SIEMENS
Ingenuity for life

PLM and Innovation Excellence

Learning Campus

Your partner for Business Learning

Siemens Core Learning Program

Requirements Engineering

Authors:
Rüdiger Kreuter, CT | Beatrice Hwong, CT | Heinrich Helmut Degen, HC | Martin Herget, CT | Georgi Markov, CT

Were are we now?

There's no point in being exact about something, if you don't even know what you're talking about.

[John von Neumann]

Source: http://commons.wikimedia.org/wiki/J%C3%A1nos_Lajos_Neumann?uselang=de

SIEMENS
Ingenuity for life

Learning objectives

- Understand that requirements are a means of communication between customer and supplier to gain common understanding of the customer‘s needs
- Understand the importance of Requirements Engineering and a Test Architects role in it
- Know attributes of good requirements
- Know how to write and review good requirements

SIEMENS
Ingenuity for life

Agenda

Introduction

Writing Good Requirements

Characteristics of good requirements

Reviewing Requirements

Summary

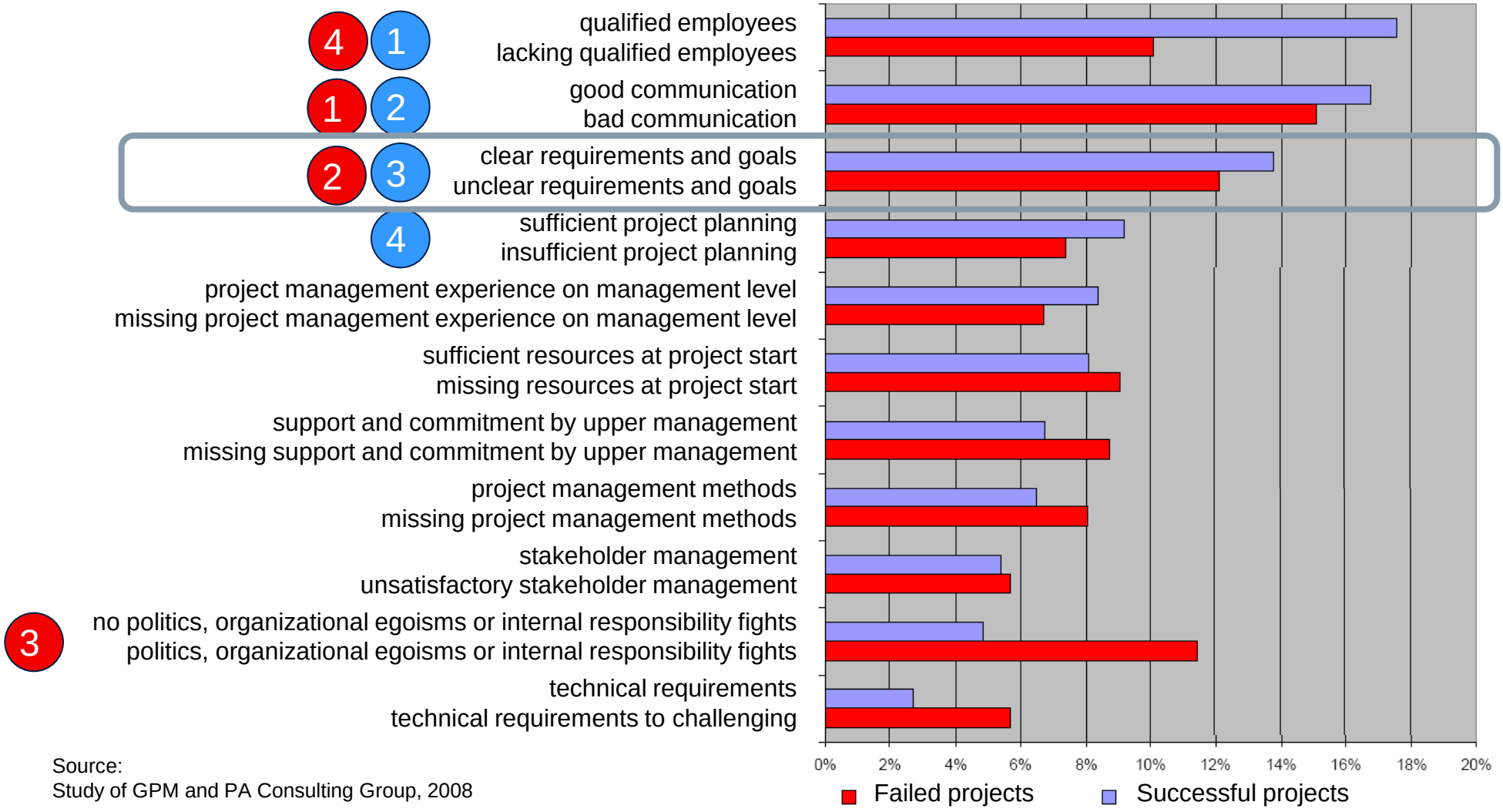
Why Requirements Engineering?

Vision: Have a common understanding of what the customer needs and expects and what the supplier will provide.

Requirements are the main vehicle to communicate and discuss the customer needs and expectations with the supplier team.

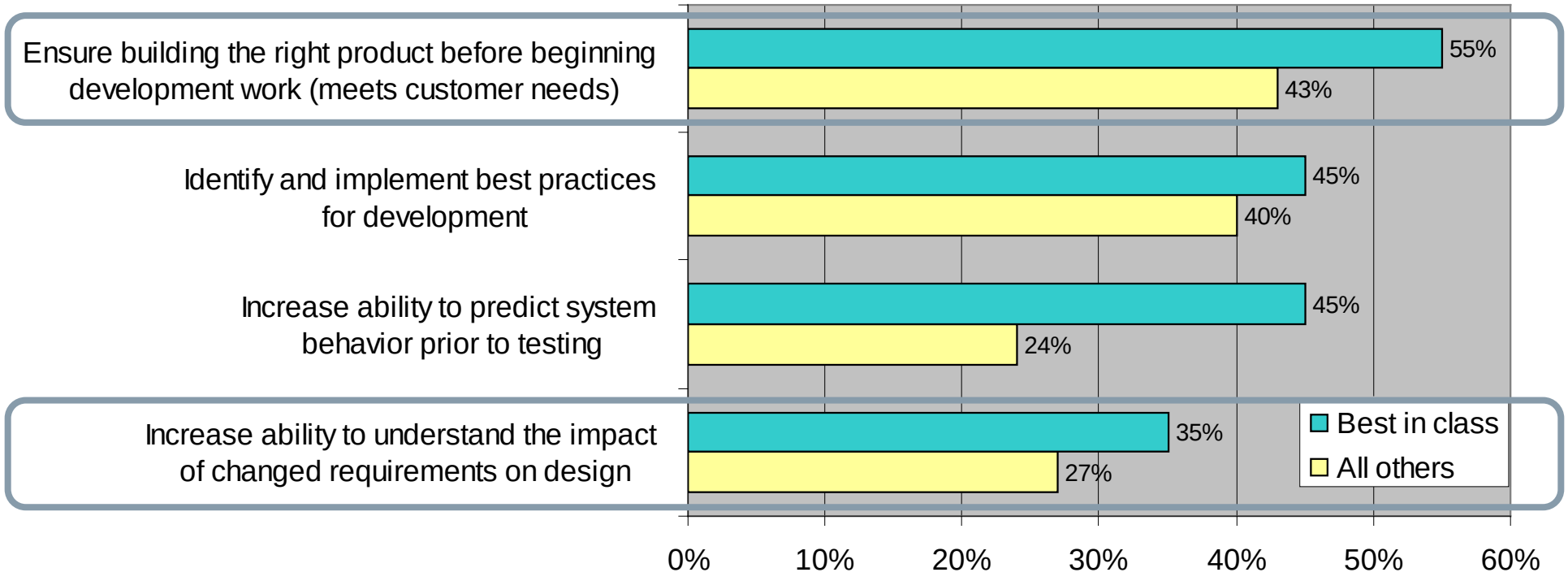
What makes Projects Successful?
What causes failure?

SIEMENS
Ingenuity for life



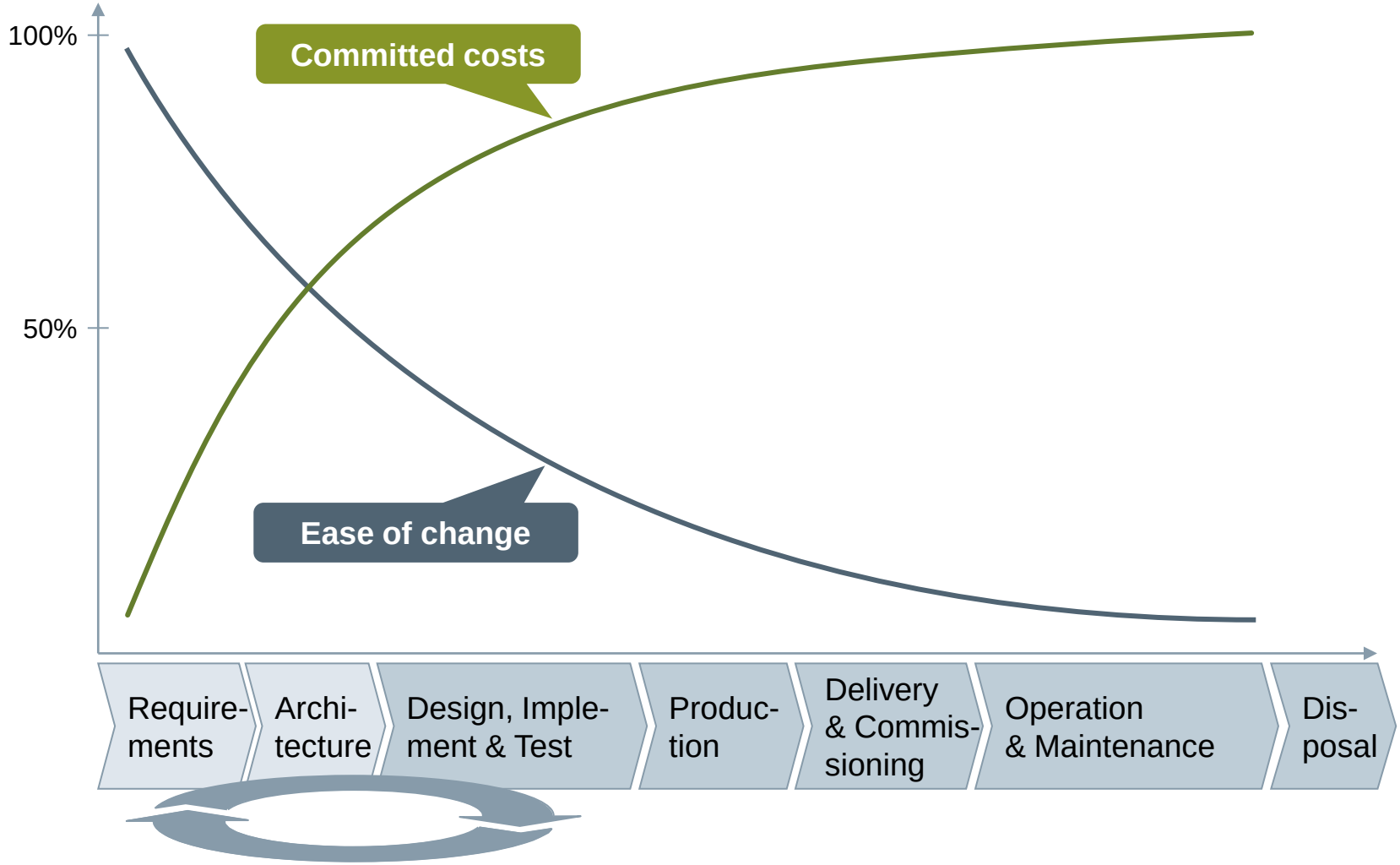
Source:
Study of GPM and PA Consulting Group, 2008

Top strategies for System Design



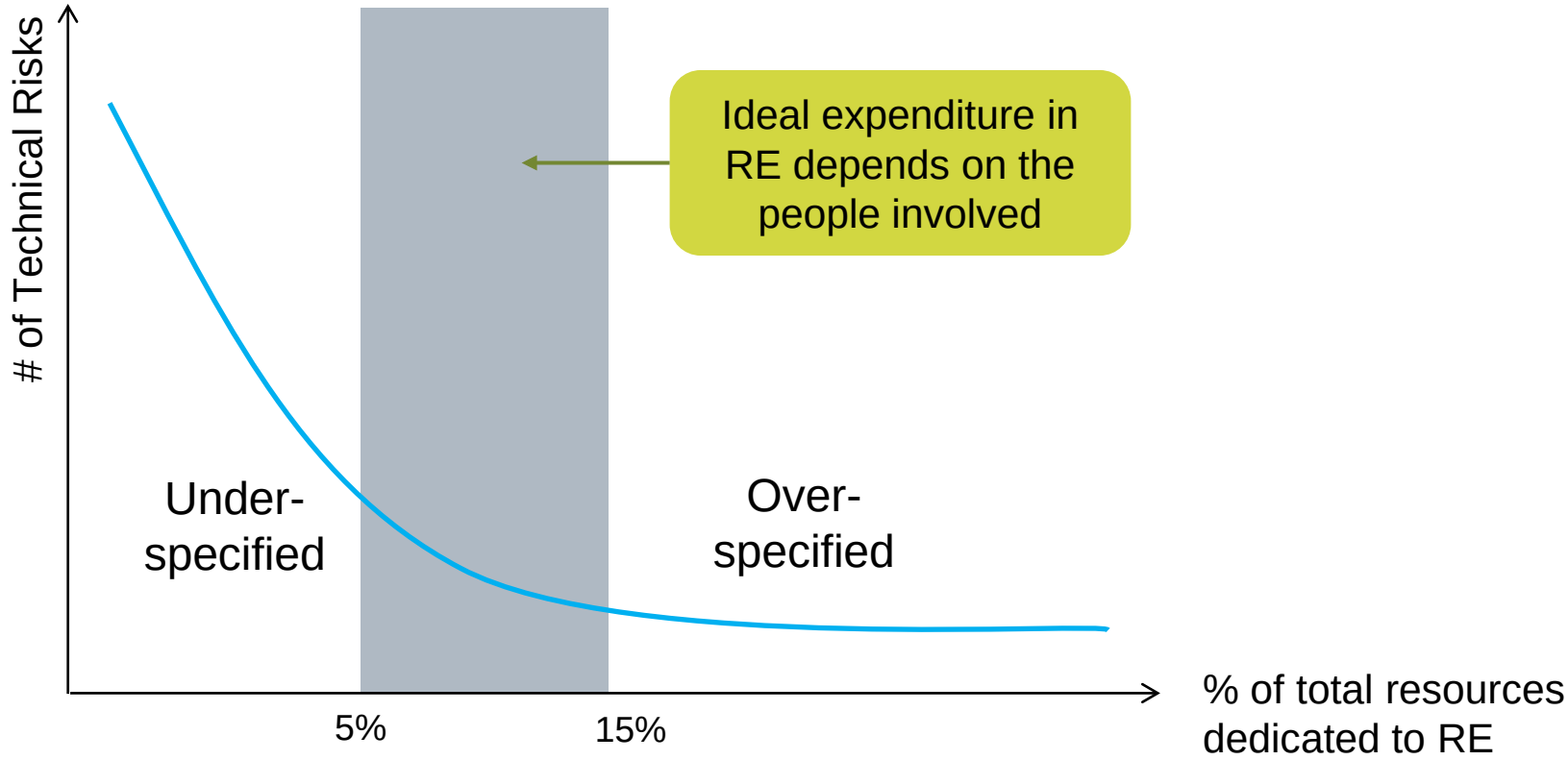
Source: Aberdeen *Group*, July 2011

Influence of an effective Requirements Engineering

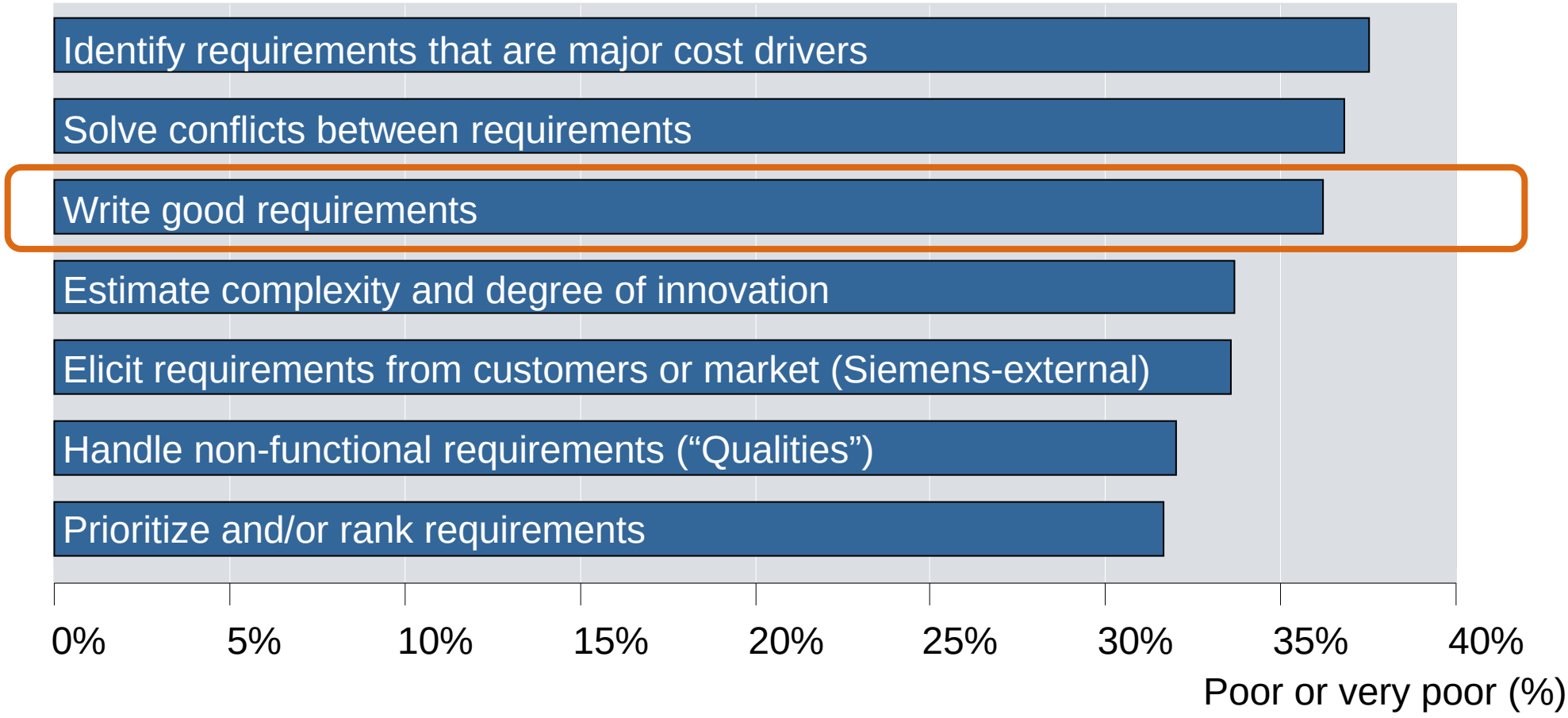


Determine the sweet spot of just enough RE

Requirements Engineering (RE) is risk mitigation investment, the return is less unplanned and costly rework.



Siemens requirements survey: Top 7 RE challenges tend to be problems in the early phases



Source: Siemens Requirements Survey 2009,
Top 7 RE tasks selected as poor or very poor

What do you need to know to build a system?

System Vision

Stakeholder

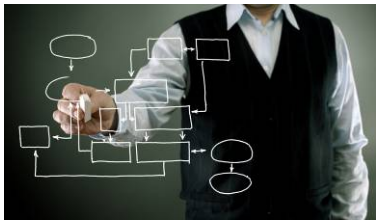
Purpose of RE

System Boundaries

Common Understanding

Agreement

Customer Needs



Quality Attributes (Non-Functional Requirements)

Baselining and Tracing

Artifacts

Prioritization

What is Requirements Engineering?

SIEMENS
Ingenuity for life

Siemens definition*

> “Requirements Engineering is a **discipline** with the purpose of developing a **common understanding** about the **right product / system** among stakeholders [1)] and **documenting** it sufficiently.”

- **Discipline** = a well defined set of engineering activities
- **Common understanding** = having a shared vision of the result
- **Right product / system** = what the customer needs and expects
- **Documenting sufficiently** = just enough
- **Stakeholders** include e.g. customers, users, product manager, project manager, architects, developers, testers, marketing and service experts

* This definition is based on different sources such as ISO / IEEE / INCOSE / DoD (US Department of Defense) and simplified to a few basic concepts, adapted for the Siemens definition.

What is the negative project impact when we lack effective RE?

SIEMENS
Ingenuity for life

Lack of effective RE	Impact
Unclear system scope	Over- or Under-engineering, missed deadlines
Insufficient customer or stakeholder involvement	Misunderstanding of the customer needs
Ignored Standards, Guidelines, and Regulations	Rejection, blocked market access
High requirements volatility (churn)	Unplanned changes to Design, Implement, and Test
Low quality requirements, poorly organized documentation	Design, Implement, and Test do not reflect the intent of the requirements, confusion
Non-Functional Requirements (NFRs) not properly documented	System architecture fails to meet important NFRs
Insufficient tracing	Difficult to do derivation, impact analysis, and coverage
Ignored (“write-only”) requirements	Wasted effort, Design, Implement, and Test have their “own” separate requirements

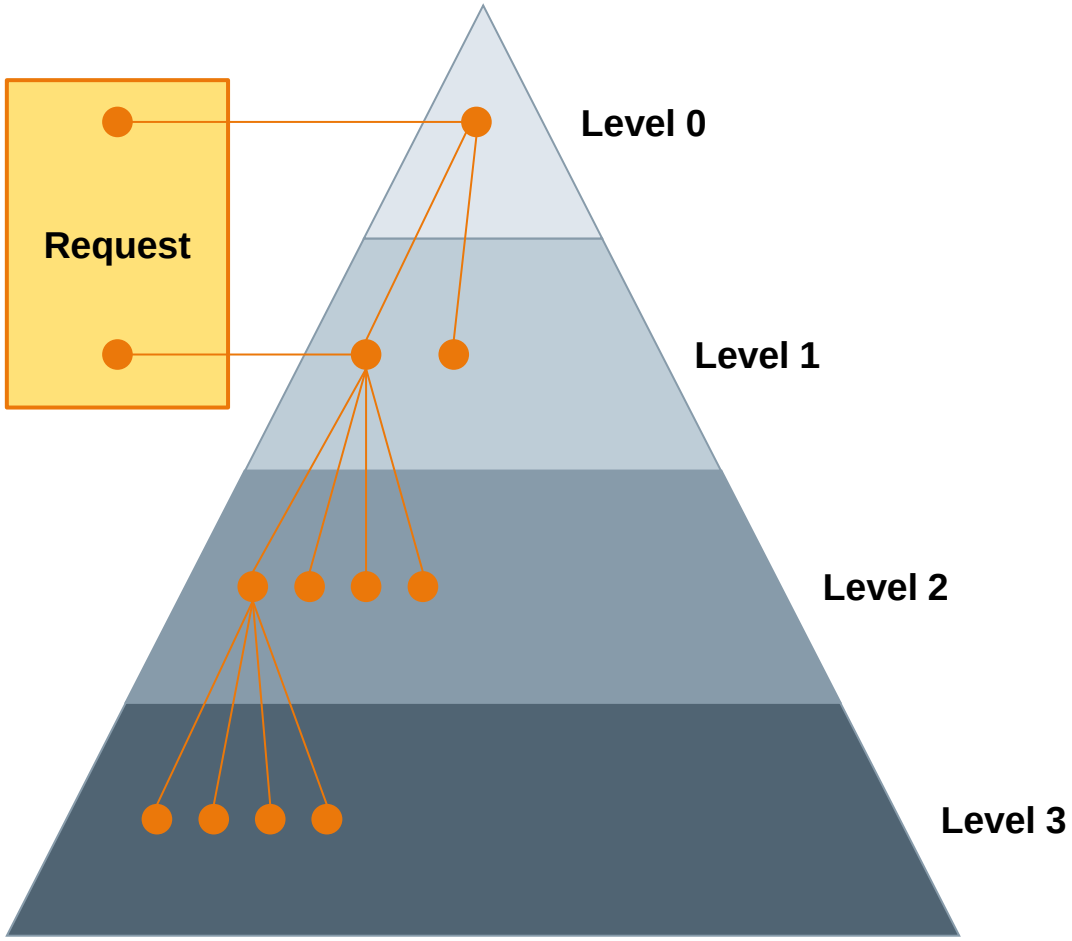
What is the positive project impact of effective RE?

SIEMENS
Ingenuity for life

Effective RE	Impact
Well defined scope and system boundary	Customer acceptance; contained costs
Ongoing involvement of customers and stakeholders	Fewer misunderstandings
Standards, Guidelines, and Regulations are addressed	Compliance with regulatory agencies
Managed change	Fewer changes mean less wasted effort and time, avoid scope creep
Well written requirements	Clear and consistent requirements, better estimate of development effort & delivery dates, common vocabulary
Non-Functional Requirements (NFRs) properly documented	System architecture satisfies NFRs
Sufficient tracing	Sufficient completeness, impact analysis, and derivation
Supplier team jointly owns the requirements and actively uses them	Design, Implementation, and Test use the owned requirements

A Framework for Requirements Engineering

- Requirements are usually organized on **different levels**
- Different levels contain **different kind** of requirements
- Requirements on different levels are **related to each other** through traces
- The levels help to organize the effort in a project and allow early baselines



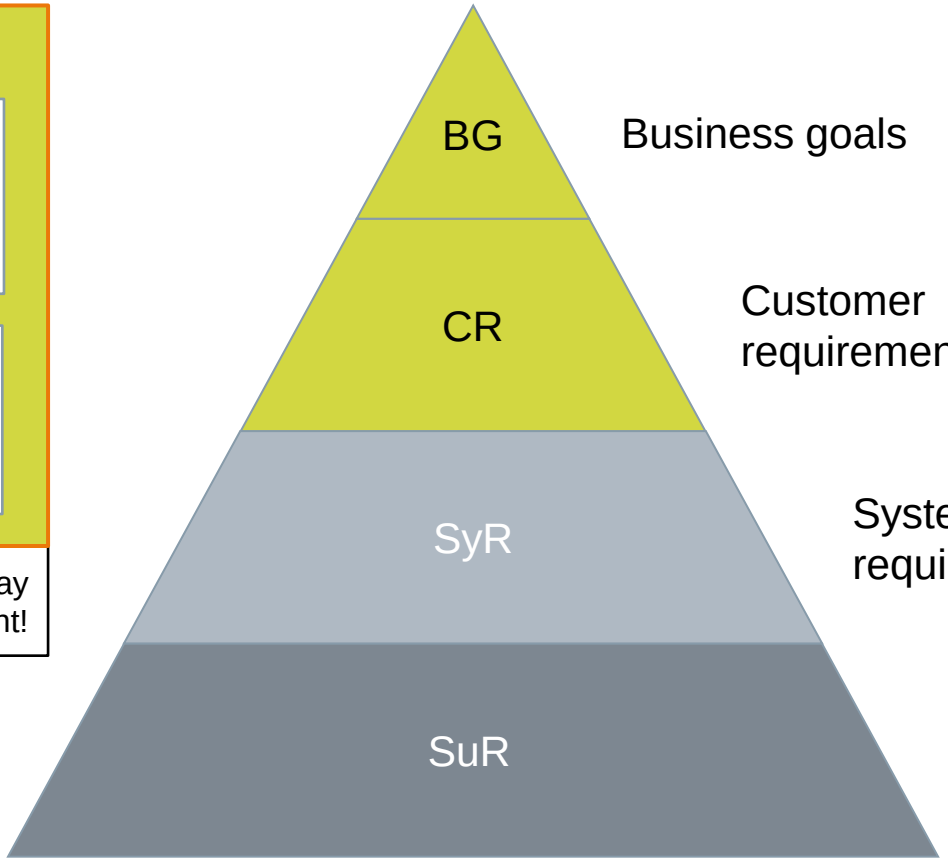
Requirements Pyramid

Request

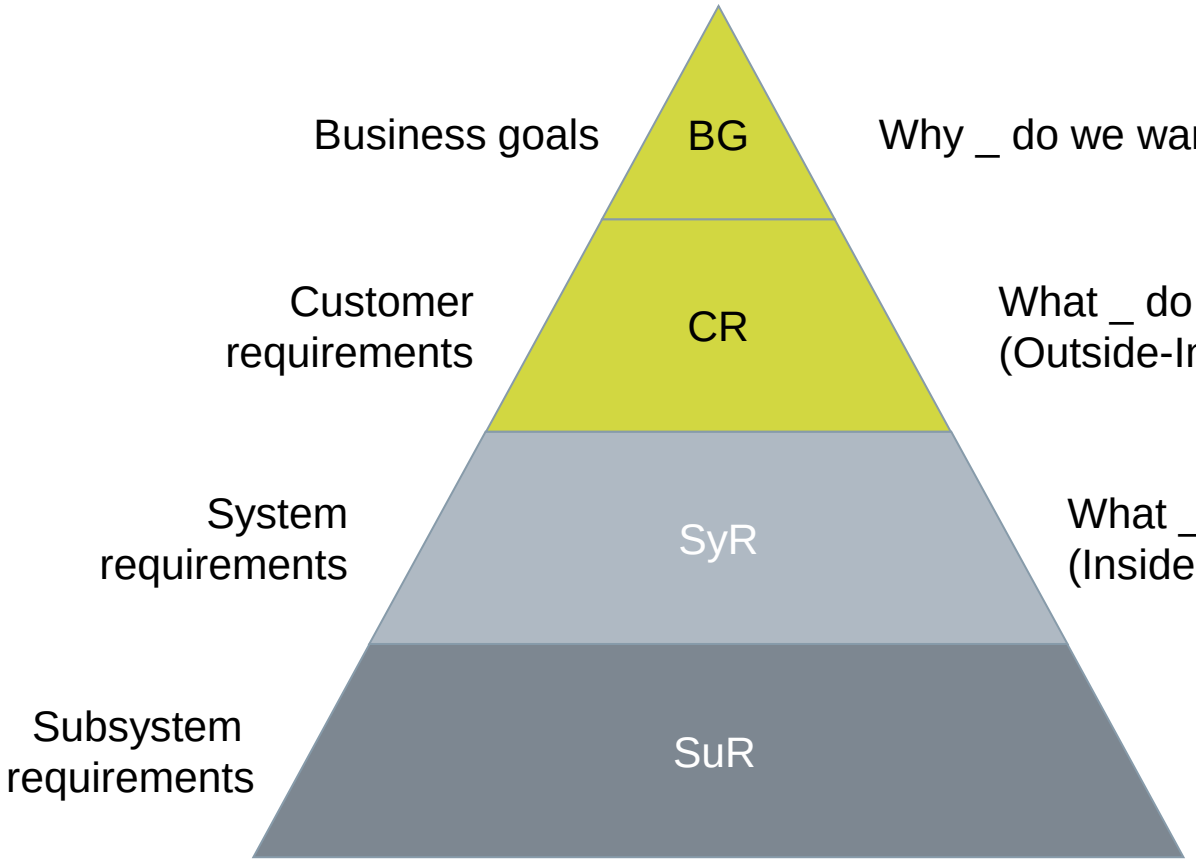
Customer, Marketing, Business model

Standards, guidelines and regulations

Note: Not every request may become a requirement!

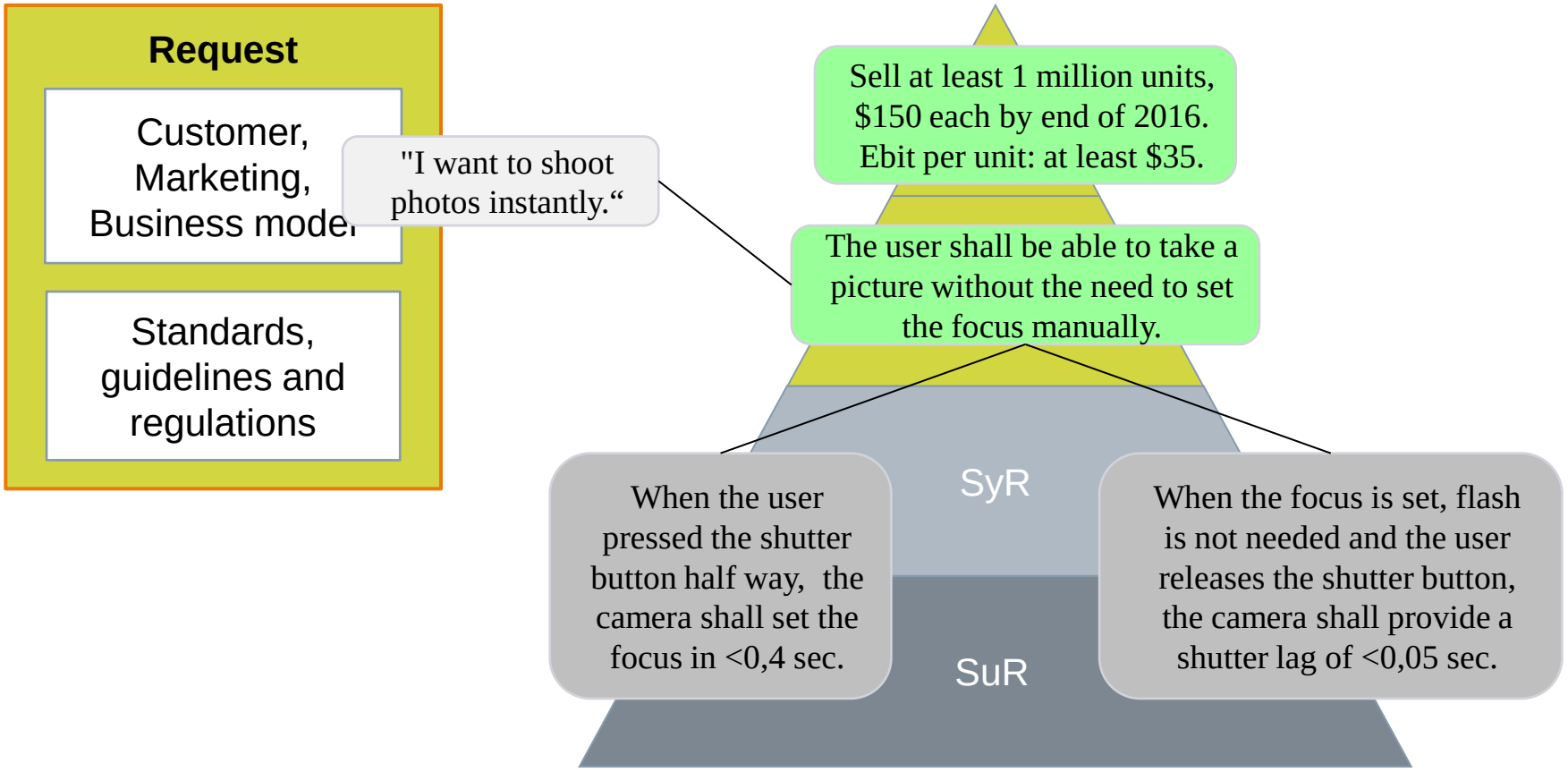


Requirements Pyramid
Reasoning of levels



(Sub-) System Requirements are a response to Customer Requirements; i.e. finally they should reflect the intent of the customer requirements.

Requirements Pyramid Example



Major steps in Requirements Engineering

- Collect from internal and external sources
- Involve Customer, use surveys
- Involve Stakeholders
- Use structured methods

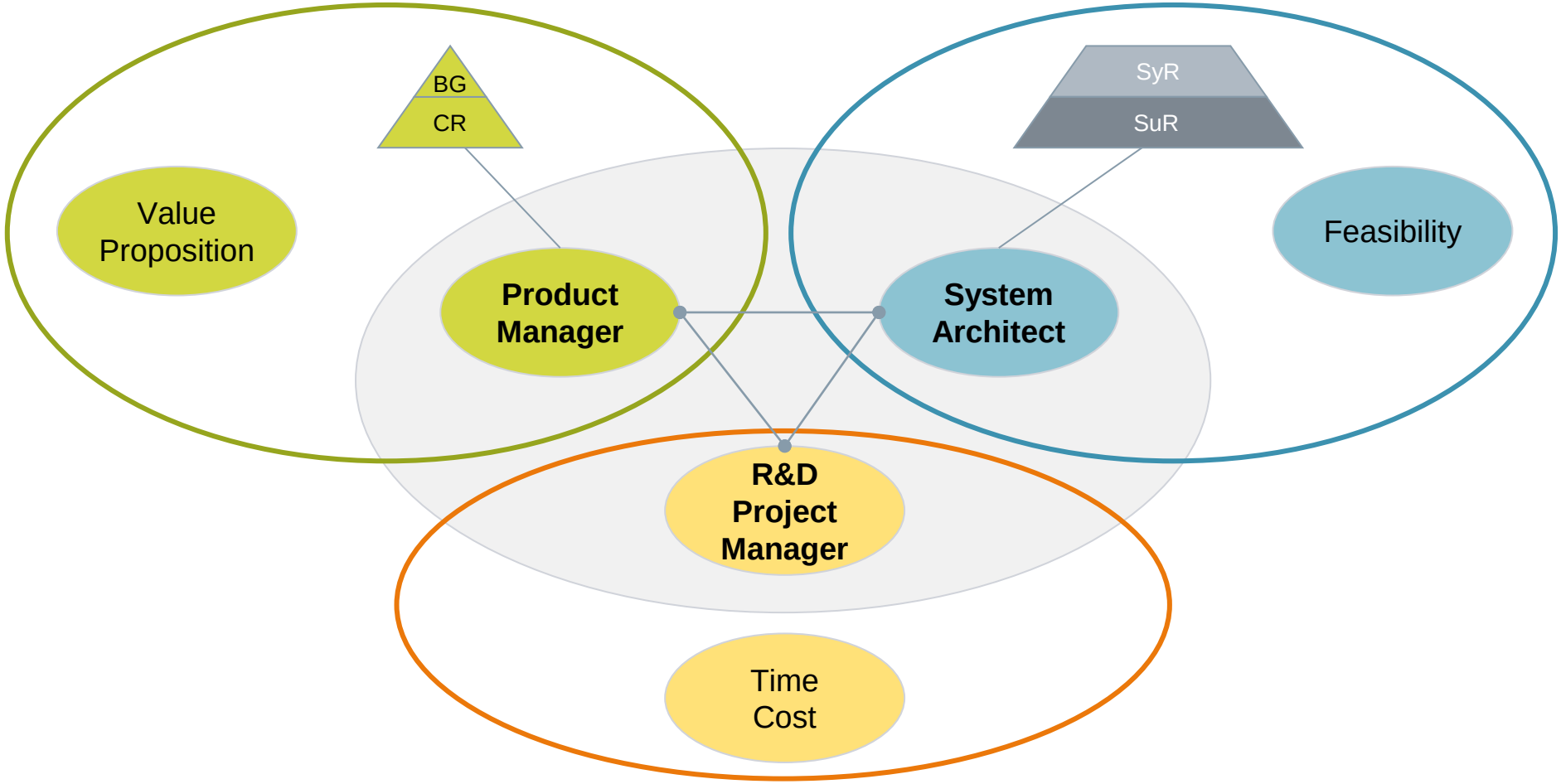
- Remove ambiguity
- Categorize
- Prioritize

- Clarify feasibility

- Proof that the requirements will fulfill the (real) customer's needs
- Review requirements (all stakeholders)
- Add trace links

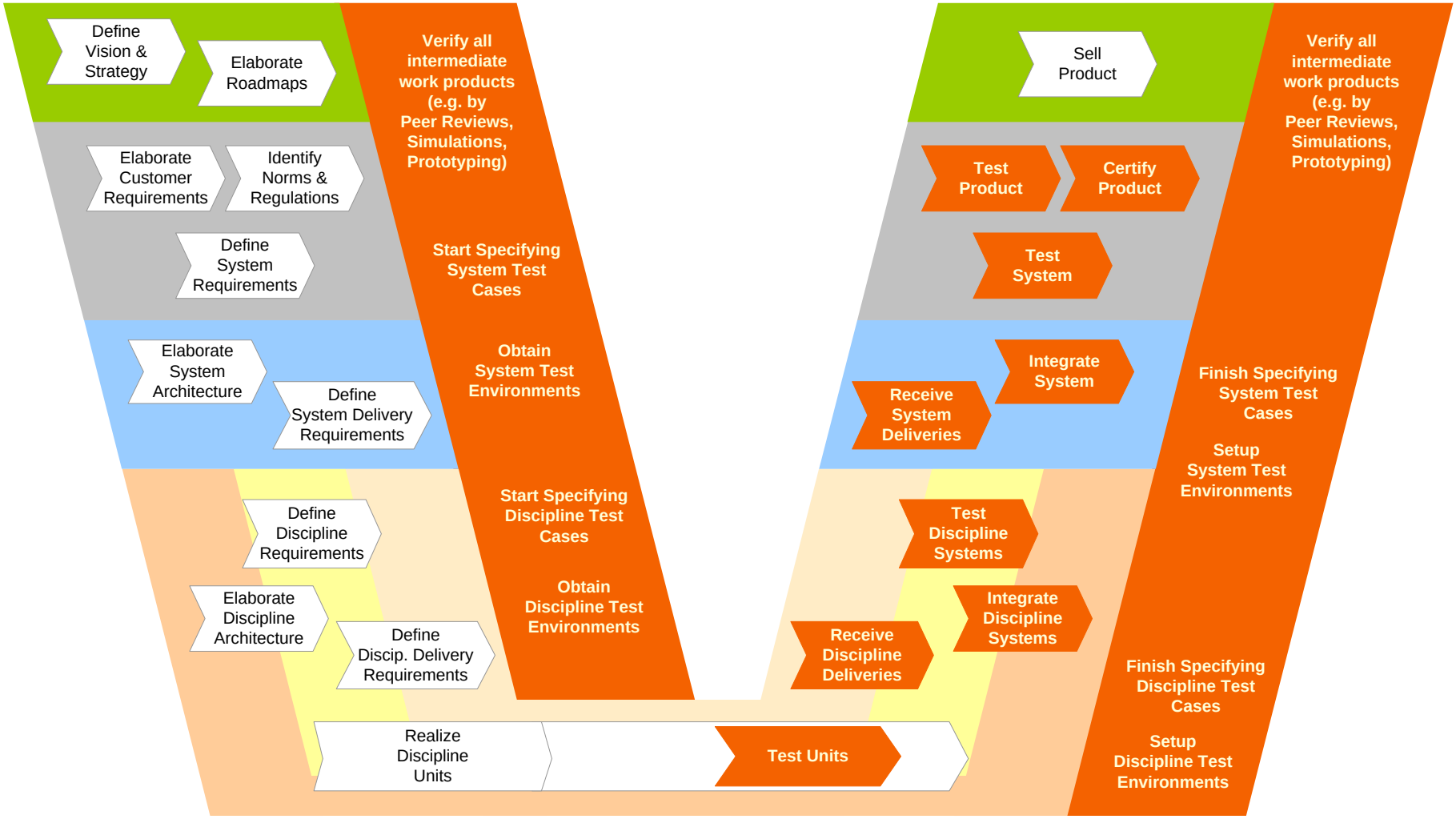
- Allocate requirements to projects or responsibles

Key roles have different interests in different requirement types

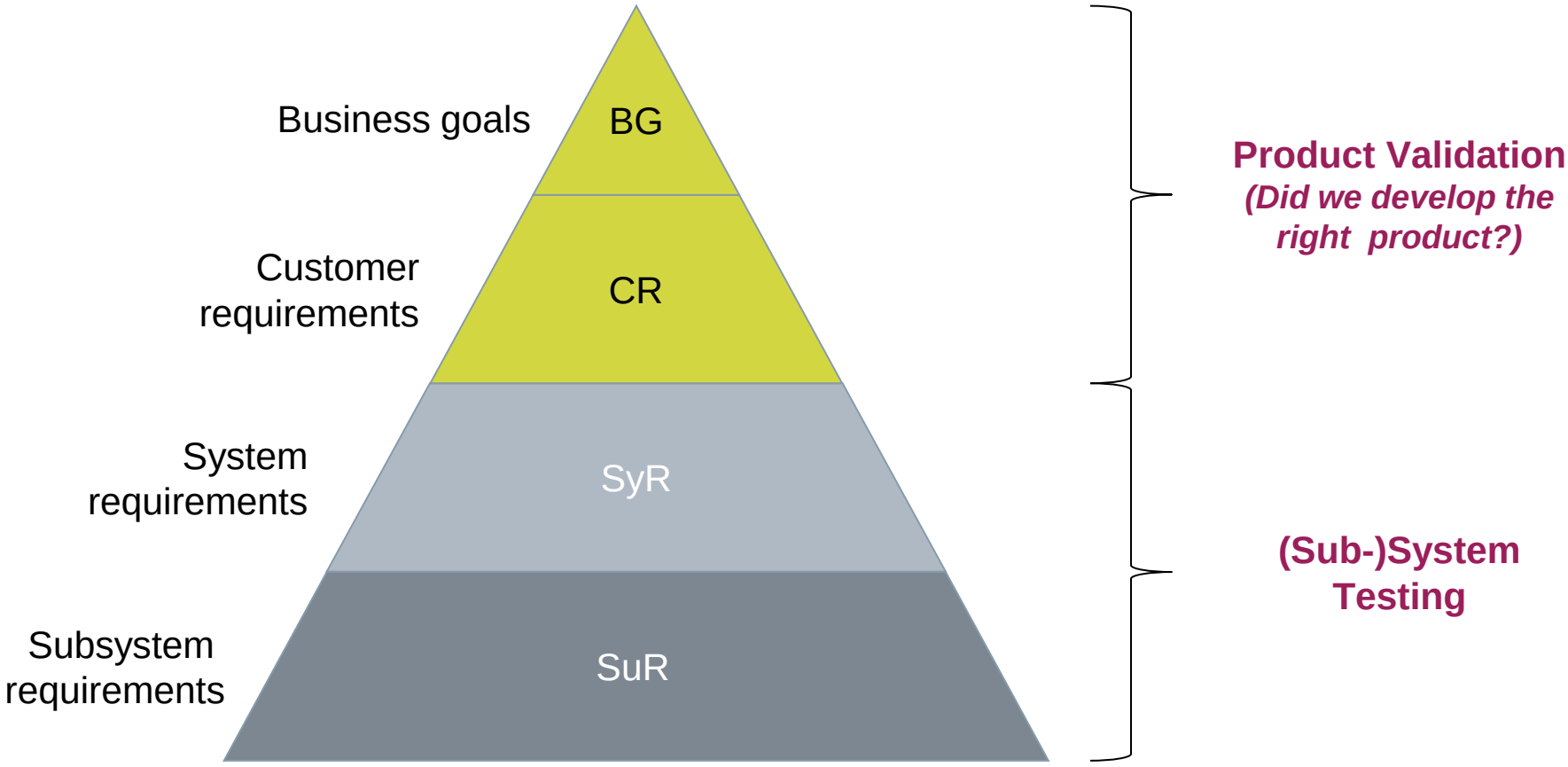


What is a Test Architect's interest in Requirements Engineering ?

System Development Process Model
𝕎-Model for Test & Quality



Requirements pyramid
Mapping to Verification / Validation



SIEMENS
Ingenuity for life

Agenda

Introduction

Writing Good Requirements

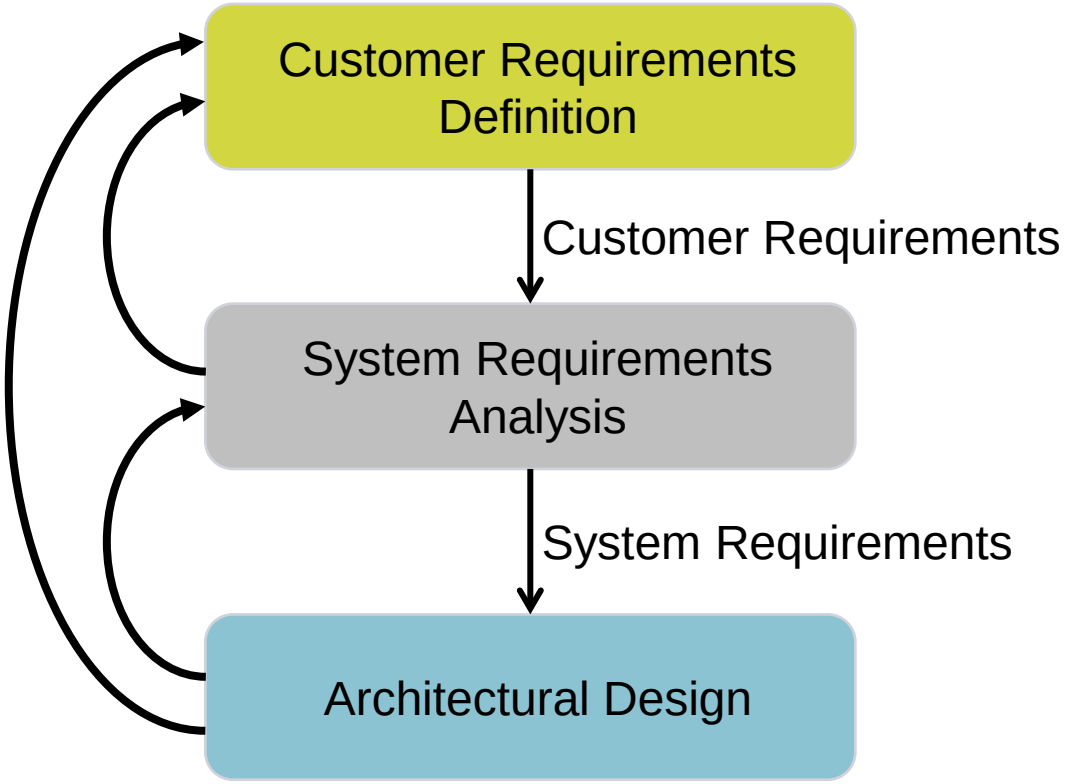
Characteristics of good requirements

Reviewing Requirements

Summary

Iterative System Requirements Development

Handshakes and iterations between Product Manager (up stream), Architects and discipline experts (down stream) are critical for defect-free system requirements.



Based on ISO 29148

Approach for developing System Requirements

1. Define **functional boundaries** of the system in terms of behavior and properties;
   i.e. define the system interfaces (static and dynamic).

2. Define **each function** that the system is required to perform;
   i.e. define the functional requirements.

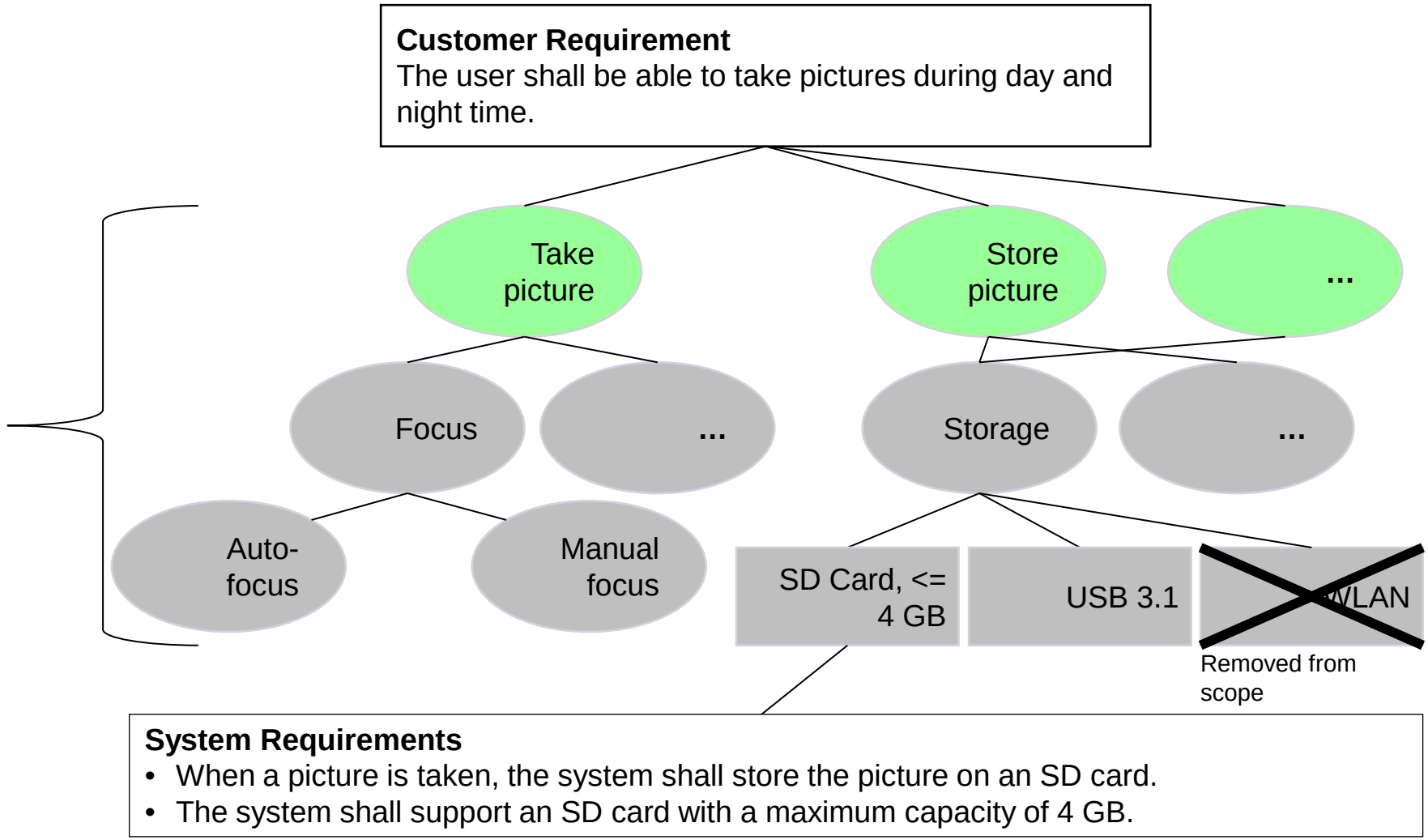
3. Define **necessary implementation constraints**,
   introduced by stakeholder requirements or unavoidable solution limitations.

4. Define **quality attributes** that explain “how good” the functional requirements shall work;
   use your domain specific quality tree or a standard like ISO/IEC 25010 as a starting point.

Based on ISO 29148

Iterative Refinement

Iterative Refinement



Inconsistencies lead to avoidable rework

Observed Problems

- Inconsistent vocabulary, E.g.,
  - picture vs. image
  - customize vs. modify
  - add vs. create

→ Define project-specific glossary

- Inconsistent sentence structure
  - The system shall take a picture.
  - The system shall be able to take a picture.
  - The system shall provide the capability to take a picture.

→ Define sentence template for requirements

Define objects, actions, and attributes

SIEMENS
Ingenuity for life

Objects

- Charger: A unit which recharges a system.
- Data: instance of a data type
- Display: A unit which visualizes data in different colors.
- Picture: group of data representing colors.
- Shutter Button: An multifunctional interaction element which triggers functions, e.g. for taking pictures or for setting the auto focus
- Trigger: a signal which releases a function.

Actions

- Capture: Save data in a RAM.
- Create: Generate data
- Delete: Erase data
- Display: Visualization of a picture to the user
- Edit: modify stored data
- Focus: Adjust the lens, so the picture is sharp
- Press: Interaction mode. Impact on an physical interaction element to trigger a signal. “Press” happens after “Touch”.
- Store: Save data in a memory.
- Touch: Interaction mode. Impact on an physical interaction element to trigger a signal. “Touch” happens before “Press”.

Mapping of objects to actions clarifies the scope

SIEMENS
Ingenuity for life

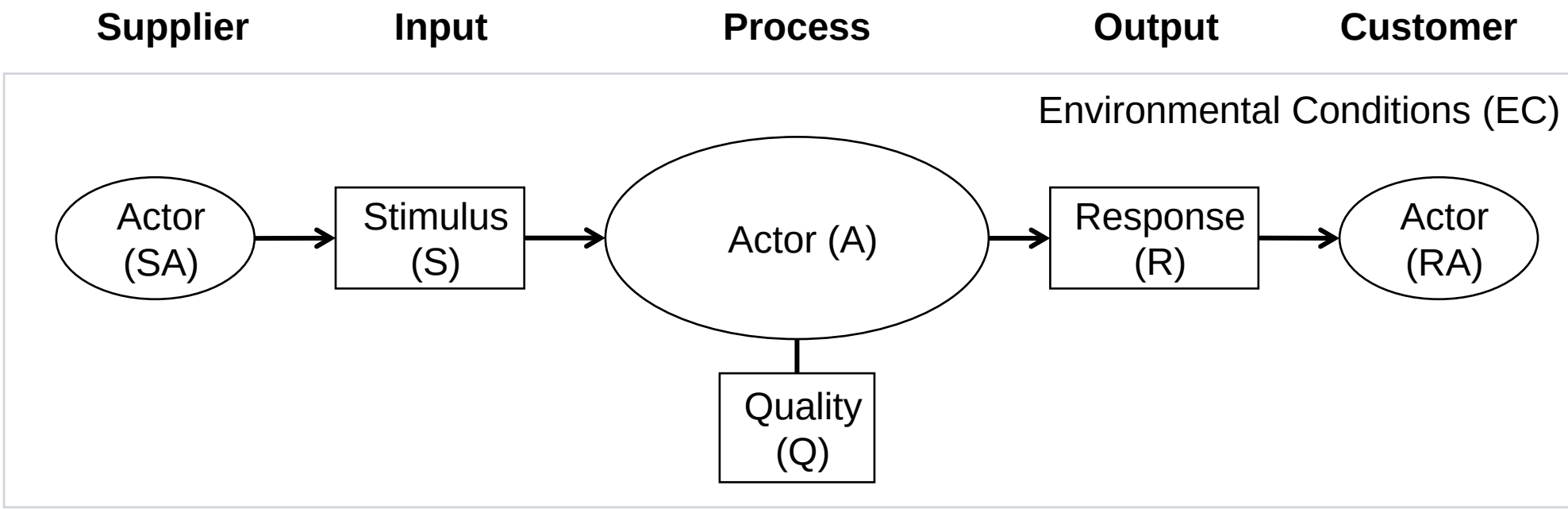
Action/Object Matrix

Objects	Actions												
	Capture	Create	Delete	Edit	Focus	Press	Receive	Release	Store	Take	Touch	Trigger	…
…													
Data		X	X	X					X				
Lens					X								
Picture	X		X	X					X	X			
Shutter Button						X		X			X		
Signal							X					X	
…													

How to read the table: <action> <object>;
Examples: take picture, edit picture, trigger signal, touch shutter button.

A system requirement statement is based on a stimulus/response model

Stimulus-Response Model



SA Actor which provides stimulus
S Stimulus
A Actor
Q Quality

R Response
RA Actor which receives response
EC Environmental Condition

A sentence template provides guidance for a consistent syntax (I)

SIEMENS
Ingenuity for life

Sentence Template for System Requirements:

<Condition>	Environmental or system (mode) conditions
<Stimulus>	Who (actor) or what (trigger) initiates the described capability
<Actor>	Provides a capability (if actor is system) or requests a capability (if actor is a user role)
Shall	Identifies a sentence as a requirement
<Action>	Specifies a capability
<Response>	An observable and testable result (object) and optionally an actor that receives the result
<Qualifier>	Quality attribute for a functional capability

Optional
Mandatory

Guidelines for different system requirements types

SIEMENS
Ingenuity for life

System requirement
- The actor is a system element, e.g. system, subsystem, component

Functional requirement
- The functional requirement should contain a stimulus actor and a stimulus as much as possible to describe which input is turned into an output.

Non-functional requirement
- It is recommended to formulate the condition under which a non-functional requirement should be met.

Interface requirement
- For an interface requirements which specifies an incoming item flow (e.g. material, information, energy): It is recommended to mention the actor which provides the item flow.
- For an interface requirements which specifies an outgoing flow (e.g. material, information, energy): It is recommended to mention the actor which receives the item flow.

Constraints
- Constraints leave less options for the system architect
- Constraints are often physical requirements.
- It is OK to mention the specific system element as the actor (instead of the system).

Example:
Deriving Functional System Requirements

SIEMENS
Ingenuity for life

Motivating customer requirement

The user shall be able to take pictures during day and night.

Derived system functional requirements (sample)

- When the system is in normal operation mode, the system is in single picture mode, and the user presses the shutter button, the system shall capture and store a single picture.
- When the ambient light is “low” and the system is in the automatic flash mode, the system shall automatically use flash.
- When the system is in normand the user touches the shutter button, the system shall automatically set the focus.
- The system shall store images in RAW, JPEG or DNG format.
- al operation mode, the system is in single picture mode,

Example:
Deriving Non-Functional System Requirements

SIEMENS
Ingenuity for life

Motivating customer requirement

The user shall be able to take pictures during day and night.

Derived system non-functional requirements (sample)

- When the environmental conditions are normal, the system is at least “very low” charged, the system is in operation mode, and after the user presses the shutter button, the system shall capture and store a picture in less then 1.0 sec.
- When the environmental conditions are normal, the system is at least “very low” charged, the system is in operation mode and after the user presses the shutter button, the system shall have a between shot speed of less then 0.5 sec to capture and store a picture.
- The system shall store at least 1,000 pictures at highest specified resolution and highest specified color depth.

Example:
Deriving Interface System Requirements

SIEMENS
Ingenuity for life

Motivating customer requirement

The user shall be able to take pictures during day and night.

Derived system interface requirements (sample)

- The system shall have a “Nikon 1” mount for attaching lenses.
- The system shall support the following apertures: f/2.8–f/5.6, f/5.6–f/11, and f/11–f/22.
- The system shall provide a USB 3.1 standard.
- The system shall have an interface to an external flash.

Example:
Deriving Constraints

SIEMENS
Ingenuity for life

Motivating customer requirement

The user shall be able to take pictures during day and night.

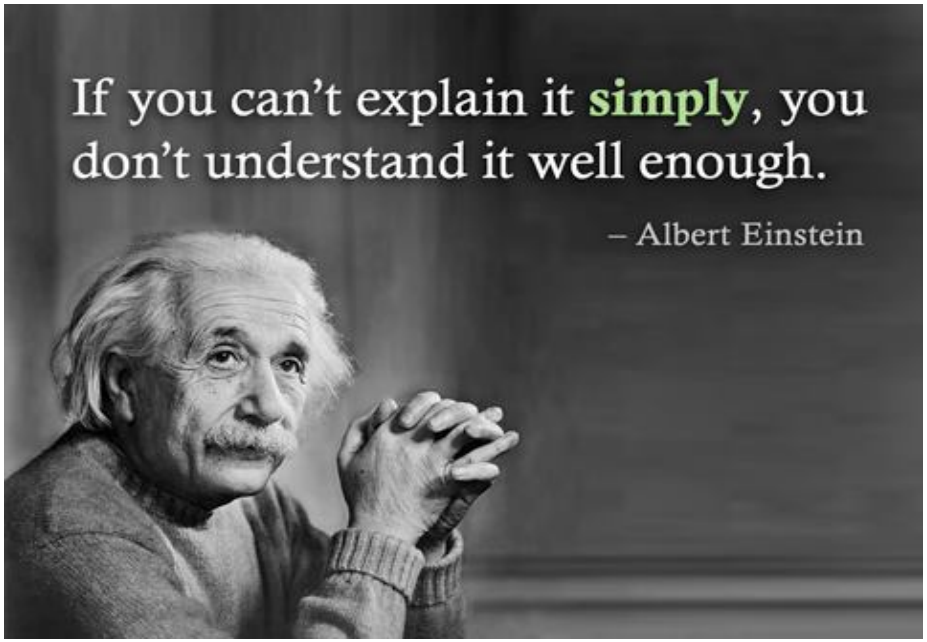
Derived system constraints (sample)

- The system shall weigh less then 5 oz.
- The system shall have dimensions less then 5 cm height, 2 cm depth and 8 cm width.
- The system chassis shall be made of aluminum.

Write Technical Requirements

Summary

- A requirement should describe a capability (turn an input into an output), not a system element.
- A glossary and a sentence template provides guidance for a consistent syntax and wording.



SIEMENS
Ingenuity for life

Agenda

Introduction

Writing Good Requirements

Characteristics of good requirements

Reviewing Requirements

Summary

Characteristics of good Requirements according to IEEE 29148

SIEMENS
Ingenuity for life

For individual requirements:

- Necessary
- Implementation free
- Unambiguous
- Singular
- Feasible
- Verifiable
- Traceable
- Consistent
- Complete

For a requirements set

- Complete
- Consistent
- Affordable
- Bounded

Mapping of IEEE 29148 characteristics to Requirement Pyramid

SIEMENS
Ingenuity for life

Quality Criteria	Business Goals	Customer Requirements	System Requirements
Individual Requirements			
Necessary	Mandatory	Mandatory	Mandatory
Implementation free	Mandatory	Mandatory	Mandatory
Unambiguous	Optional	Mandatory	Mandatory
Consistent	Mandatory	Mandatory	Mandatory
Complete	Optional	Optional	Mandatory
Singular	Mandatory	Optional	Mandatory
Feasible	Optional	Optional	Mandatory
Verifiable	Optional	Mandatory	Mandatory
Traceable	Mandatory	Mandatory	Mandatory
Set of Requirements			
Complete	Optional	Optional	Mandatory
Consistent	Mandatory	Mandatory	Mandatory
Affordable	Optional	Optional	Mandatory
Bounded	Optional	Optional	Mandatory

IEEE 29148 Characteristics of good Requirements "Necessary"

SIEMENS
Ingenuity for life

Necessary

A requirement is necessary, if it defines an essential capability, characteristic, constraint, and/or quality factor.

Source: Request
E.g.: The camera should take focused pictures automatically.

✗ **Example**
The system shall provide a camera bag.

✓ **Example**
When the system is in normal operation single picture mode, and the user touches the shutter button, the system shall set the focus automatically.

IEEE 29148 Characteristics of good Requirements "Implementation free"

SIEMENS
Ingenuity for life

Implementation free

The requirement states what is required, but not how the requirement should be met.

The requirement avoids placing unnecessary constraints on the architectural design, while addressing what is necessary and sufficient in the system.

✗ **Example**
The system shall allow the user to change the current shooting mode of operation by performing a two-finger swipe on the touch screen.

✓ **Example**
The system shall allow the user to select the current shooting mode of operation.

IEEE 29148 Characteristics of good Requirements "Unambiguous"

SIEMENS
Ingenuity for life

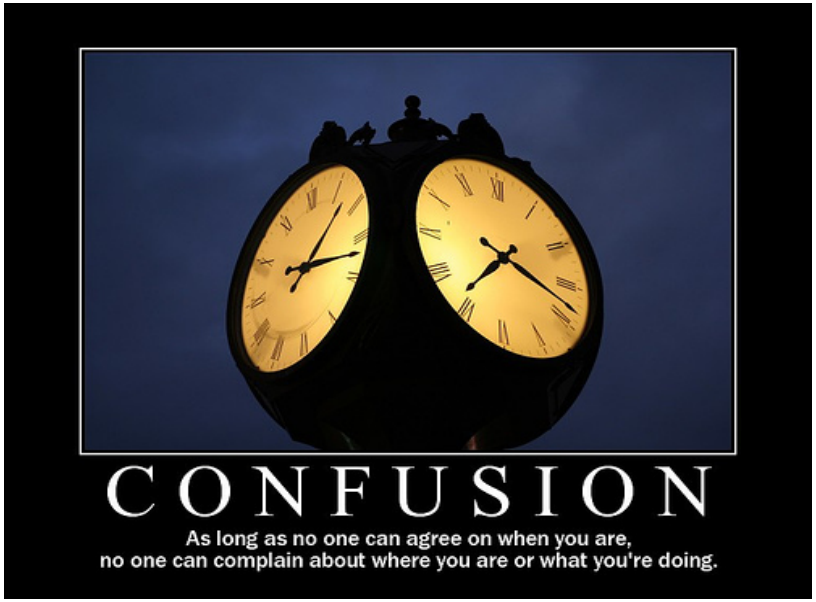
Unambiguous

The requirement is stated in a way that allows only one interpretation.

It should be stated simple and be easy to understand.

× **Example**
The system shall recharge the battery in a short time.

✓ **Example**
When the system is attached to a power outlet, and power is provided to the system, the system shall fully recharge in less than two hours.



Causes for ambiguity
common pitfalls in writing requirements I

SIEMENS
Ingenuity for life

Pitfall	such as …	example
Superlatives	'best', 'most', 'lowest', 'fastest'	The system shall have the lowest weight.
Subjective language	'user friendly', 'easy to use', 'cost effective'	The system shall be user friendly.
Loopholes	'if possible', 'as appropriate', 'as applicable'	The weight of the system shall not exceed 300g, if possible.
Ambiguous adverbs and adjectives	'almost always', 'significant', 'minimal'	The system shall have minimal down time.
Universal Quantifiers	'all', 'never', 'always', 'none', plural form	The system shall never fail.
Anonymous Antecedents	'he', 'she', 'it', 'this'	When a picture is taken, the system shall display the picture. This shall be done in 100ms or less.
Confusing Conjunctions	'and', 'or', 'but', 'now', 'so', 'for', 'yet'	The system shall have redeye reduction and 3x optical zoom or image stabilization

Causes for ambiguity
common pitfalls in writing requirements II

SIEMENS
Ingenuity for life

Pitfall	such as …	example
Comparative phrases	'better than', 'higher quality'	Images taken shall have higher quality than that of our competitors.
Unnecessary or Misplaced Modifiers	‘only’, ‘also’	‘The camera chassis shall only be red.’ vs. ‘Only the camera chassis shall be red.’
Negative statements	statements of system capability not to be provided	The system shall not be powered by AAA batteries.
Passive voice often hides the executer		‘A shutter sound shall be played.’ {Who does that?}
Poor sentence structure, confusing the meaning		When the camera is powered off, picture storage will be available for use.
Open-ended, non-verifiable terms	'provide support', 'but not limited to', 'as a minimum'	The camera shall as a minimum support SD Cards.

Causes for ambiguity
common pitfalls in writing requirements V

SIEMENS
Ingenuity for life

Pitfall	such as …	example
Verbose, failing to be concise		‘The Modules shall support the use of …’ vs. ‘The Modules shall use …’
Inconsistent vocabulary	using miscellaneous terms for the same meaning or using the same term with different meanings	‘picture’ vs. ‘image’, ‘add’ vs. ‘create’, ‘customize’ vs. ‘modify’
Incomplete references	not specifying the reference with its date and version number; not specifying just the applicable parts of the reference to restrict verification work	The camera shall support an USB interface.
Lazy Abbreviations	‘etc.’, ‘TBD.’	The system shall have multiple automatic shooting modes: portrait, landscape, night, etc.
Lazy Punctuation	slash ‘/’, parentheses ‘()’, ‘[]’	The system shall allow the user to take a picture and/or take a picture continuously.

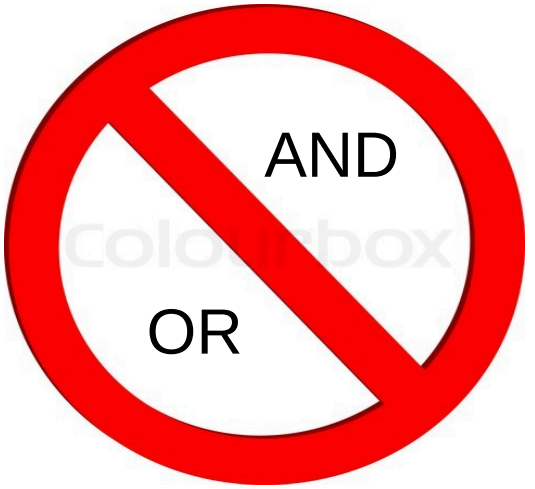
IEEE 29148 Characteristics of good Requirements "Singular"

SIEMENS
Ingenuity for life

Singular

The requirement statement includes only one requirement without use of conjunctions.

✗ **Example**
*When the system is in normal operation single picture mode, and the user touches the shutter button, the system shall set the focus automatically **and** the system shall capture **and** store a single picture*



✓ **Example**
When the system is in normal operation single picture mode, and the user touches the shutter button, the system shall set the focus automatically.

IEEE 29148 Characteristics of good Requirements "Feasible"

SIEMENS
Ingenuity for life

Feasible

The requirement is technically achievable, does not require major technology advances, and fits within system constraints (schedule, cost, technical, legal, regulatory, ...) with acceptable risk.

✗ **Example**
The system shall have 100x optical zoom.

✓ **Example**
The system shall have at least 20x optical zoom.

IEEE 29148 Characteristics of good Requirements "Verifiable"

SIEMENS
Ingenuity for life

Verifiable

The requirement includes explicit or inherited unambiguous measurable acceptance criteria.

✗ **Example**
The camera shall be reliable.

✓ **Example**
The camera shall fail at most once while taking 10,000 pictures.

IEEE 29148 Characteristics of good Requirements "Traceable"

SIEMENS
Ingenuity for life

Traceable

Requirement documentation provides (bi-directional) trace links, between:

- Requirement and its source; e.g.
  - Documented stakeholder need
  - Higher level requirements
  - Standards, regulations, guidelines
- Requirement and related test cases
- Requirement and detailing artefacts; e.g.
  - Lower level requirements
  - Design artefacts

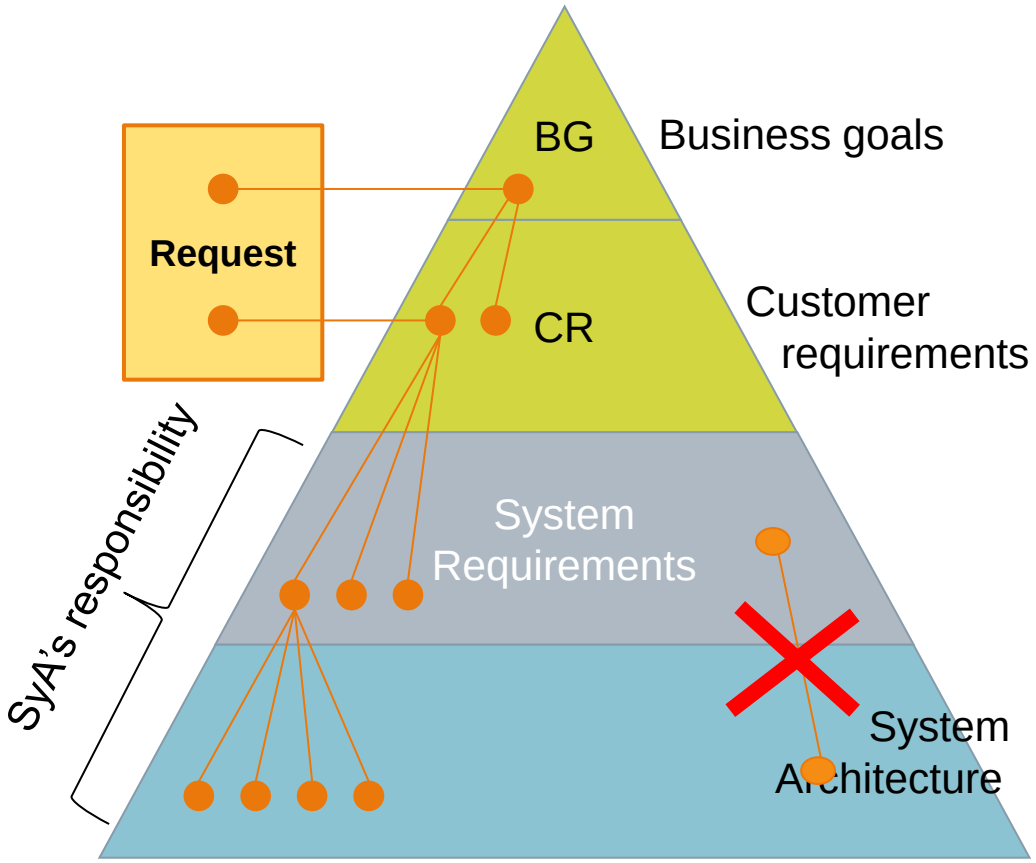
✗ Incorrect Trace Example	✓ Correct Trace Example
Customer Request	**Customer Request**
The camera should make high quality pictures which I like to see afterwards.	*The camera should make high quality pictures which I like to see afterwards.*
↑	↑
Customer Requirement	**Customer Requirement**
The user shall be able to take 5,000 pictures without recharging.	*The camera shall allow users to take pictures with >=15 megapixels resolution.*

What is requirements traceability?

Requirement traceability means:

Requirement documentation provides (bi-directional) trace links, between a requirement and ...

- its source; e.g.
  - documented stakeholder need
  - higher level requirements
  - standards, regulations, guidelines
- related test cases
- detailing artefacts; e.g.
  - lower level requirements
  - design artefacts
  - documentations
- risk analysis results



Usage of Tracing

Type of Analysis	Description	Processes supported
Impact Analysis	Following incoming links, in answer to the questions: “What if this were to change?”	Change management
Derivation Analysis	Following outgoing links, in answer to the question: “Why is this here?”	Cost-benefit analysis
Coverage Analysis	Counting statements that have links, in answer to the question: “Have I covered everything?” Most often used as a measure of progress.	General engineering, management reporting

IEEE 29148 Characteristics of good Requirements "Complete" (single requirement)

SIEMENS
Ingenuity for life

Complete

- The stated requirement needs no further amplification because it is measurable and sufficiently describes the capability and characteristics to meet the stakeholder's need

Source: Request

I want to take at least 5,000 pictures without recharging the battery.

× **Example**

The system shall take many pictures.

✓ **Example**

If the system is charged 98% or higher, the system shall take at least 5,000 pictures without recharging the battery.

IEEE 29148 Characteristics of good Requirements "Complete" (set of requirements)

SIEMENS
Ingenuity for life

Complete

- The set of requirements covers needs of all known internal and external stakeholders, incl. Quality Attributes.

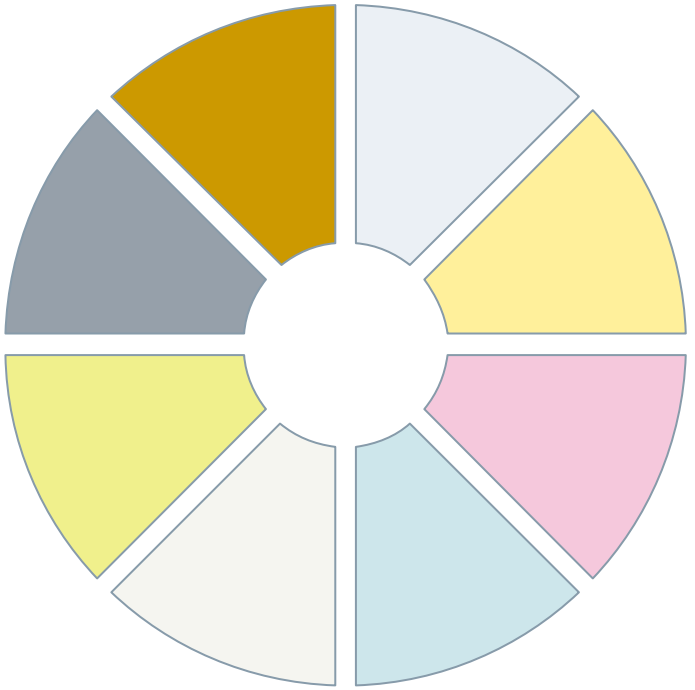
External stakeholders

Besides users and operators also regulation and standardization bodies and society.

Internal stakeholders

E.g. Product and Senior Management, R&D, Production, Engineering, Commissioning, and Service.

Categorizing and structuring, as well as (pre-filled) templates for requirement sets and lists of domain typical Quality Attributes help to gain completeness.



IEEE 29148 Characteristics of good Requirements "Consistent"

SIEMENS
Ingenuity for life

Consistent

The set of requirements does not include contradictory requirements.

Requirements are not duplicated.

The same term is used for the same item in all requirements.

✗ **Example**

*The system shall allow the user to browse the **pictures** stored on the memory card.*

*When an **image** is taken, the system shall display the **image**.*

Example

*The system shall allow the user to browse the **pictures** stored on the memory card.*

*When a **picture** is taken, the system shall display the **picture**.*

✗ **Example**

- *The system shall allow the user to set the flash on or off.*
  The system shall trigger the flash when taking a picture in the fully automatic mode.

✓ **Example**

When the system is not in fully automatic mode, the system shall allow the user to set the flash on or off.

IEEE 29148 Characteristics of good Requirements "Affordable"

SIEMENS
Ingenuity for life

Affordable

The complete set of requirements can be satisfied by a solution that is obtainable/feasible within lifecycle constraints (cost, schedule, technical, legal, regulatory, ...).

Source: Request
The camera should have a price below $300.

✗ **Example**
The system shall have a chassis made of gold.

✓ **Example**
The system shall have a chassis made of aluminum.

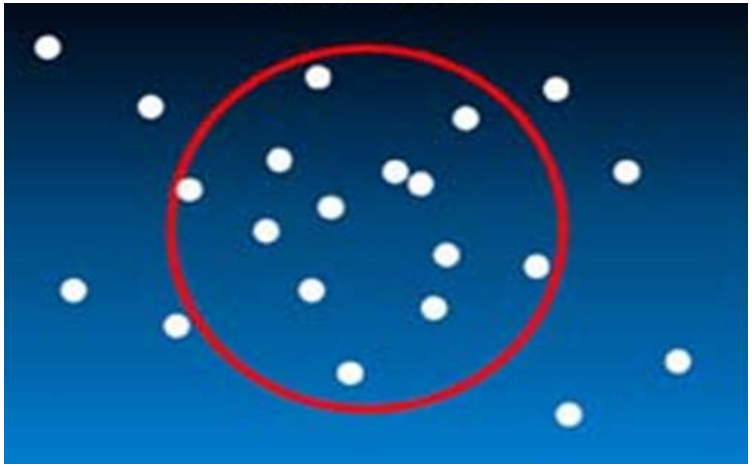
IEEE 29148 Characteristics of good Requirements "Bounded"

SIEMENS
Ingenuity for life

Bounded

The set of requirements maintains the identified scope for the intended solution without increasing beyond what is needed to satisfy user needs.

Source: Request
The camera should allow emailing of pictures.



✗ **Example**
The user shall be able to send e-mails.

✓ **Example**
The user shall be able to e-mail pictures taken to a specified e-mail address.

Agenda

Introduction

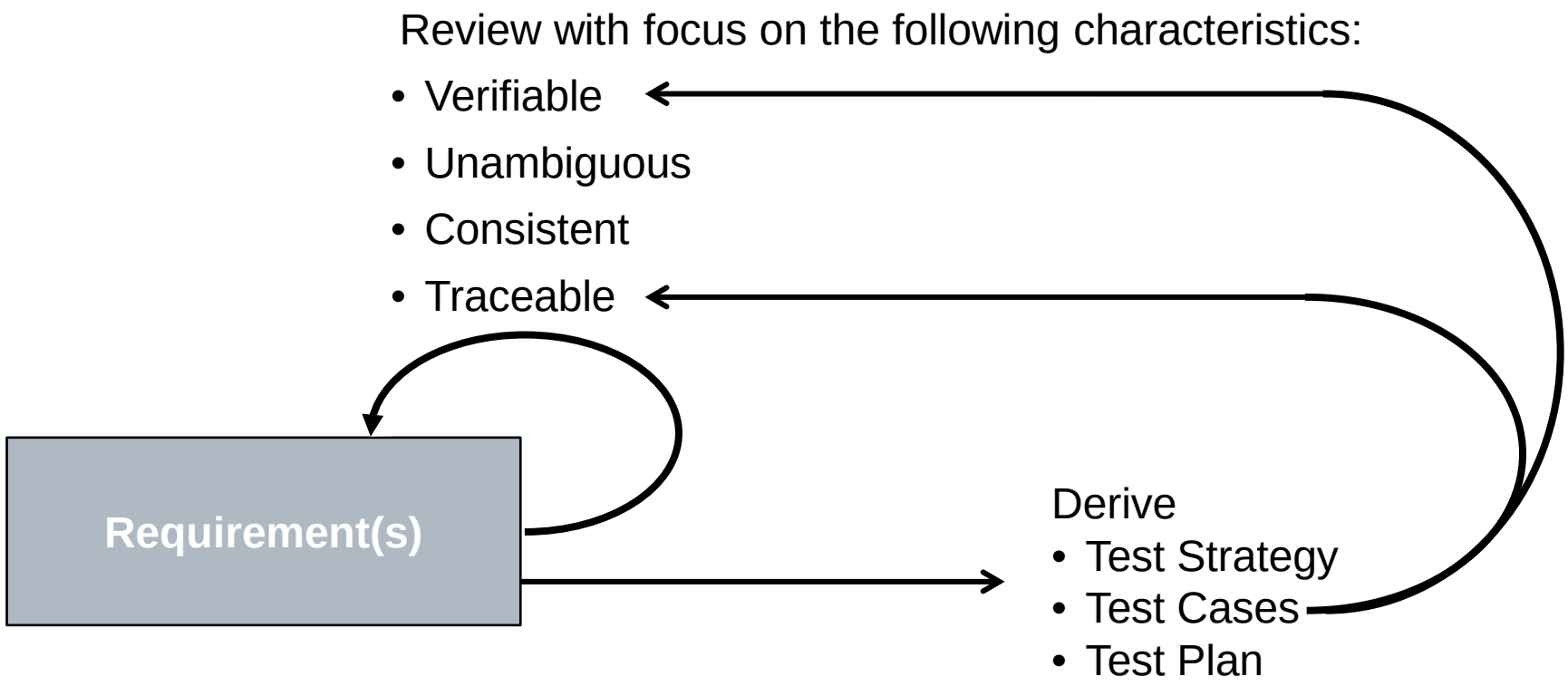
Writing Good Requirements

Characteristics of good requirements

Reviewing Requirements

Summary

Reviewing Requirements as a Test Architect and Test Manager



Mapping of IEEE 29148 characteristics to Roles in the context of Customer Requirements

SIEMENS
Ingenuity for life

Quality Criteria	Product Manager	System Architect	Project Manager	Test Architect & Test Manager	Quality Engineer
Single Requirements					
Well written					in focus
Necessary	in focus				
Implementation free		in focus			
Unambiguous	in focus			in focus	in focus
Consistent	in focus				in focus
Complete	n/a	n/a	n/a	n/a	n/a
Singular	n/a	n/a	n/a	n/a	n/a
Feasible	n/a	n/a	n/a	n/a	n/a
Verifiable				in focus	
Traceable		in focus			in focus
Set of Requirements					
Complete	n/a	n/a	n/a	n/a	n/a
Consistent		in focus			
Affordable	n/a	n/a	n/a	n/a	n/a
Bounded	n/a	n/a	n/a	n/a	n/a

Mapping of IEEE 29148 characteristics to Roles in the context of System Requirements

SIEMENS
Ingenuity for life

Quality Criteria	Product Manager	System Architect	Project Manager	Test Architect & Test Manager	Quality Engineer
Single Requirements					
Well written					in focus
Necessary	in focus				
Implementation free		in focus	in focus		
Unambiguous		in focus		in focus	in focus
Consistent		in focus		in focus	in focus
Complete		in focus			
Singular		in focus			
Feasible		in focus			
Verifiable				in focus	
Traceable		in focus		in focus	in focus
Set of Requirements					
Complete		in focus			
Consistent	in focus				
Affordable		in focus	in focus		
Bounded	in focus				

Reviewing Technical Requirements

- Requirements review require quality criteria.
  We suggest the requirements quality as described in ISO 29148

- Requirements quality criteria should be distributed to different roles;
  each role emphasizes different aspects.

- A good written review comment …
  … makes a reference to a specific requirements statement,
  … identifies the defect quality criterion, and
  … describes the problem.
  It can suggest a solution.

- Time and effort for reviews should not be underestimated
  and be allocated in the project plan.

Write a Good Review Comment

- A review comment refers to an individual requirements object
- A review comment states the type of defect against criteria as defined in the Requirements Management Plan
- A review comment describes the specific problem/deviation.
- A review comment is actionable
- Optionally, a review comment proposes how to repair a requirement

→ Receiving well written review comments can avoid rework.

Review Comment Sheet Example

SIEMENS
Ingenuity for life

Date	Rev	Req ID	Comment	Cat	Response	Status
03-Jun-11	HD	SyRS_1234	Not complete: Does not reflect aspect of “autofocus”	A	Agreed: Autofocus added as a separate requirement	Open
03-Jun-11	HD	SyRS_1245	Ambiguous: “user” not defined	B	Rejected: “user” is defined in the project glossary	Open

Agenda

- Introduction
- Writing Good Requirements
- Characteristics of good requirements
- Reviewing Requirements
- **Summary**

Purpose of High-Quality Requirements Specification

- **A high-quality requirements specification…**

  … is the documentation of all requirements describing **what a product needs to provide**,

  … is the basis for a **commercial contract** to build the

  … is the basis for deriving a **test specification**,

  … defines the product's requirements in a **measurabl**

  … is the **prerequisite for tracing** product requiremen

  … influences the **project plan** to build the product.

At Siemens the specified solution is usually a system

Test Architect's involvement in requirements engineering

SIEMENS
Ingenuity for life

Understand

- Do understand users' problems, potential risks, and cost drivers
- Do manage and understand change!

Analyze and assess

- Actively identify your architecturally relevant requirements
- Check the quality criteria of your requirements – don't overdo
- Analyze the testability of each requirement – ideally provide a test case

Contribute and drive

- Participate in the requirement analysis process, with focus on testability
- For test environment’s requirements: involve stakeholders to address their real needs and do it early

Guiding principle: **As simple as possible, but not simpler!**

Departing thought

The most difficult part of requirements gathering is not the act of recording what the user wants, it is the exploratory development activity of helping users figure out what they want.

[Steve Mc Connell]

Further readings

Use the SSA Wiki :
https://wiki.ct.siemens.de/x/fReTBQ

and check the “Reading recommendations”:
https://wiki.ct.siemens.de/x/-pRgBg

- Architect's Resources:
  - Competence related content
  - Technology related content
  - Design Essays
  - Collection of How-To articles
  - Tools and Templates
  - Reading recommendations
  - Job Profiles for architects
  - External Trainings
  - ... more resources