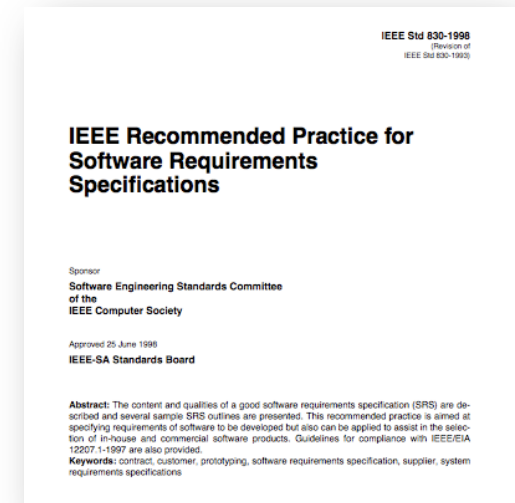
K Rayker, stock.xchng

Reference Models

- AMDiRE
- IEEE 830
- Cockburn template
- UML2 standard

Constructive QA in RE: Examination of Quality Criteria acc. to IEEE 830-1998

- **Completeness:** Systematically run through all cases (You can *only* find incompleteness, not guarantee completeness!)
- **Consistency:** Relate all to one system model
- **Unambiguity:** Check phrasing
- **Correctness:** Validation
- **Structuredness:** Examine structure
- **Traceability:** Are requirements sufficiently linked?
- **Changeability:** Can expected changes in requirements be conducted efficiently?
- **Understandability:** Check formulations
- **Agreed upon:** Check with stakeholders



Checklist: Questions and Criteria

- Completeness:
- Consistency:
- Unambiguity:
- Correctness:
- Structuredness:
- Traceability:
- Changeability:
- Understandability:
- Agreed upon:

Linguistics in RE

- Classification of **linguistic quality defects**
 - lexical/ontological (what does „green“ mean?)
 - syntactic (“I saw the man on the hill with a telescope”)
 - semantic (“All persons have a unique national insurance number”)
 - pragmatic (“The trucks shall treat the roads before they freeze”)
 - weak phrases: (“as soon as possible”)
 - Omission or generalization
- **Syntax patterns**
 - [when?] [under what conditions?]
THE SYSTEM SHALL | SHOULD | WILL
<process> <thing to be processed> [<process detail>*]

Exercise: Improve phrasing



- 1. The system shall respond as fast as possible.*
- 2. Students take 10 courses per semester.
Students take 1000 courses per semester.*
- 3. Shortly before the due date the medium is extended, unless somebody else reserved it.*

Phrasing: Do's and Don'ts

1. *The system shall respond as fast as possible.*



In 90% of all cases, the system shall respond to all queries within 3s.



2. *Students take 10 courses per semester.*

Students take 1000 courses per semester.



Every student takes 10 courses per semester.



3. *Shortly before the due date the medium is extended, unless somebody else reserved it.*



Three days before the due date, the system checks whether the medium has been reserved. If not reserved, the system extends the lending period.



Guidelines & Checklists

How would you write a guideline and a checklist for ... ? Team up!

- Stakeholder Model
- Goal Model
- System Vision
- Usage Model
- Non-functional Requirements

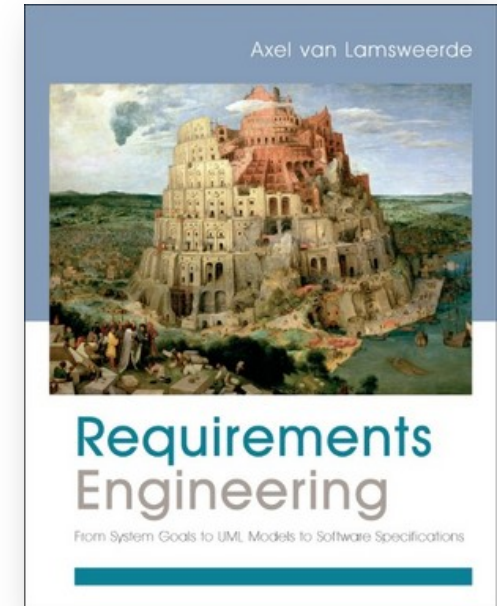
Overview: Quality Assurance

- Motivation and Terminology
- Quality of Requirements Documents
- Principles of Quality Assurance
- Techniques for constructive QA
- Techniques for analytical QA
 - Checklists
 - Quality Gates
 - Fagan Inspections

K Rayker, stock.xchng

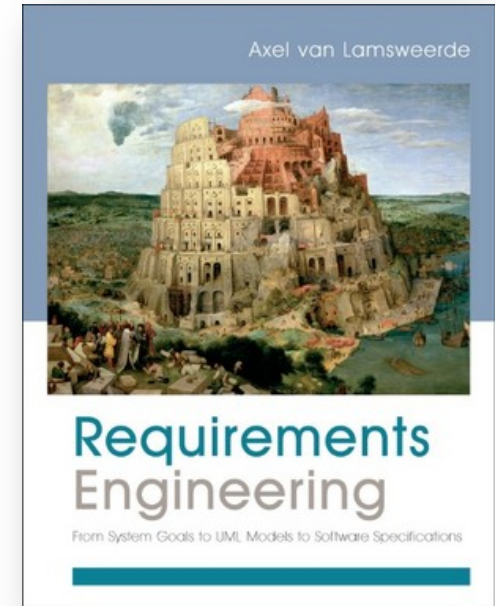
Exemplary Check list according to Lamsweerde (1/2)

- General check: *Must be clear "what, who, when, where"*
- Defect-based criteria (as in IEEE 830):
 - *Contradiction*
 - *Inadequacy*
 - *Unmeasurability*
 - *Unfeasibility*
 - *Poor structuring*
 - ...
- Quality-specific criteria:
 - *Is there any unspecified response in this operation to not receiving an expected input value, or receiving it too early or too late?*
 - *Does the logical OR of the input condition on this operation form a tautology?*
 - ...

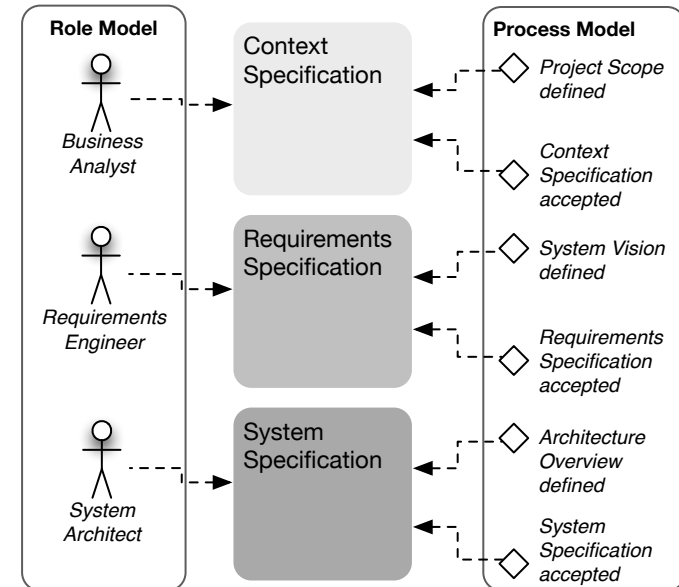


Exemplary Check list according to Lamsweerde (2/2)

- Domain—specific criteria:
typical issues in the particular domain
- Content-related criteria:
 - Templates
 - *all fields filled*
 - *identifier user consistently*
 - *statement type correct*
 - ...
 - Graphical notations
 - *data flow consistent*
 - *ER Diagram declaration*
 - ...
 - Formal specifications



Quality Gates



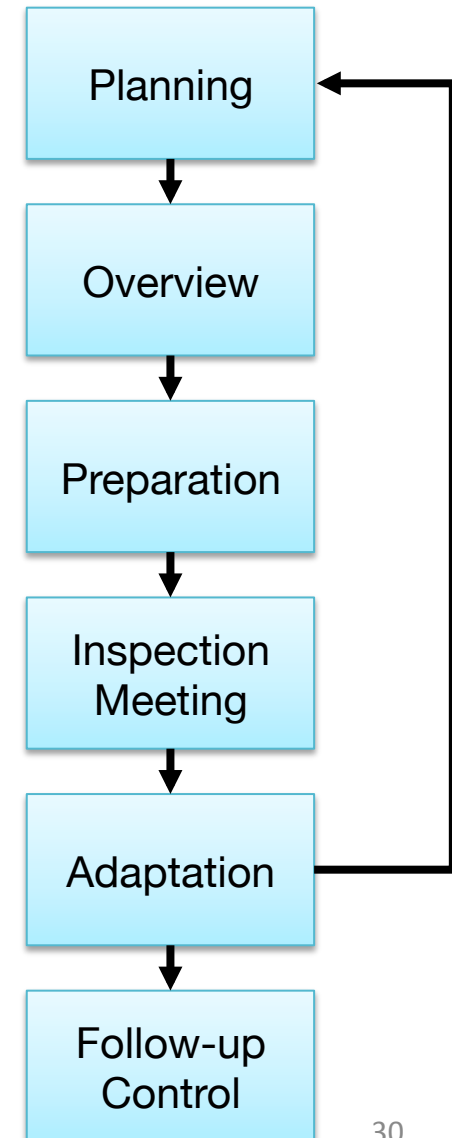
Specific milestone in a software project that checks

- **Content:** The „usual“ quality criteria: Completeness, consistency, ...
- **Documentation:** Compliance with format, understandability, unambiguity, ...
- **Accordance:** Every requirement agreed upon, conflicts resolved, ...

Fagan Inspection

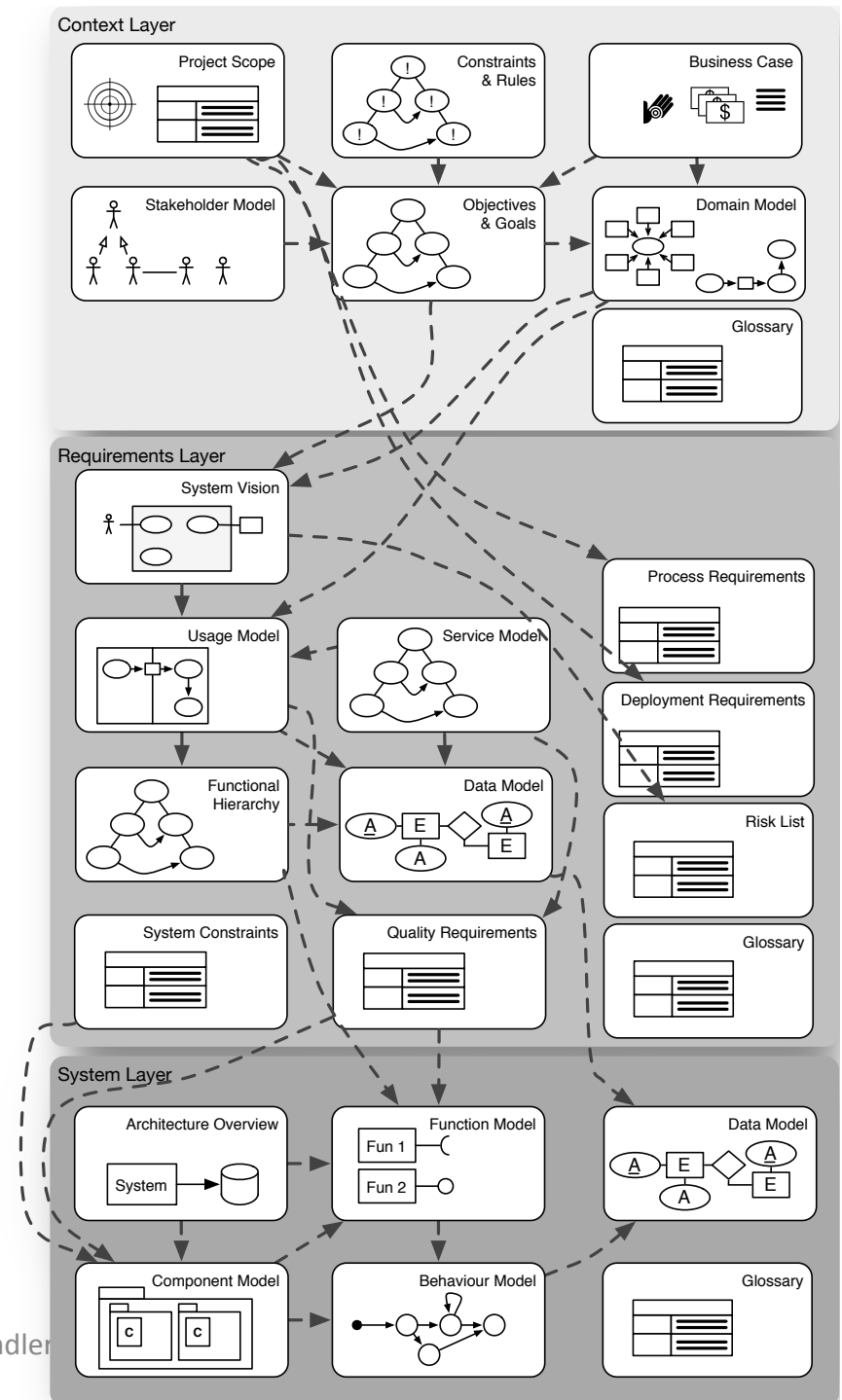
The term *Fagan inspection* refers to a structured process of trying to find defects in development documents. It includes the following phases:

- **Planning:** Moderator plans review process
- **Overview:** Author describes the background of the document under inspection
- **Preparation:** Every reviewer examines the document in order to find defects.
- **Inspection meeting:** A specific reader walks through the document chapter by chapter, and the inspectors point out found defects.
- **Adaptations:** The author of the document corrects the found defects according to the action plan agreed upon in the meeting.
- **Follow-up control:** The inspectors check whether the defects were fixed correctly.



Take-away: QA

- Definitions
 - Quality Assurance
 - Quality Defect
- Constructive QA
 - Guidelines and criteria
 - Reference models
- Analytical QA
 - Quality gates
 - Fagan inspection
 - Checklists



THE END



NO, IT ISN'T

Requirements Negotiation

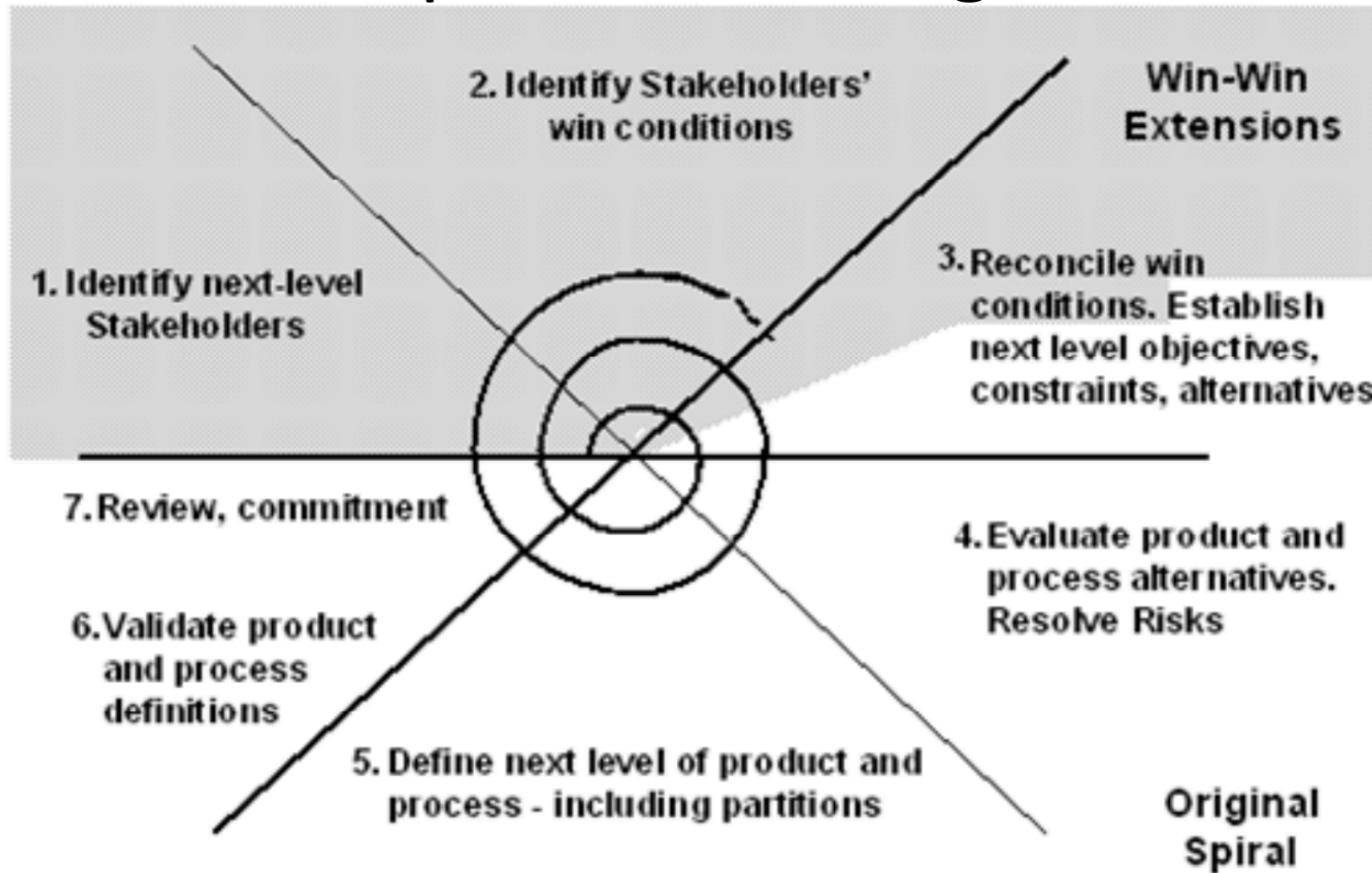


Figure 8.1: The WinWin Spiral Model of Software Engineering includes front-end activities (gray) that show where objectives, constraints, and alternatives come from. This lets users more clearly identify the rationale involved in negotiating win conditions for the product.