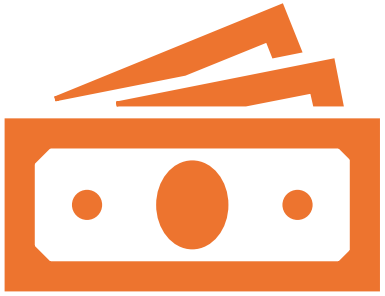
The pragmatic requirements for requirements

Hazel Woodcock, ESEP

hazel.woodcock@scsrailways.co.uk

Why

Common understanding

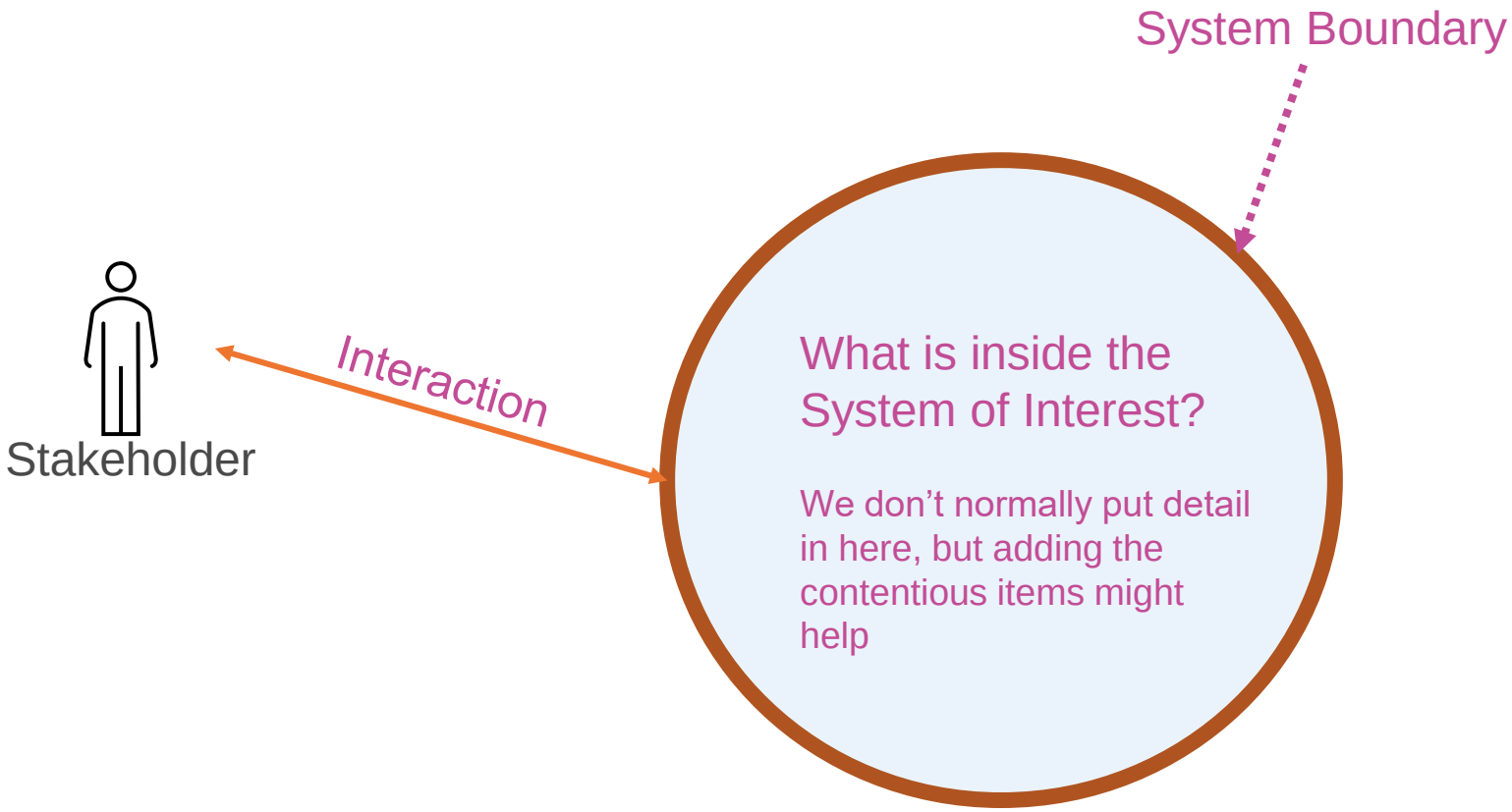


Get paid

Quality is Conformance to Requirements

– Phil Crosby, Quality is Free

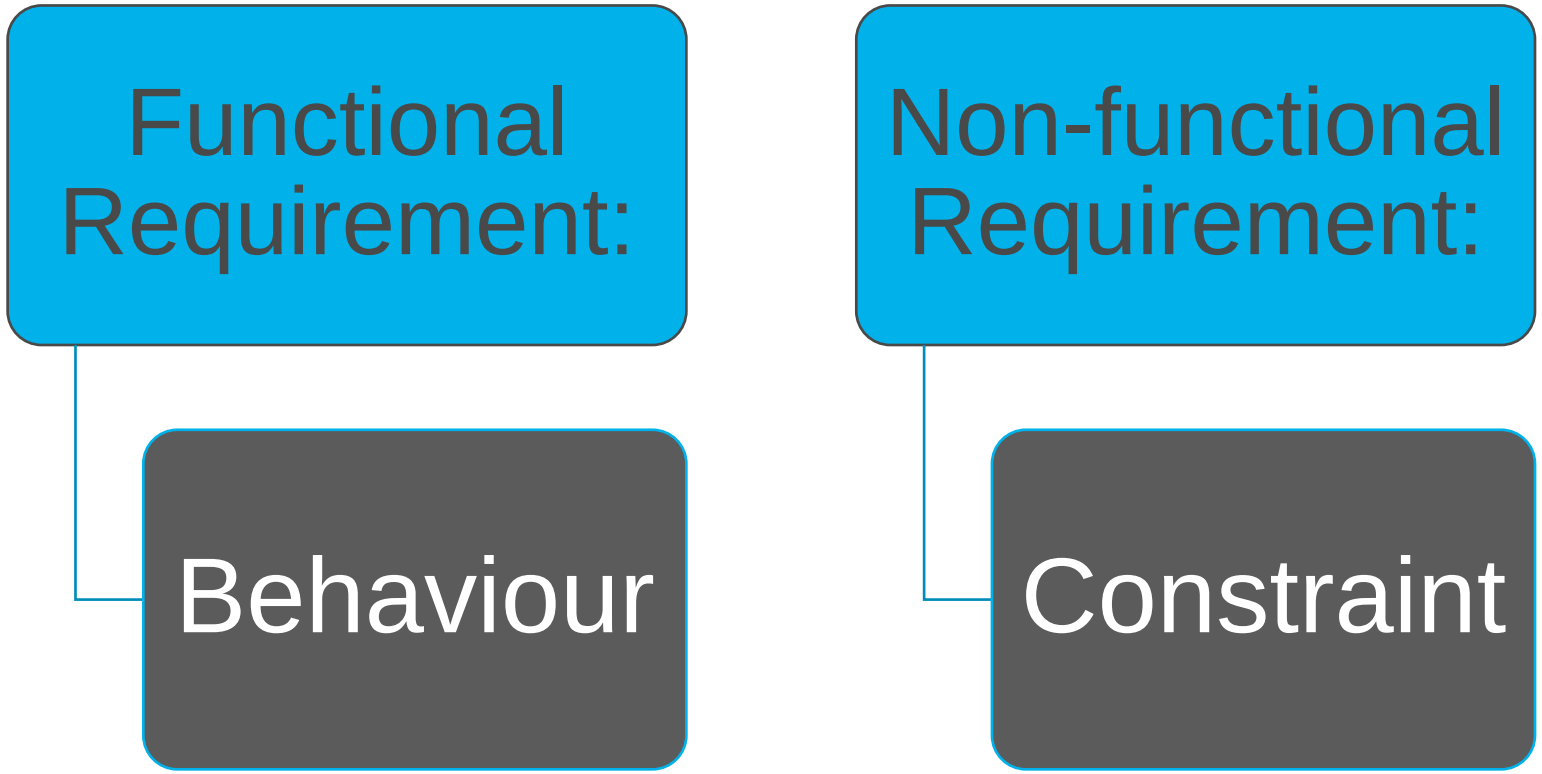
Q1: What is the context



What is outside the System of Interest?

Put the stakeholders here, including interfacing systems and note the key interactions / interfaces.

Types of Requirement: Functional/Non-functional



Types of Requirement: Stakeholder/System

Stakeholder

Problem domain

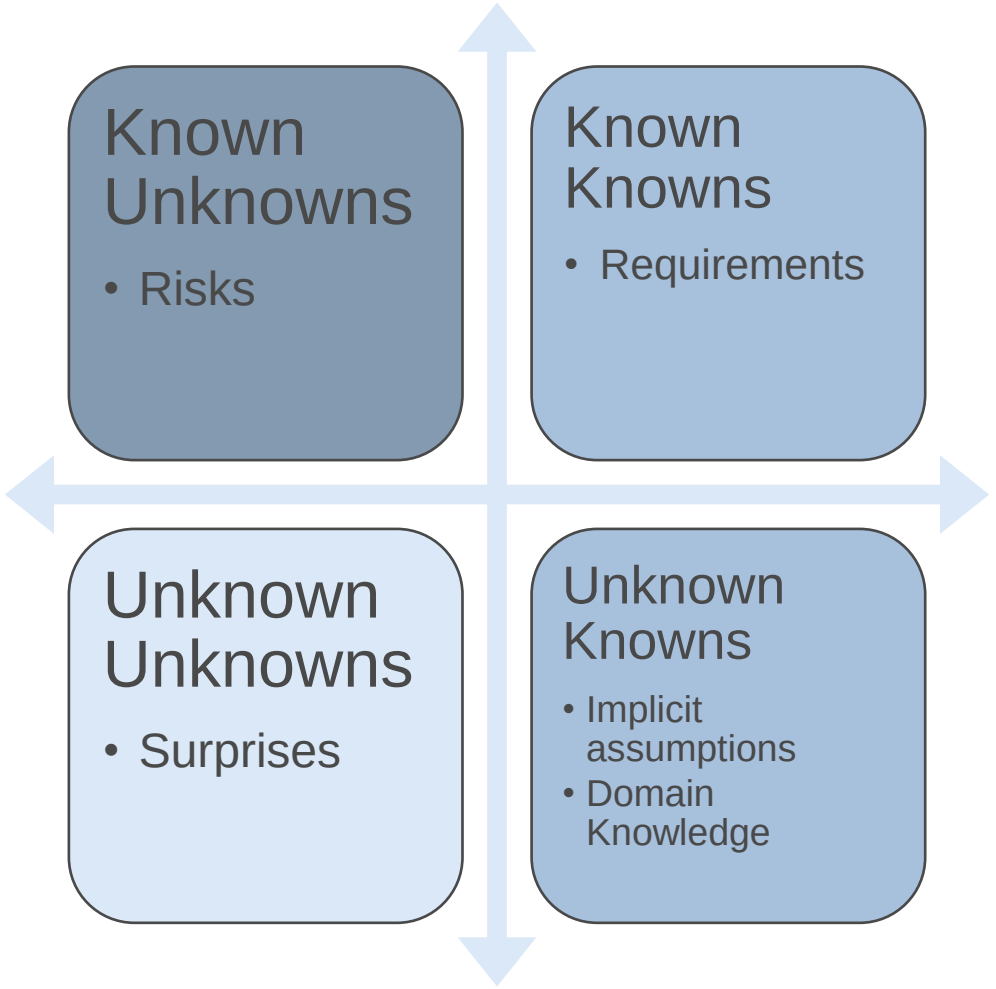
System

Solution domain

Rumsfeldian Analysis

*Reports that say that something hasn't happened are always interesting to me, because as we know, there are **known knowns**; there are things we know we know. We also know there are **known unknowns**; that is to say we know there are some things we do not know. But there are also **unknown unknowns**—the ones we don't know we don't know. And if one looks throughout the history of our country and other free countries, it is the latter category that tends to be the difficult ones.*

February 12, 2002



Track attributes to answer questions

Status

- Progress against plan
- Work remaining

Ownership

- Team or system

Verification method

- Is this verifiable?

Traceability through lifecycle

- Through to design and test

Managing Requirements

Elicitation

Validation

Decomposition

Change

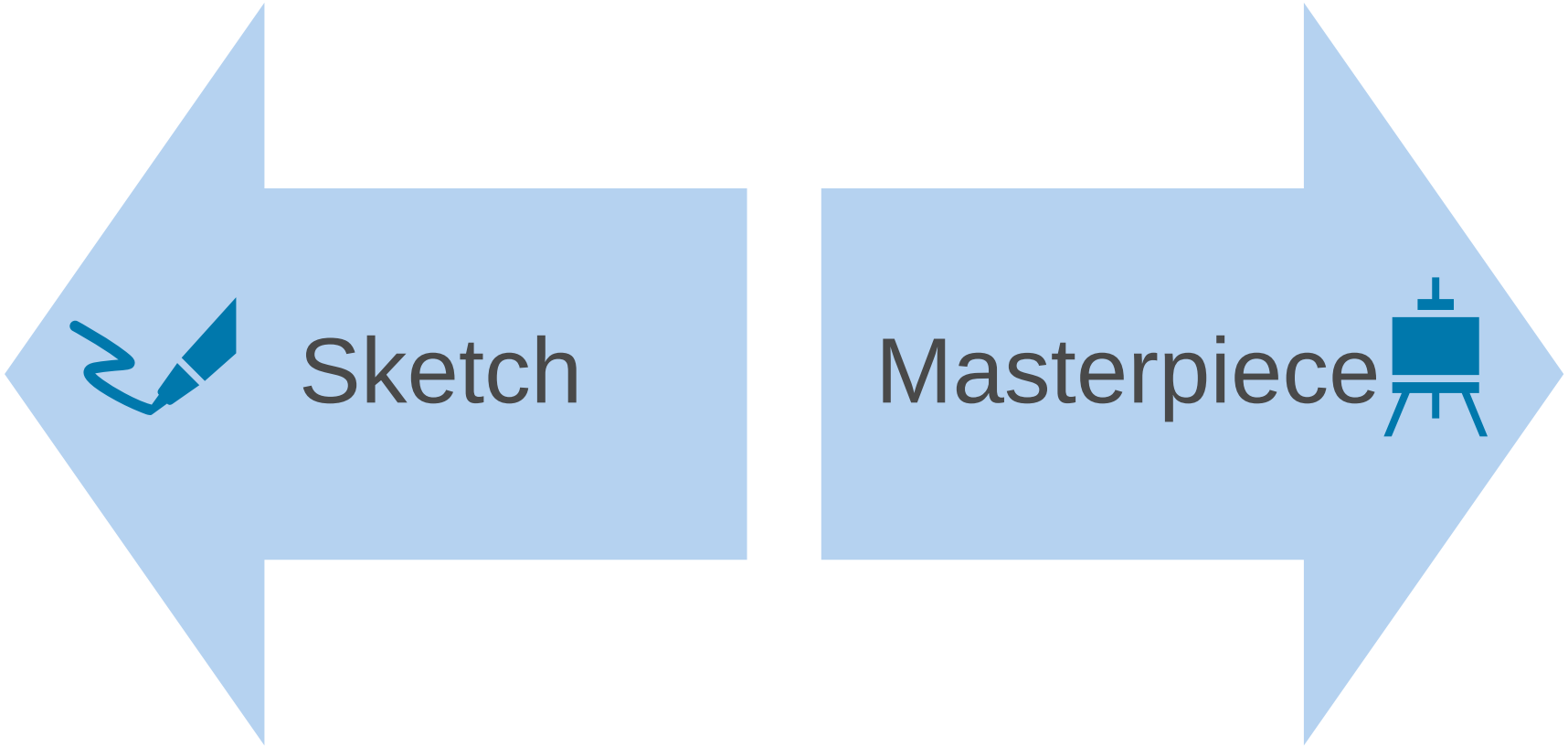
Traceability

Verification

Implementation

Formality of language

A complete and correct set of requirements



How do you know when you are done?

Lifecycle

Incremental

Iterative

Support with attributes

Tailor the formality

Enough effort at the right time – a pragmatic approach

Review

Single Requirements

Set of Requirements

Requirements Document

Language

MoSCoW

- Must
- Should
- Could
- Won’t

Shall / Will

- Shall – needs to be done
- Will – safe to expect it to be there

Language

Problem domain

- The <stakeholder> shall be able to <achieve a goal>
- Solution independent

Solution domain

- The <system> shall <do stuff>
- Implementation independent

Agile therefore Use Cases?

Use cases are a good SOURCE of requirements

Use Case Diagrams ARE NOT Use Cases

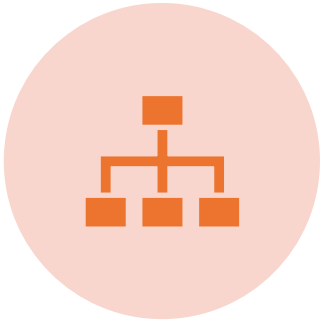
Verification, Validation, MBSE

Work with…

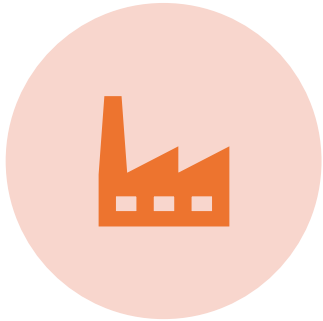
Trace to…

Report on…

Pragmatism



Know the ‘rules’

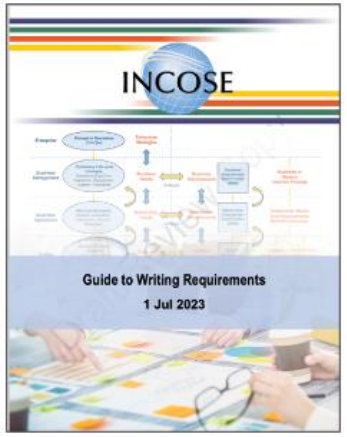


Tailor your process

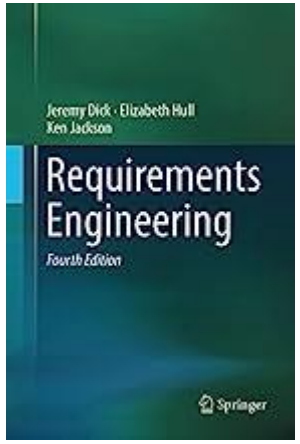
Focus on the basics

Find the *appropriate* level of formality

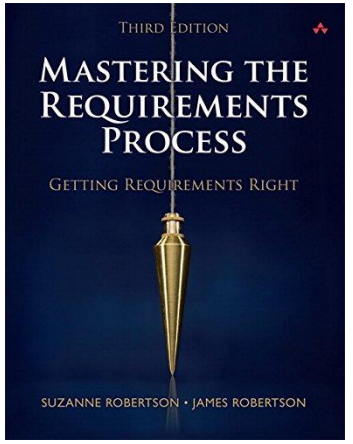
Resources



INCOSE Requirements Writing Guide



- **ISBN-10 :** 3319869973
- **ISBN-13 :** 978-3319869971
- Requirements Engineering
- by Jeremy Dick, Elizabeth Hull, Ken Jackson



- **ISBN-10 :** 0321815742
- **ISBN-13 :** 978-0321815743
- Mastering the Requirements Process: Getting Requirements Right
- by Suzanne Robertson, James Robertson