- Webinar rules:
  - You'll be muted all along the Webinar
  - There's a chatting box to ask your questions or send your comments when you want
  - Please address these comments and questions to the user "The REUSE Company" and not to the presenter directly
  - If you have any technical issue, please use this chatting box, or mail us at: support@reusecompany.com
  - The Webinar will be recorded. A link to the recording will be sent to you in few days

Optimization of the Requirements Engineering Process

Christer Fröling
CEO
REUSE Scandinavia
christer.froling@reusecompany.com

Cecilia Karlsson
Marketing & Communication
The REUSE Company
cecilia.karlsson@reusecompany.com

THE REUSE COMPANY

- Introduction to The REUSE Company and the speakers
- Knowledge driven Systems Engineering
- Work product quality
- SE tool's interoperability
- Traceability of work products
- Verification & Validation
- Q&A

Introduction to

TRC and the presenters

About The REUSE Company (TRC)

01 The company was created in **1999**

As a spin-off of a University in Europe

02 **Systems + Software Engineers**

Smart combination between Company staff and R&D from Academia

03 **Headquarters:**
Madrid (Spain)

International offices:
Stockholm (Sweden)
Tokyo (Japan) Delegation

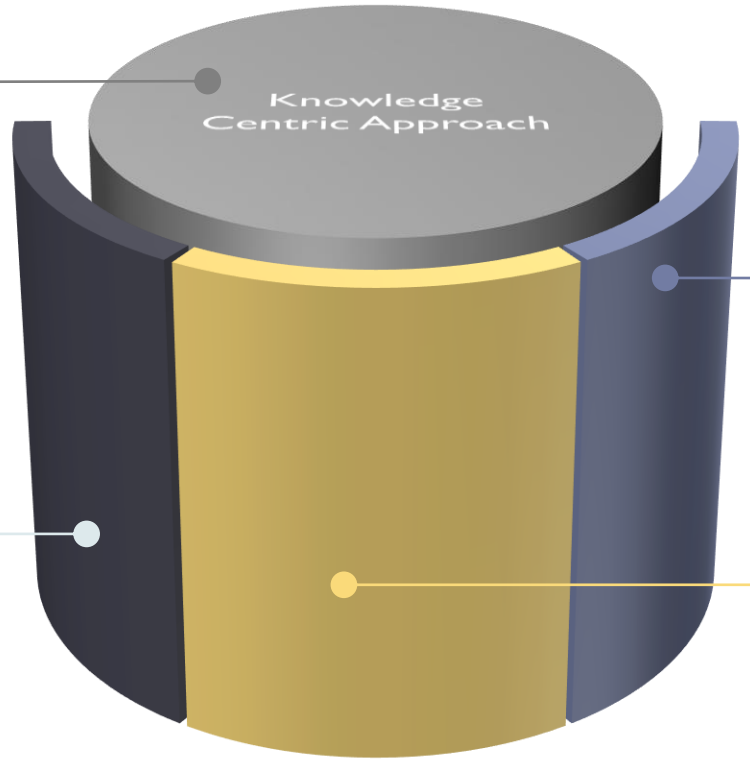
2021:
USA
Chicago/Detroit/Miami

04 To promote a **reusable**, **scalable** and global solution to a **smart** and **interoperable** Systems Engineering environment, by offering a **semantic knowledge centric** approach.

T(he) R(euse) C(ompany)

Knowledge Centric Systems Engineering (KCSE)

Connectivity & Interoperability –
Configuration Management –
Lifecycle Management –
Global Repositories –



QUALITY (ies)

- Quality Management
- Verification & Validation management
- Risks Management
- Smart Authoring

TRACEABILITY

Links and Interoperability –
Transformations –
Change management –

REUSABILITY

- Interoperability
- Retrieval & Archiving
- Adaptability to Existing Toolsets

The Presenter

Christer Fröling

christer.froling@reusecompany.com

+46 (0)72 232 24 63

@ReuseCompany

https://www.linkedin.com/in/christerfroling/

Christer Fröling is a Swedish citizen acting in the role of the **CEO for Reuse Company Scandinavia**. He has over **two decade of experience in** successful implementation of **Systems Engineering (SE)** and its sub-disciplines in a variety of roles and technical domains.

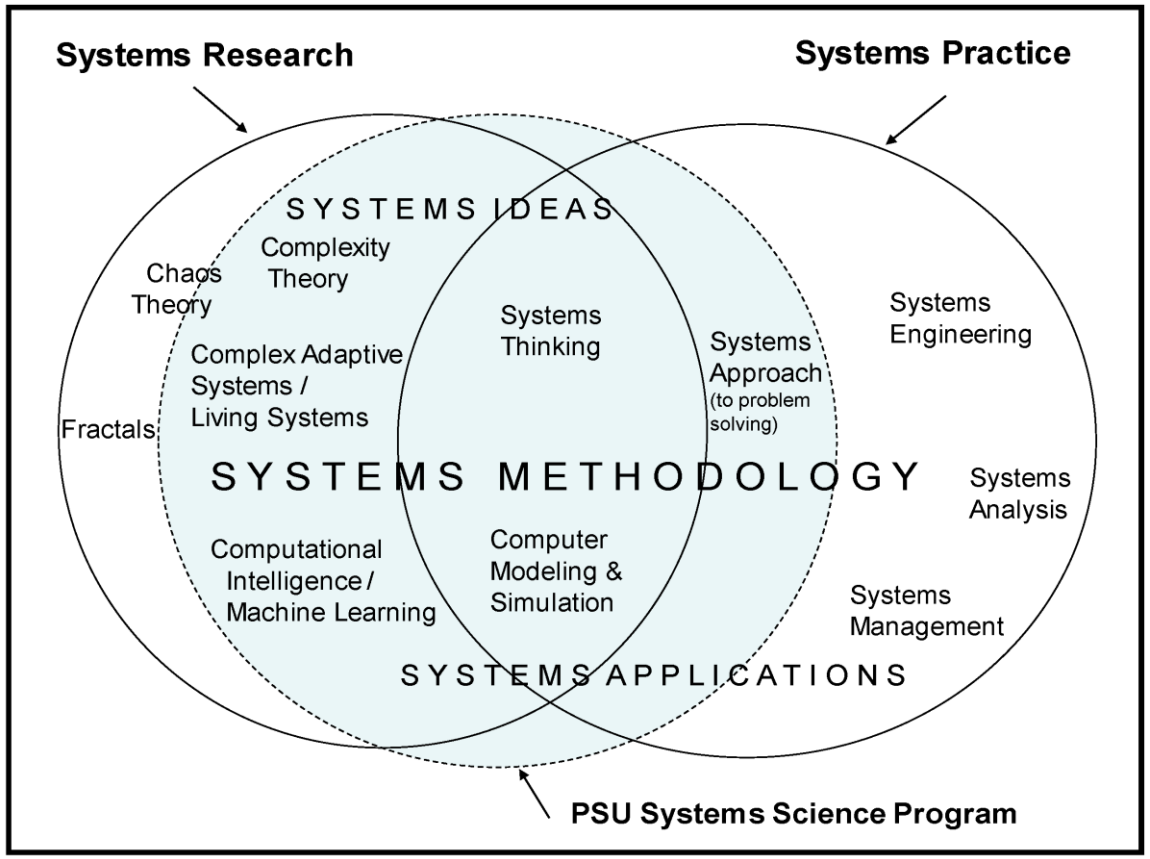
He has **experience** from both **developing advanced technical systems** as well as **helping public organizations** in the specification and **procurement** of complex infrastructure projects.

Christer specializes as a **principal consultant in applying SE and "design thinking"** into organizations willing to adopt change and implement a **knowledge driven** and **Lean SE approach** focusing on information quality, knowledge buildup and reuse with a passion of coaching others.

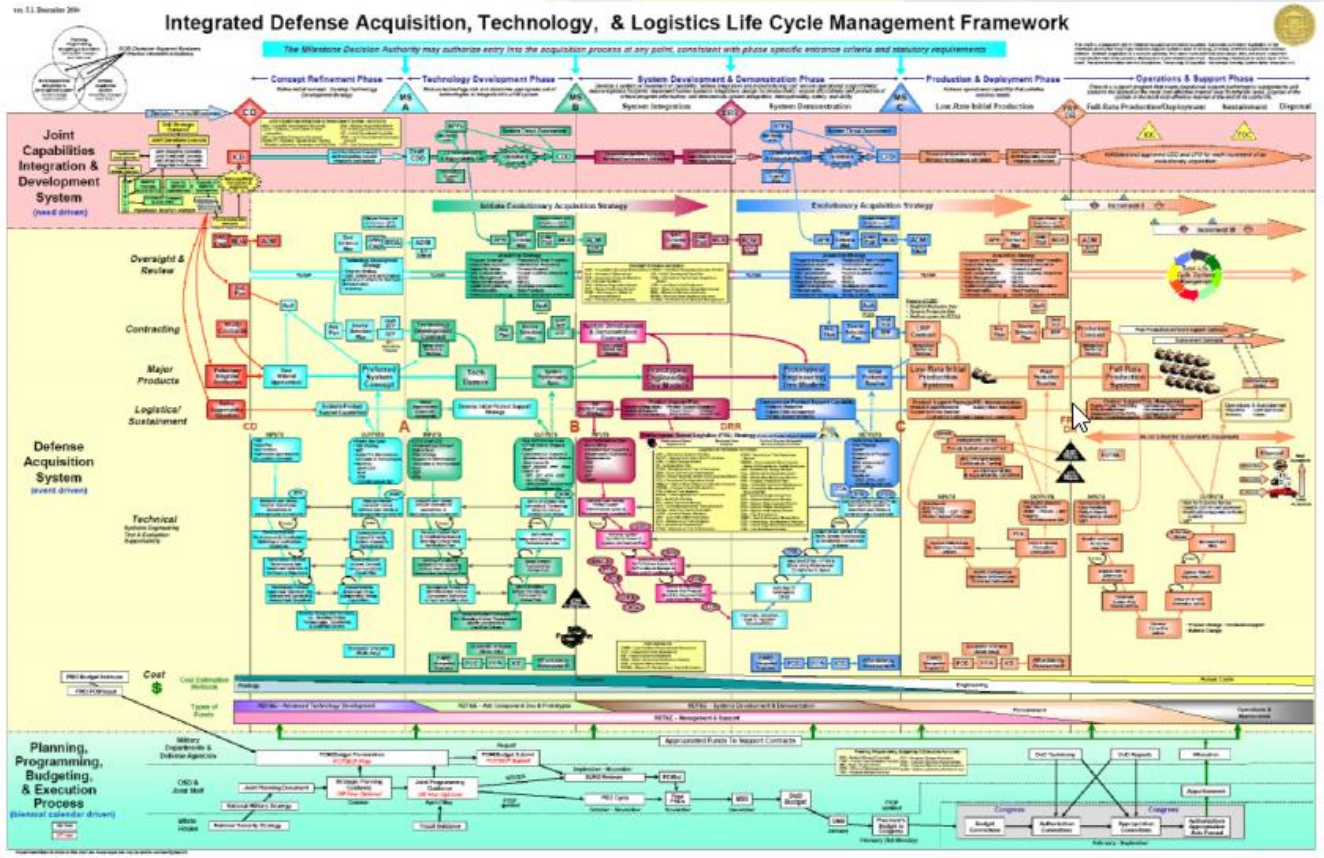
He is an **appreciated lecturer**, **teacher** and a strong **believer in knowledge sharing** and networking.

Knowledge driven Systems Engineering

SYSTEMS ENGINEERING – The way to manage Complexity, but…

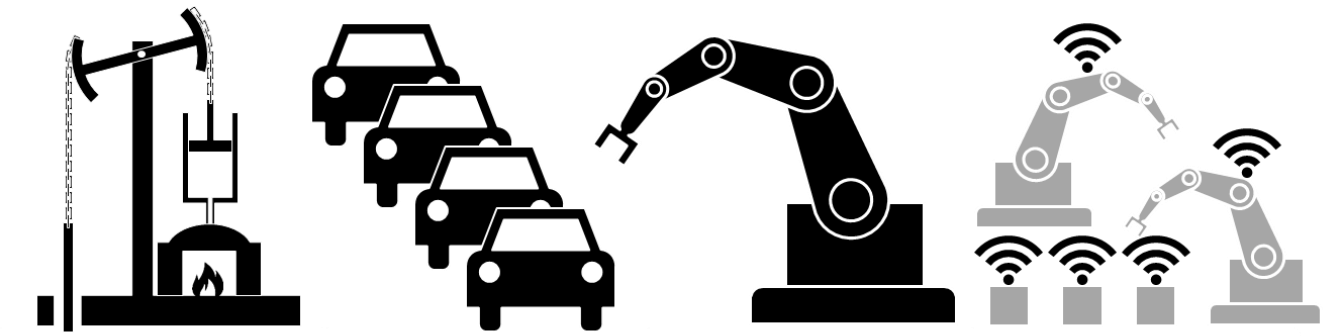


Source: https://www.sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)

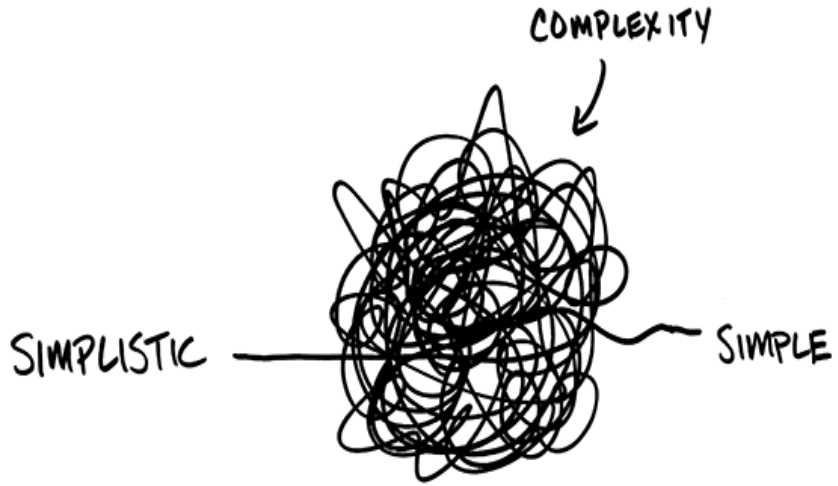


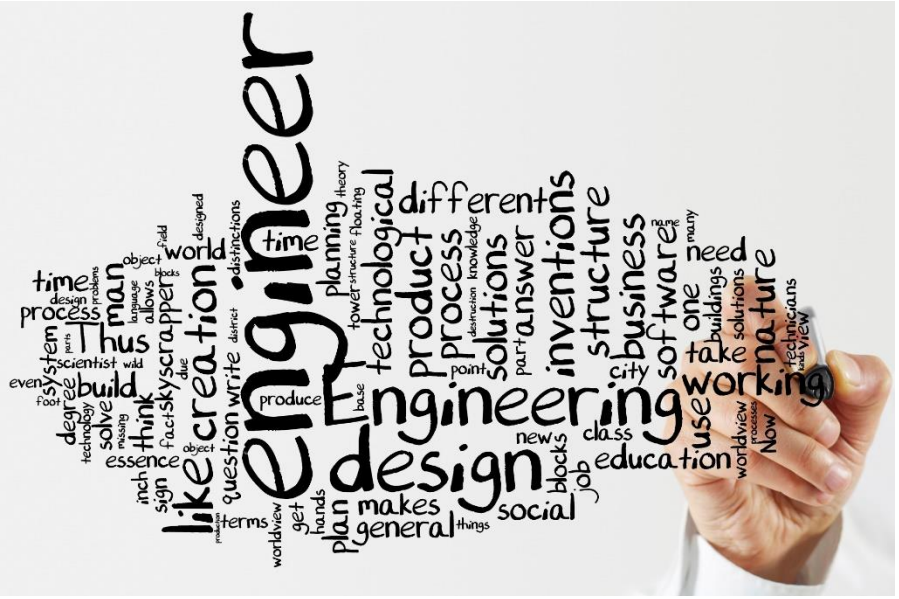
Source: NDIA 8th Annual Systems Engineering Conference, 2005

Industry 4.0 – But have the SE practices been left behind?



1st	2nd	3rd	4th
Mechanization, water power, steam power	Mass production, assembly line, electricity	Computer and automation	Cyber Physical Systems





Imperative: Be Agile, Learn and Apply Learning

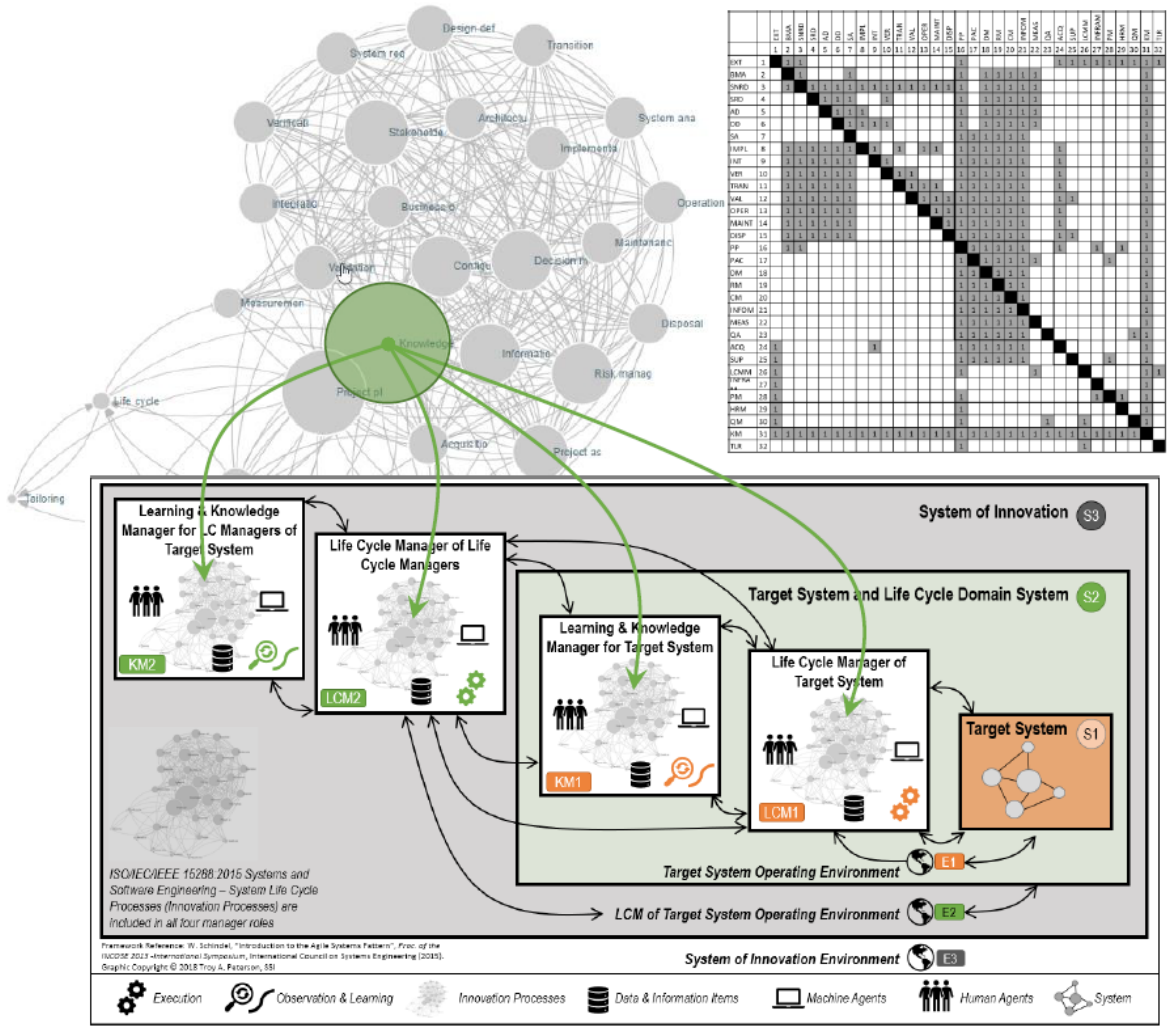
SSI

Agreement Processes
- Acquisition Process (6.1.1)
- Supply Process (6.1.2)

Organization Project-Enabling Processes
- Life Cycle Model Management Process (6.2.1)
- Infrastructure Management (6.2.2)
- Project Portfolio Management (6.2.3)
- Human Resource Management (6.2.4)
- Quality Management (6.2.5)
- Knowledge Management Process (6.2.6)

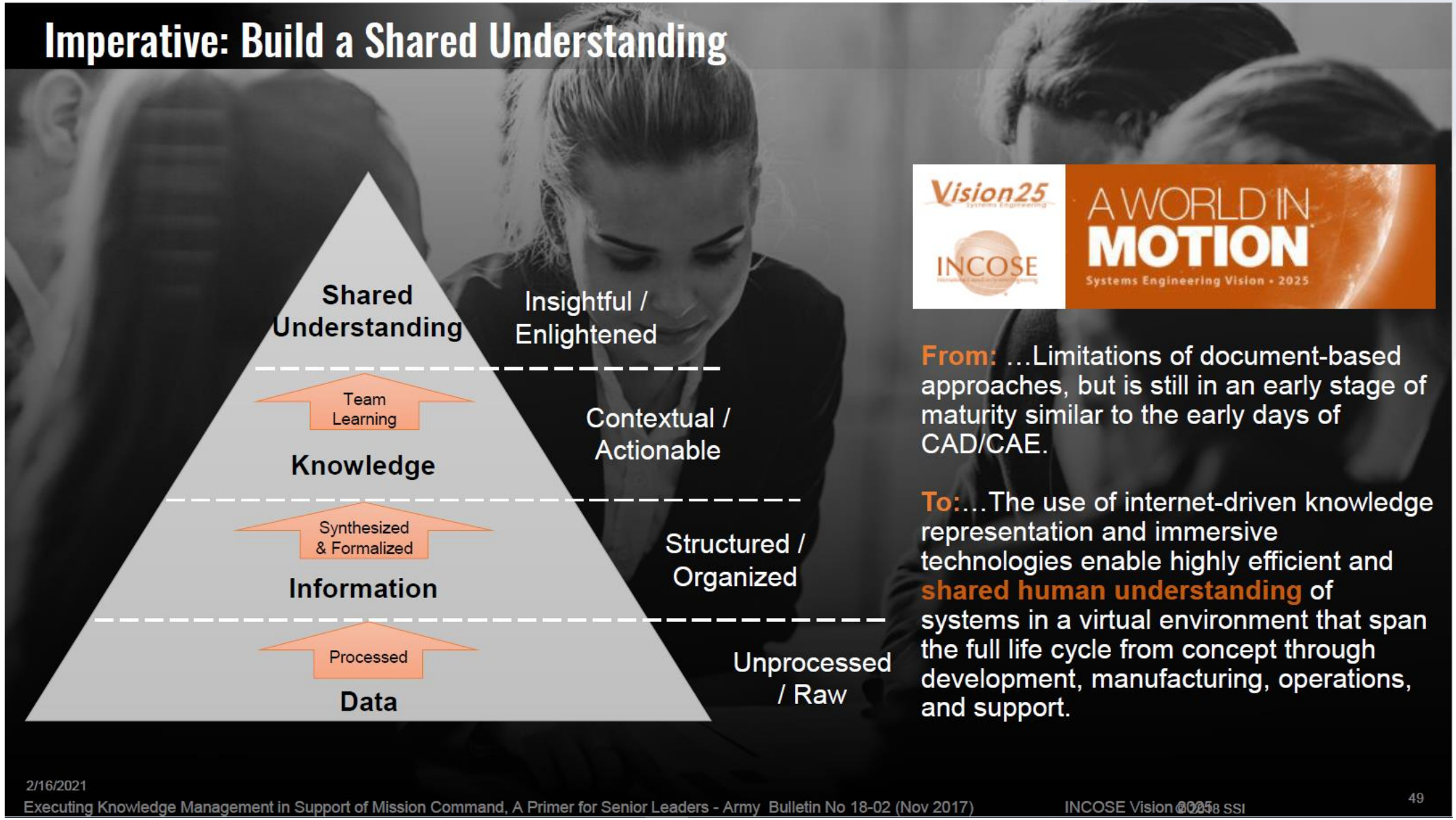
Technical Management Processes
- Project Planning Process (6.3.1)
- Project Assessment and Control Process (6.3.2)
- Decision Management Process (6.3.3)
- Risk Management Process (6.3.4)
- Configuration Management Process (6.3.5)
- Information Management Process (6.3.6)
- Measurement Process (6.3.7)
- Quality Process (6.3.8)

Technical Processes
- Business or Mission Analysis Process (6.4.1)
- Stakeholder Needs & Requirements Definitions Process (6.4.2)
- System Requirements Definition Process (6.4.3)
- Architecture Development Process (6.4.4)
- Design Definition Process (6.4.5)
- System Analysis Process (6.4.6)
- Implementation Process (6.4.7)
- Integration Process (6.4.8)
- Verification Process (6.4.9)
- Transition Process (6.4.10)
- Validation Process (6.4.11)
- Operation Process (6.4.12)
- Maintenance Process (6.4.13)
- Disposal Process (6.4.14)

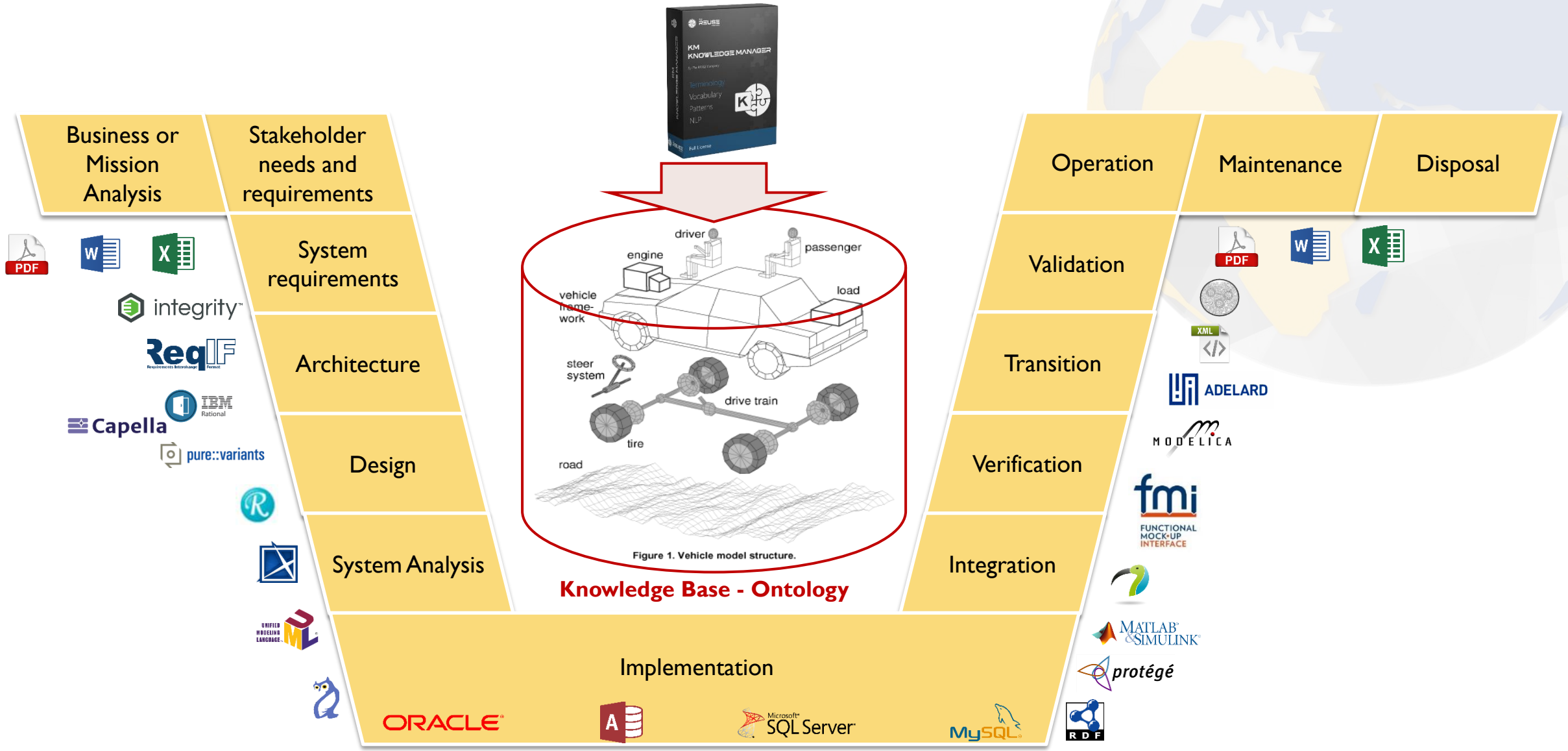


2/16/2021 ISO 15288 © 2021 SSI - UNCLASSIFIED 51

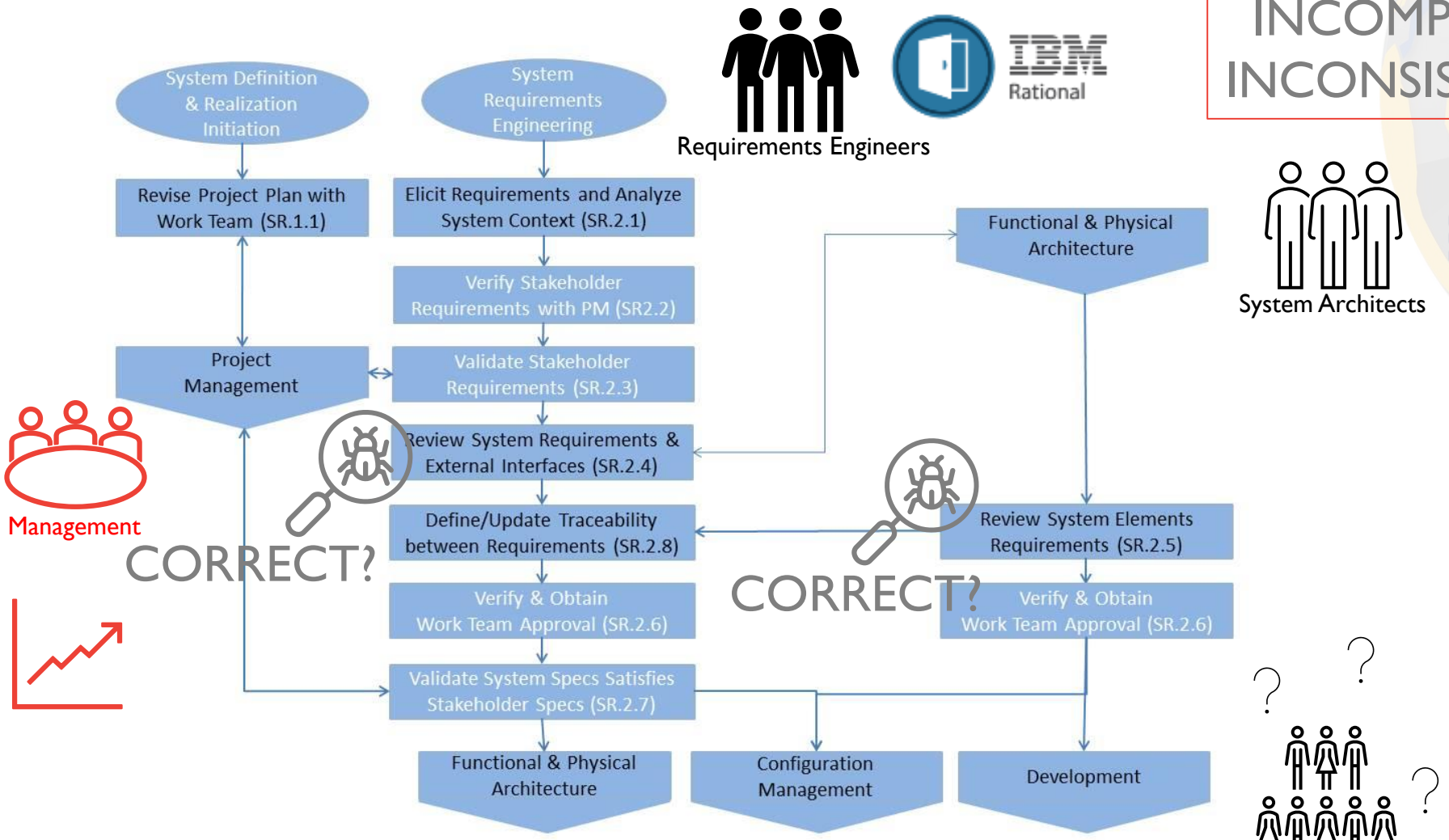
Presentation of Matthew Hause



Knowledge Centric Systems Engineering



The need for recursive engineering: TEAM COMMUNICATION



Requirements Engineers

IBM Rational

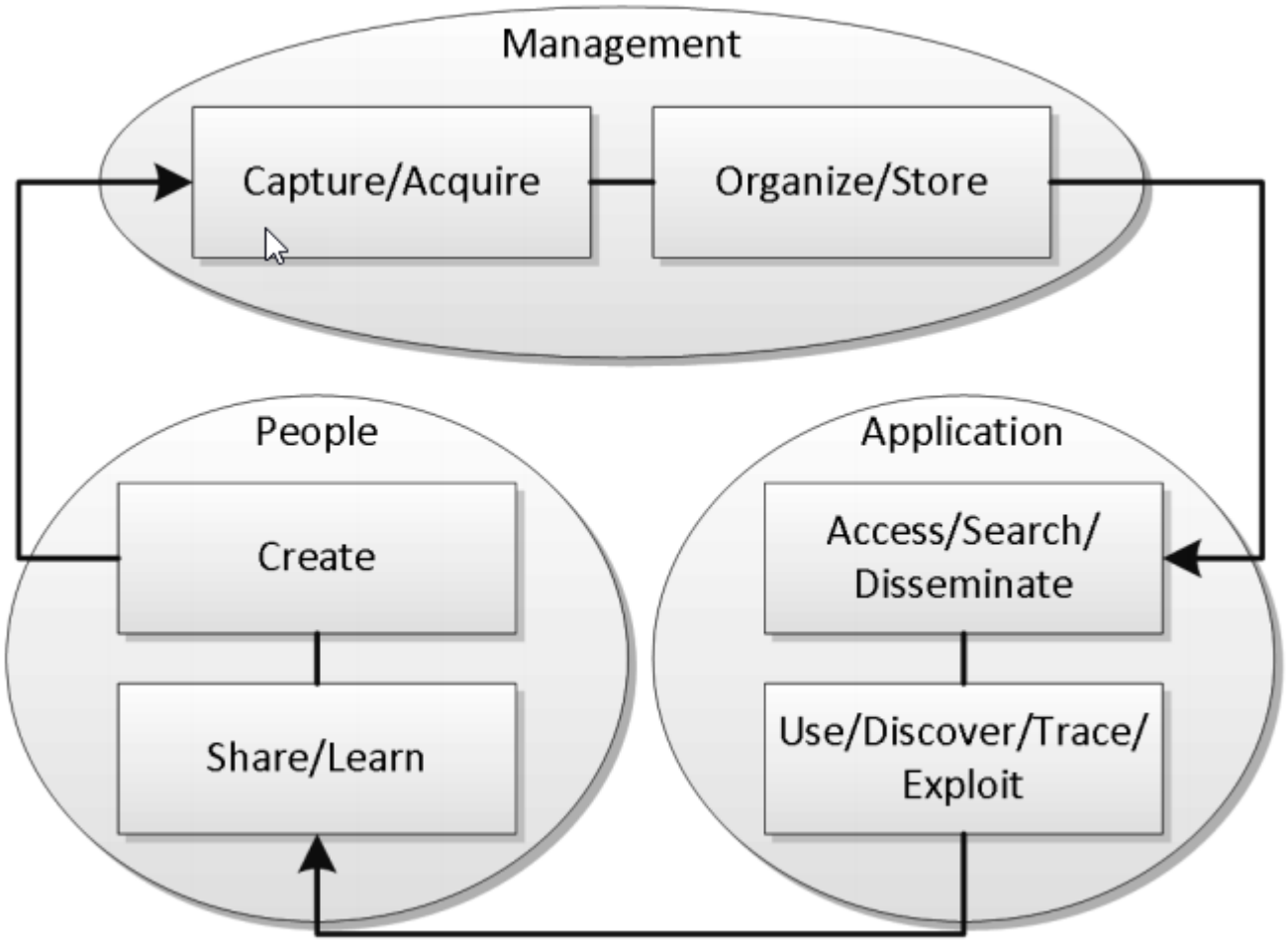
INCOMPLETE !?
INCONSISTENT !?

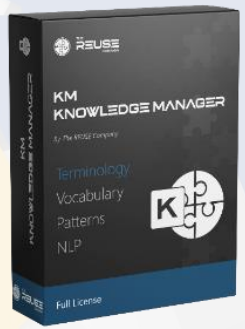
System Architects

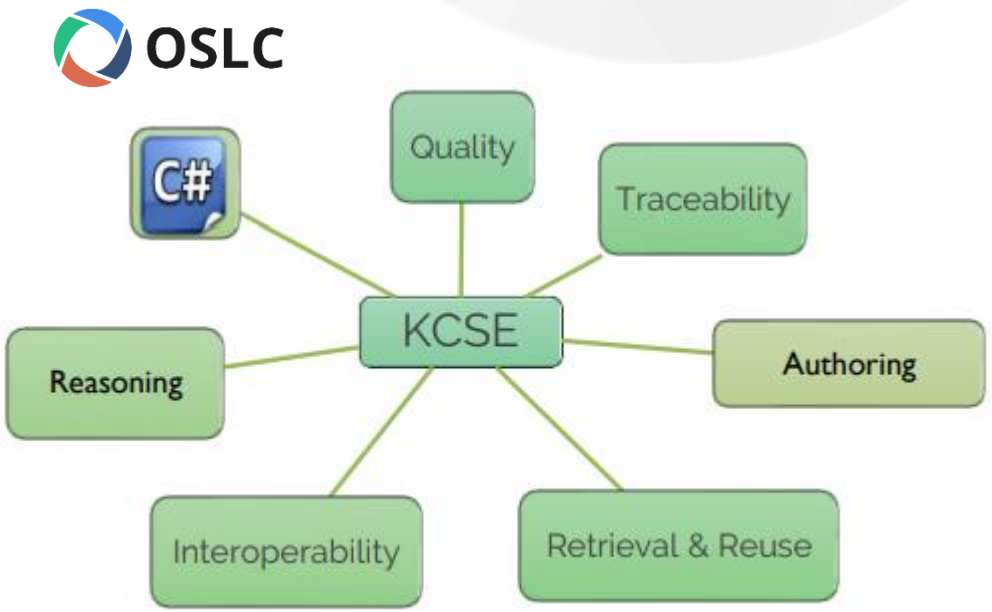
Capella

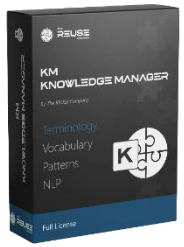
Money and schedule overrun!!

Source: ISO/IEC 29110 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs)









01 Vocabulary

Controlled Organizational and Project Vocabulary for a common understanding among stakeholders

02 SCM/Architectures

Recreate and capture the system architectures represented in views and models. Stablish relationships among system and system elements

03 Patterns

Represent requirements similarities and enable formal representation, automatic recognition and aid authors

04 Formalization

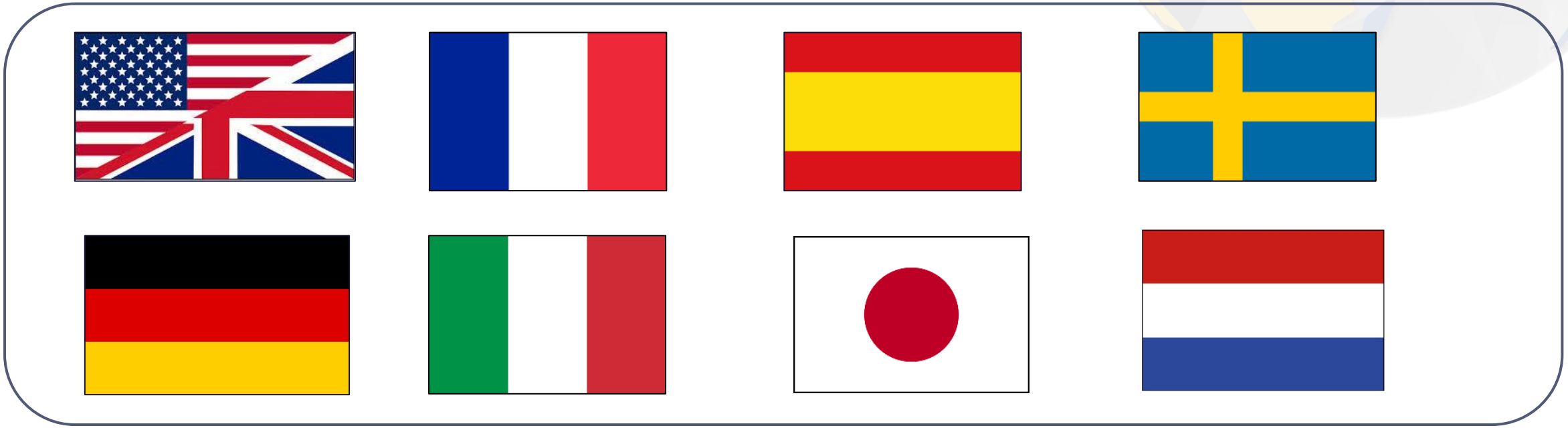
Representation of assets semantic through SRL – System Representation Language

05 Reasoning

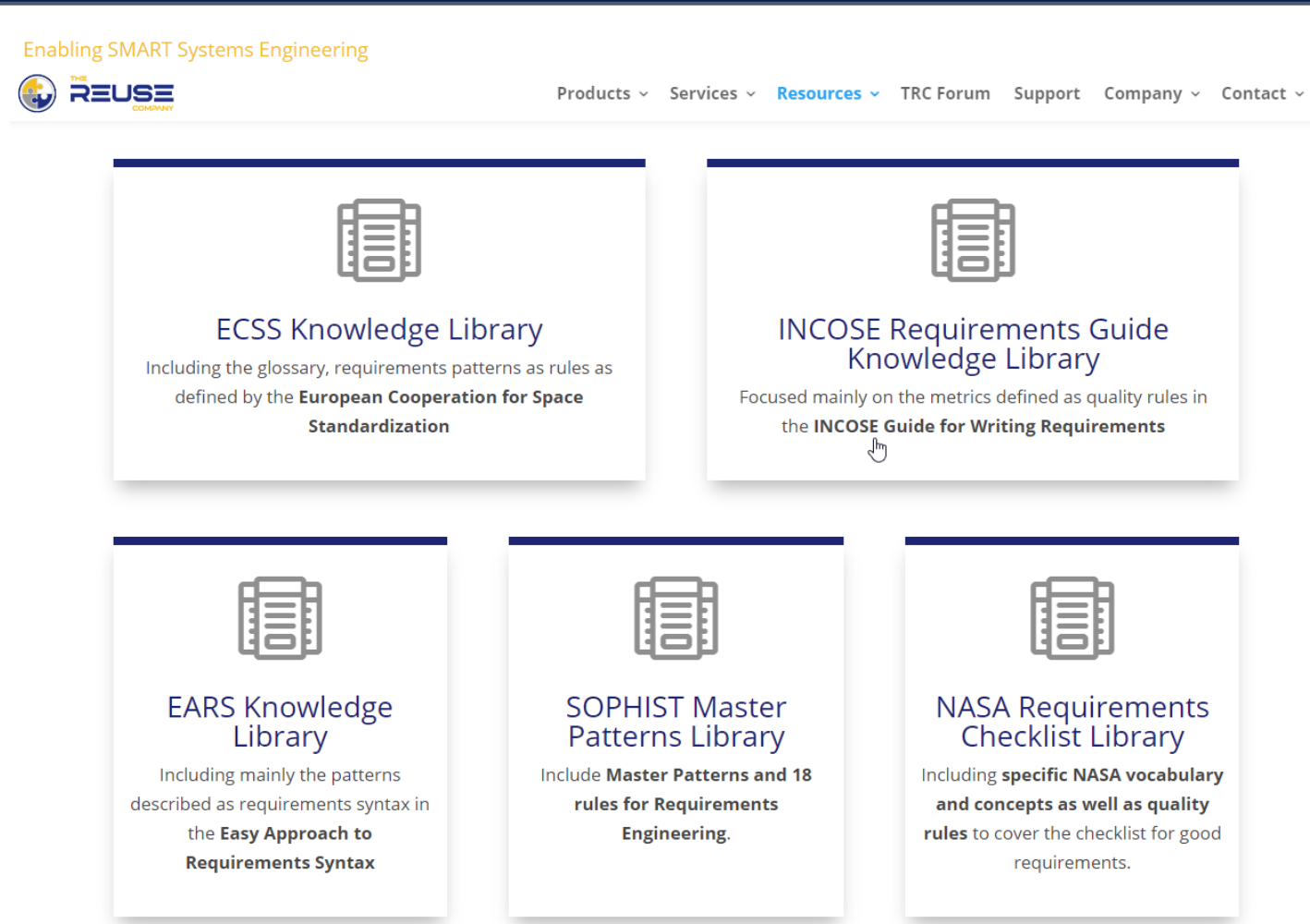
A combination of rules, tasks and groups to infer information from valuable assets

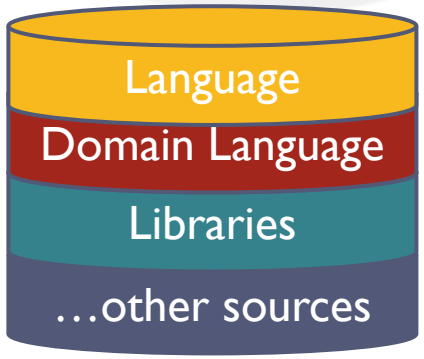
Language Ontologies: Supported languages

- The TRC SES Suite is highly dependent of the language of the requirements
- Languages supported so far:

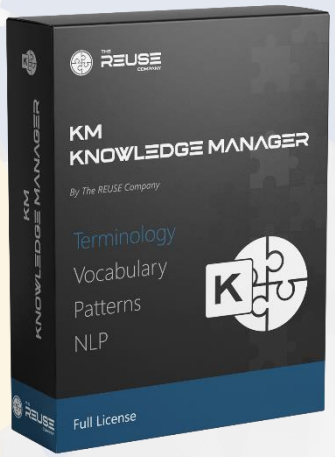


Building a System Knowledge base using existing Knowledge





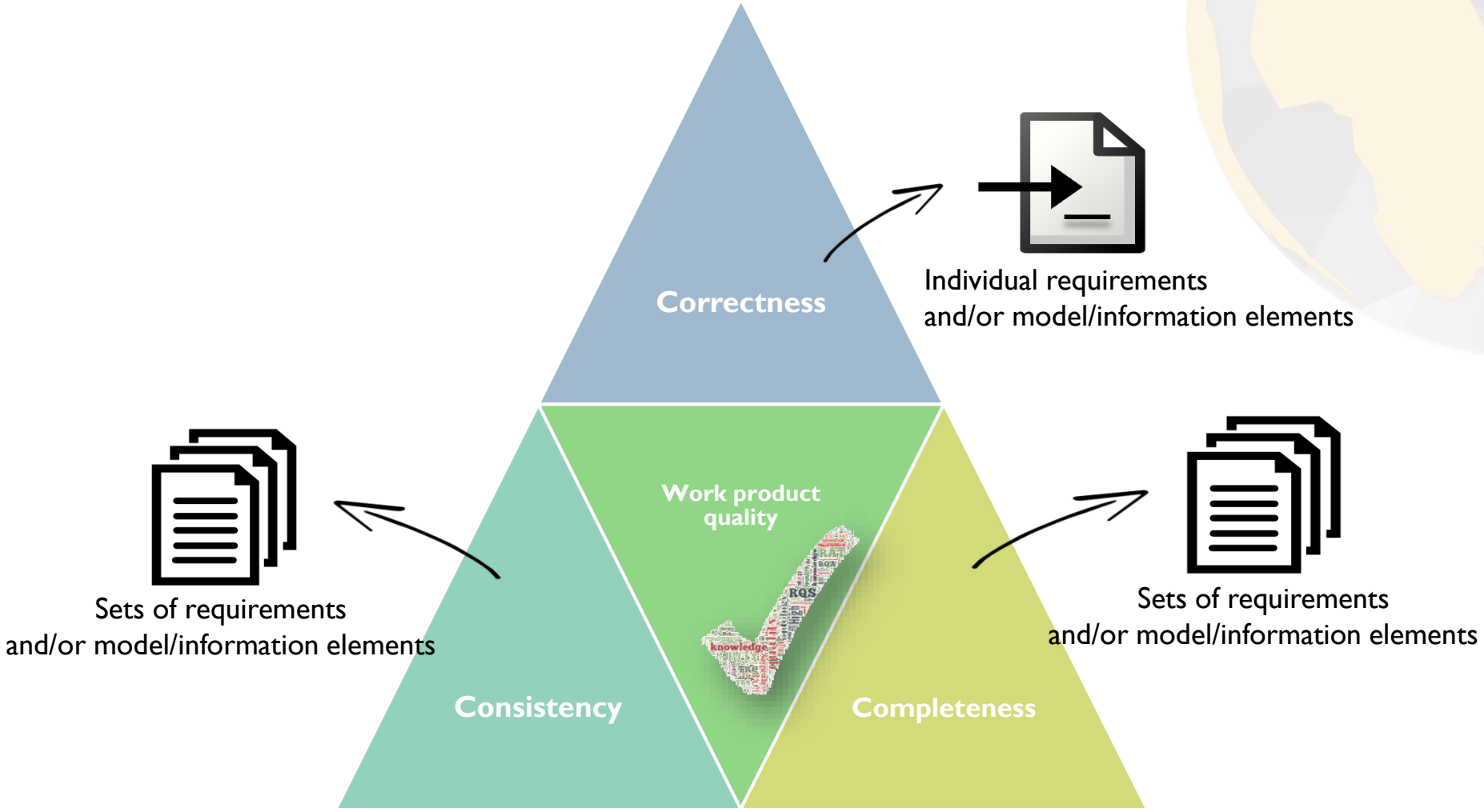
Free libraries possible to download and reuse



https://youtu.be/olysDbzwxCs

Work product quality

CCC – Correctness, Consistency and Completeness

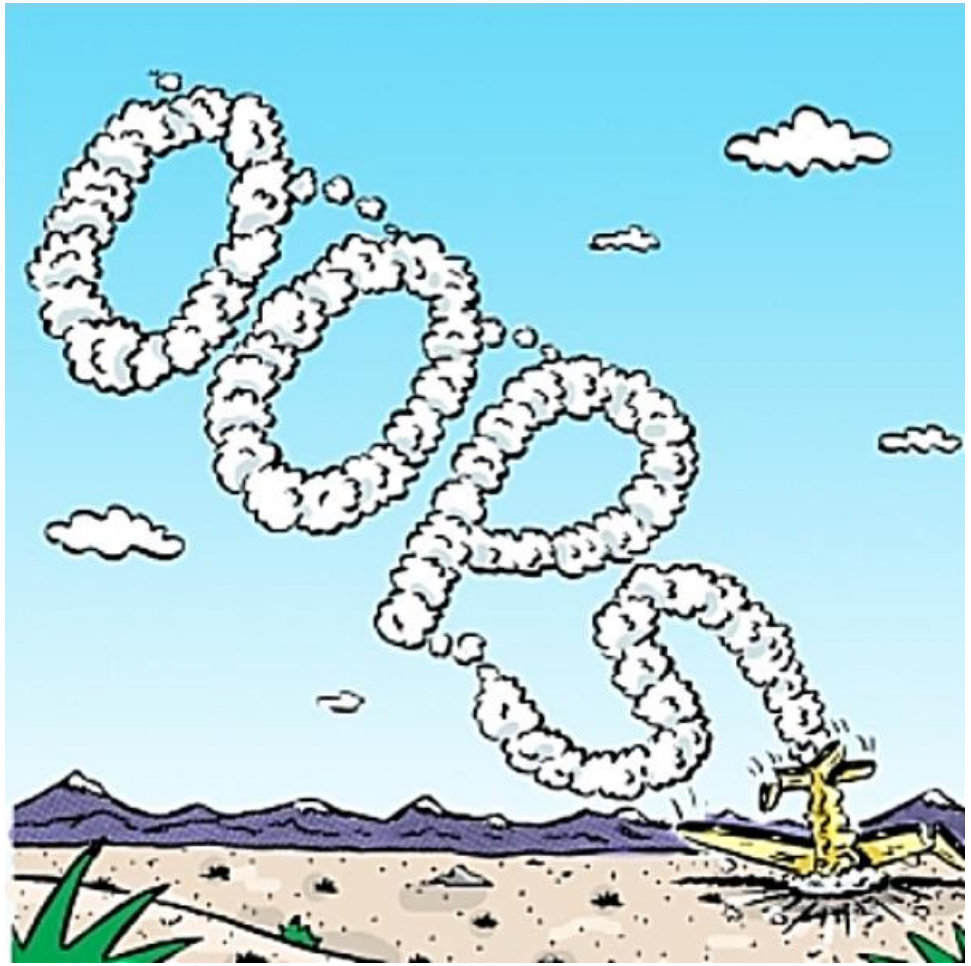


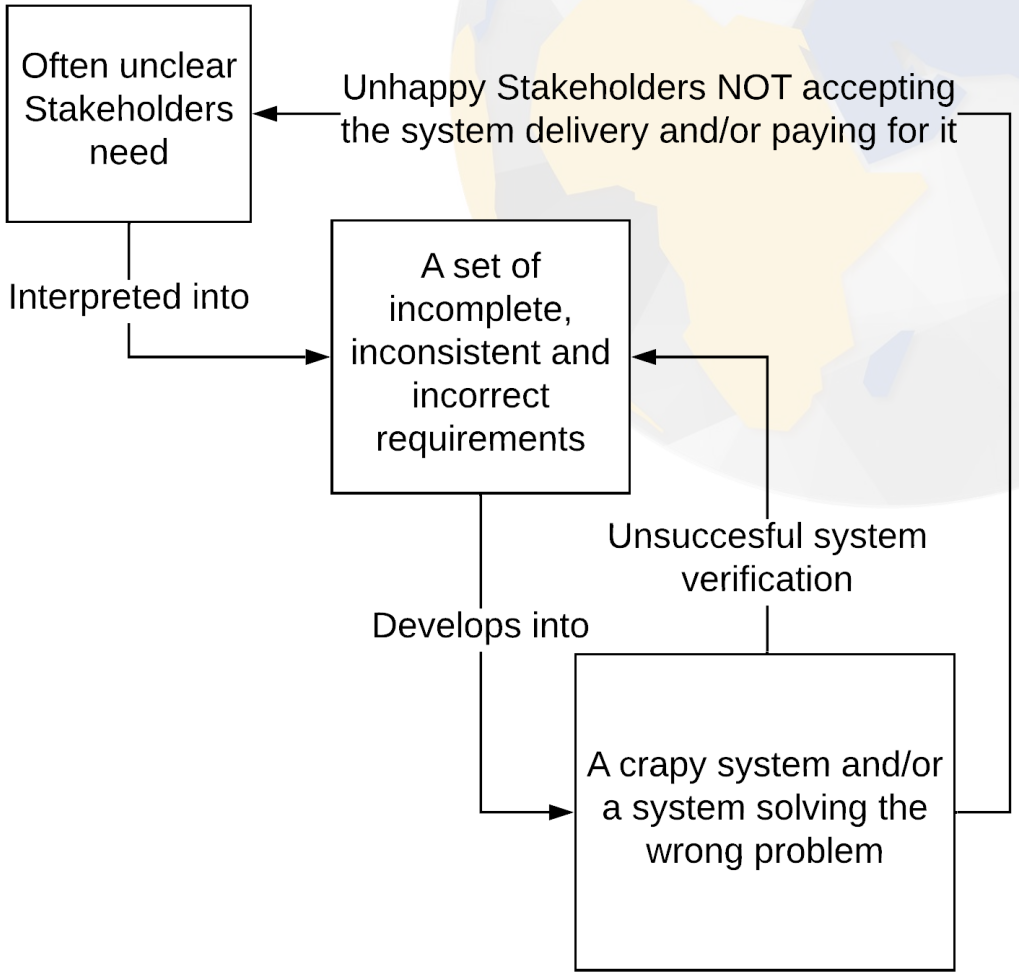
Poor requirements = High odds of failure!

*by Dr. Gina Guillaume-Joseph who studied over 200 failed projects with a total value of **15 billion dollars in sunken cost**. I was not surprised by the result. The main reason for project failure was ranked like this:*

Project failure factor	*Odds of project failure*
Project Schedule Delays	*78,6%*
Changing or Unclear Requirements	*29,9%*
Project Failure in Test Phase	*11,4%*

(Source: Improving Software Project Outcomes Through Predictive Analytics, ENGINEERING MANAGEMENT REVIEW, 2015)





PROBLEM-DOMAIN

Problem description
Why?

Stakeholder need issues

There is a strong risk that we don't understand the problem

- Defining the wrong problem (not the root cause)
- Risk that we don't understand the complete problem

SOLUTION-DOMAIN

Abstract solution
What?

Purchaser

System requirements issues

Risk that we don't capture the whole need (uncomplete)

- Risk that we are limiting the possible solution to early
- The solution is not optimal

Contractor

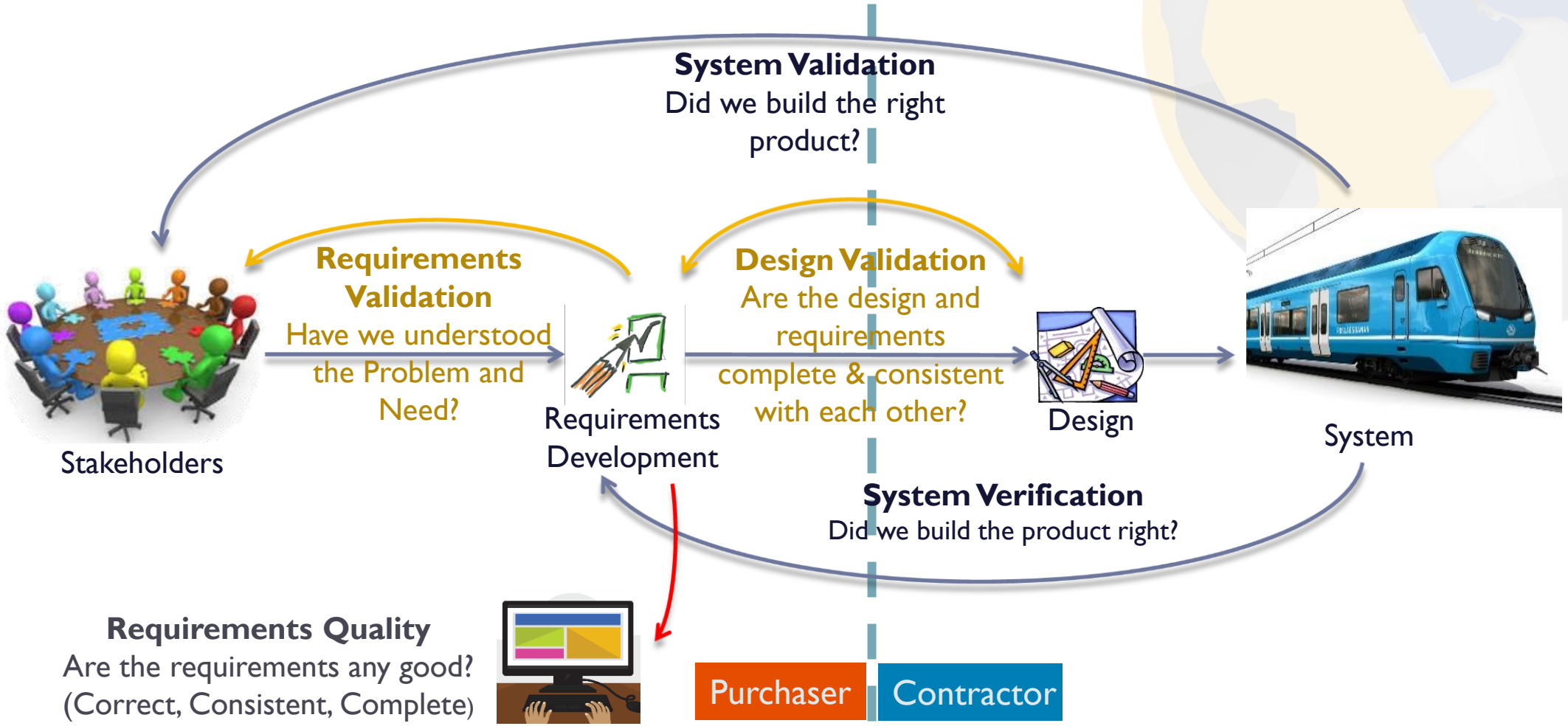
Specific solution
How!

Design issues

Risk that the Contractor don't meet the requirements

- Can not verify design against system requirements
- NDMA can not validate the against NAF need

Evolution of the Verification and Validation activities - More focus on information quality

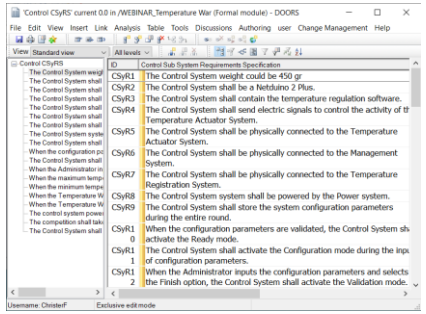


*"It's hard to imagine a world without **text**. A picture may well be worth a thousand words, but it's likely a different thousand words for each of us. Text gives our ideas a precision that we can rarely approach with images alone."*

Need model

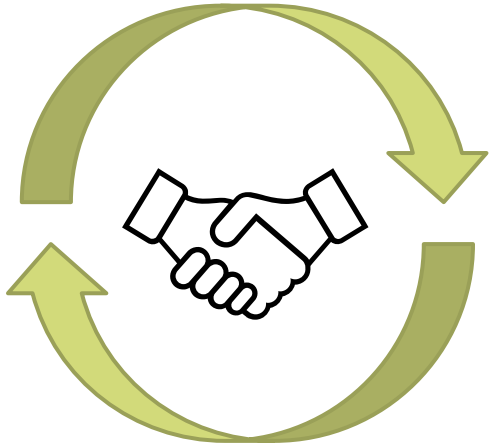
helps formalize and consolidate customer and system requirements

*"Our brain is hard-wired to process the world in a **visual form**. It's part of our 'native OS'. For at least 40,000 years, humans have been transferring information from one person to another with the help of images, pictograms and graphic symbols."*



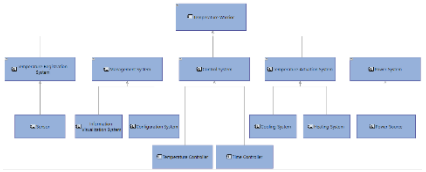
Textual requirements

are at the heart of the current engineering practices

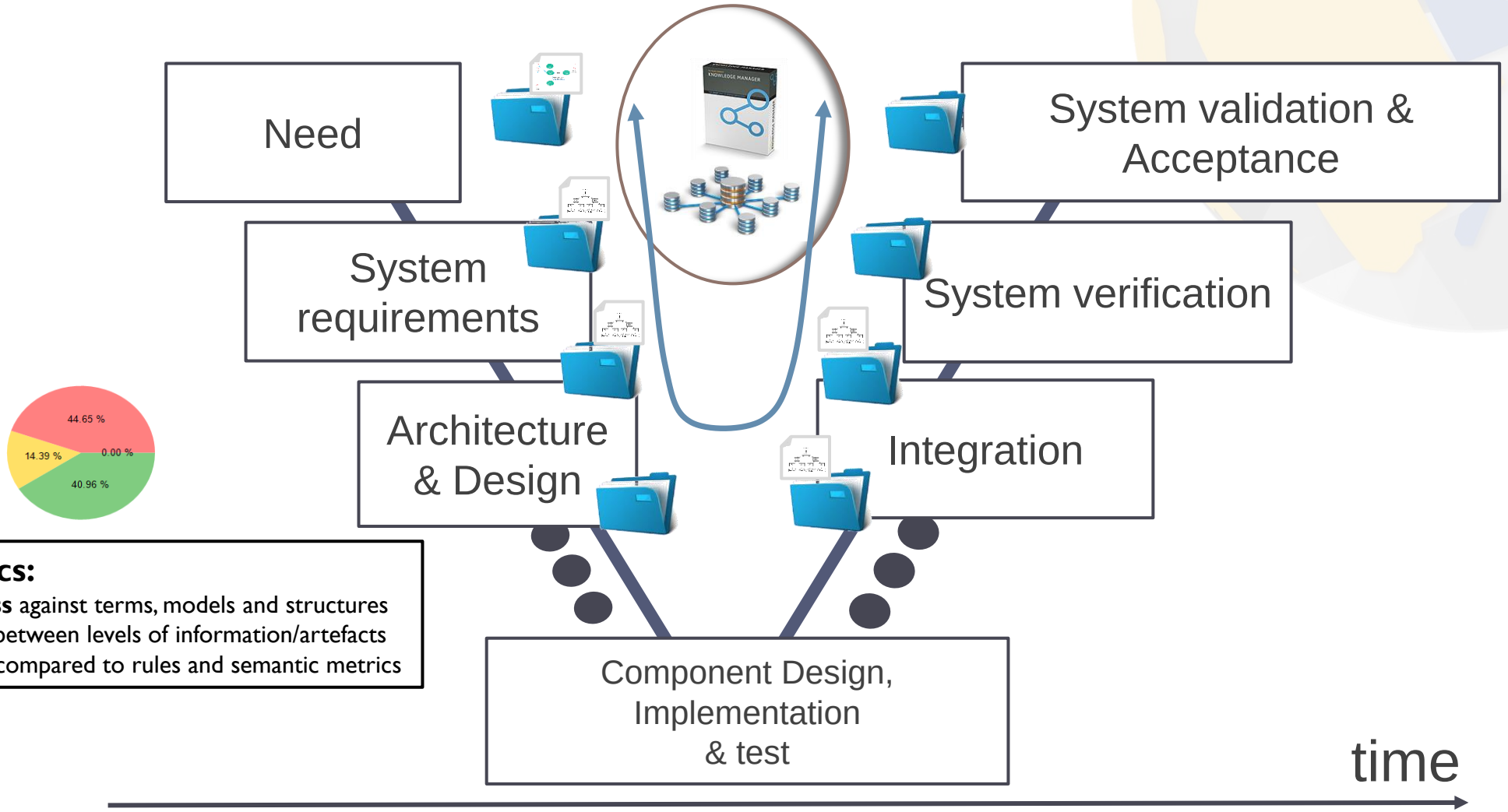


Solution model

helps validate feasibility, elicit/justify new requirements for the system/subsystems



System information quality – Need for Consistency and Completeness



QA Metrics:
Completeness against terms, models and structures
Consistency between levels of information/artefacts
Correctness compared to rules and semantic metrics

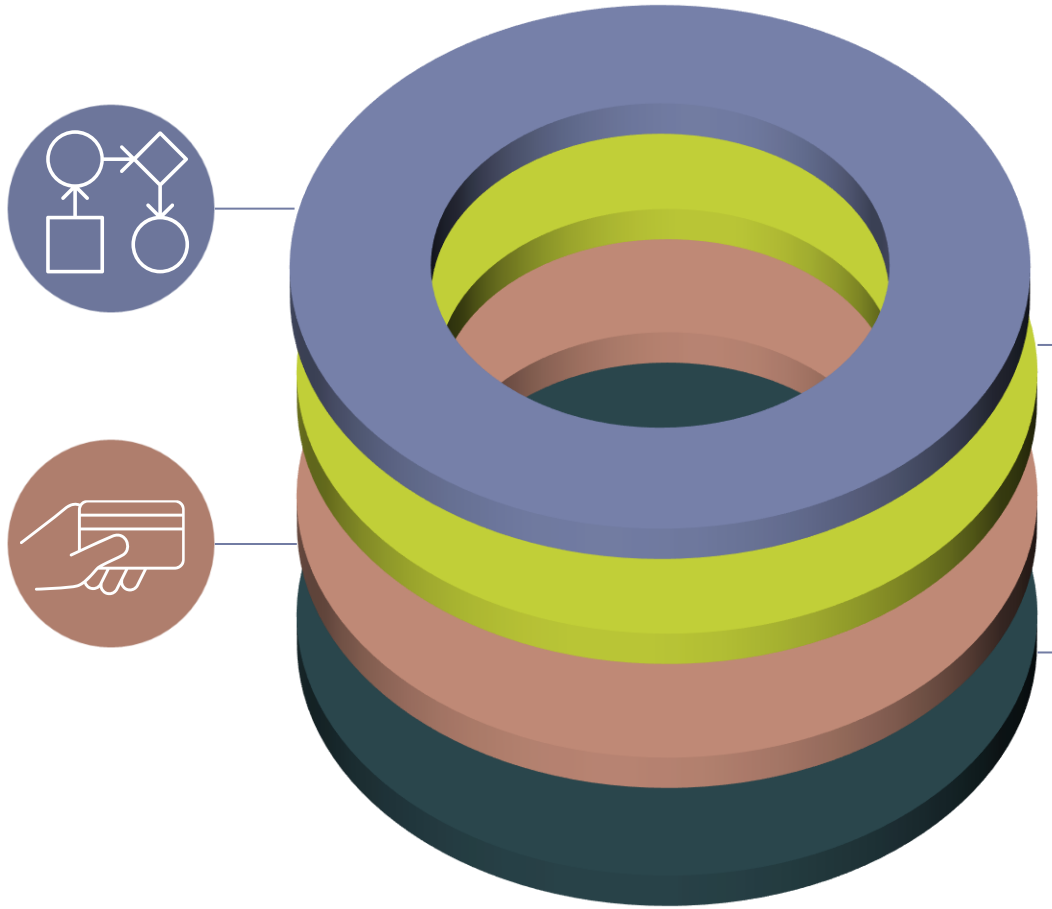
Real-time quality analysis: **Consistency**

Requirements-models Consistency

e.g. allocation of properties

Naming consistency

Among models elements and elements in textual requirements



Consistency Among requirements:

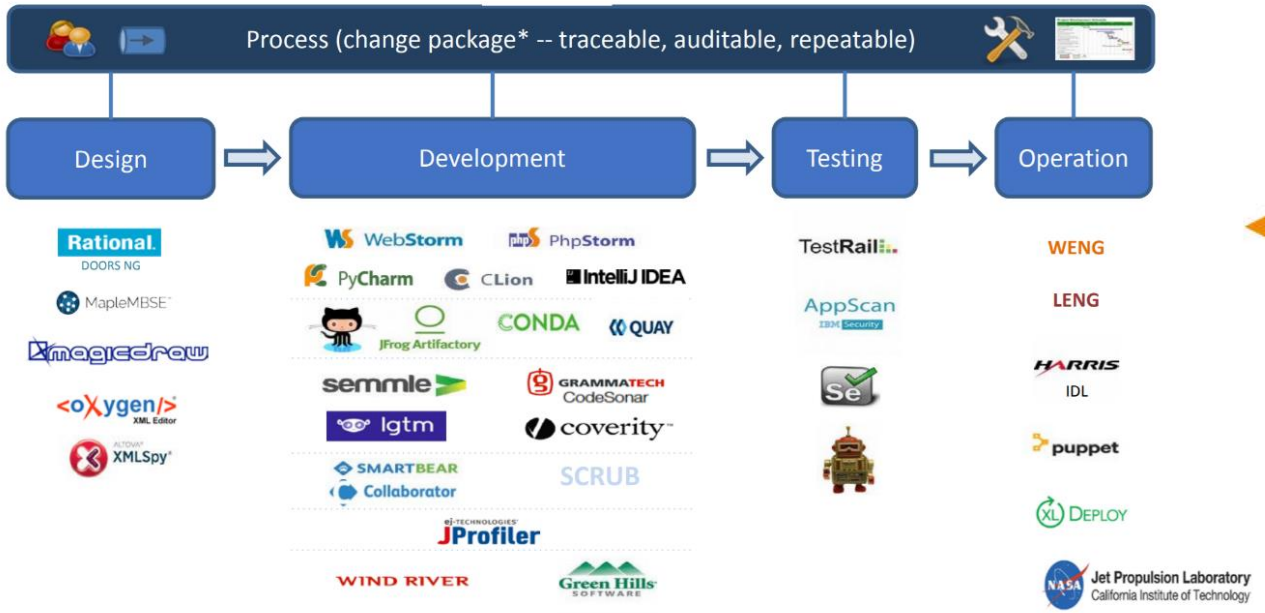
e.g. overlapping

Pattern-based writing

https://youtu.be/2yA1fe5t85Y

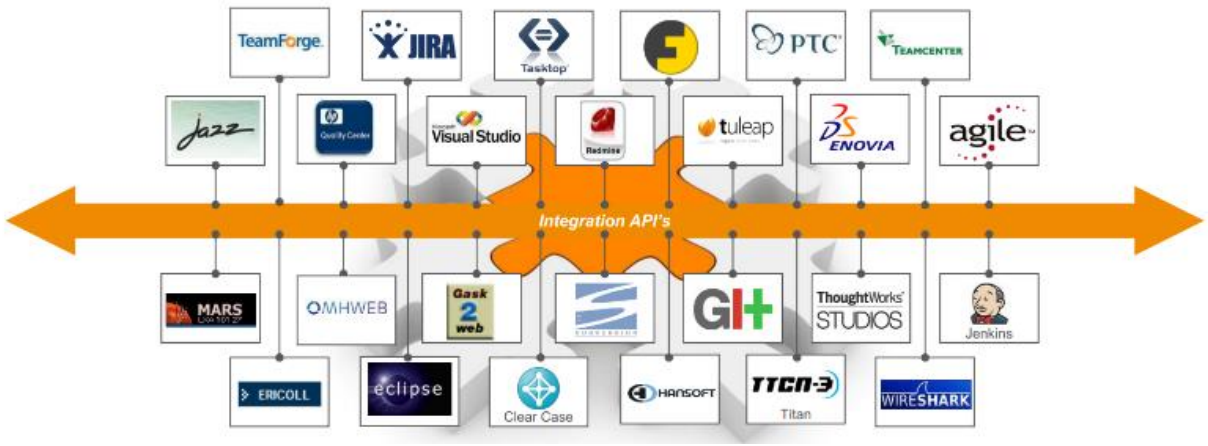
SE tool's interoperability

Engineering Digitalization – Some Tools Ecosystems



Safety-Critical Software Environment

Source: https://www.nist.gov/system/files/documents/2019/04/05/14_delp.pdf



Mats Berglund (Ericsson)

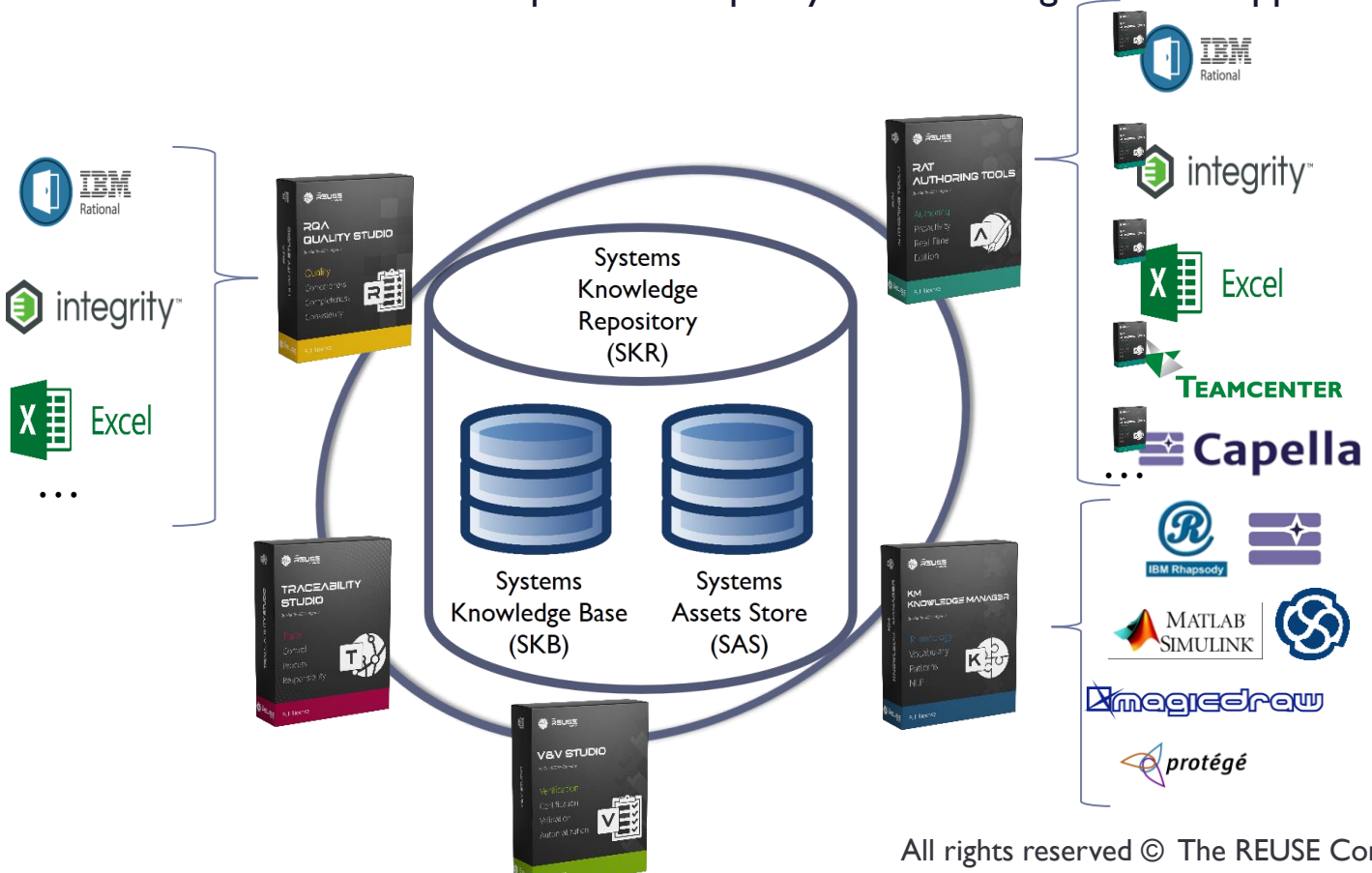
http://www.ices.kth.se/upload/events/13/84404189f85d41a6a7d1cafd0db4ee80.pdf

Tool interoperability

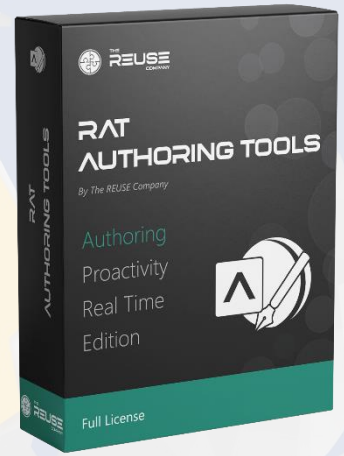
		Capabilities	Requirements Tools									Modeling Tools										Others		
			DOORS	DNG	PTC ILM	VISURE	Teamcenter	Excel	Reqif	Reqtify	3DX	Capella	Rhapsody	Cameo	EA	Simulink	XMI	Papyrus	ASCE	FMI/FMU	Pure Variants	OWL	Word	XML
Req. Analysis	Quality assessment	Analyze the quality of requirements (RQA)	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓		✓	✓	✓					✓
		Filter by views	✓	✓																				
		Filter by artifact type		✓																				
		Assessment by baseline		✓																				
		Store quality results back in the tool	✓	✓	✓	✓		✓																
		CCC with RQA	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓		✓	✓	✓					✓
	Requirements Authoring	RAT.exe. Correctness	✓	✓	✓	✓		✓	✓	✓		✓	✓											
		RAT.exe. CCC	✓	✓	✓	✓		✓	✓	✓														
		RAT.exe. Pattern-based authoring	✓	✓	✓	✓		✓	✓	✓														
		RAT Plug-in. Correctness	✓	✓	✓	✓	✓	✓			✓	✓	✓										✓	
		RAT Plug-in. CCC	✓		✓																			
		RAT Plug-in. Pattern-based authoring	✓		✓			✓				✓	✓										✓	
Contents		Capture vocabulary						✓				✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		
	Logical Models	Extract info from class/block diagrams											✓	✓	✓		✓	✓						
		Classes and Interfaces										✓	✓	✓	✓		✓	✓						
		Extract properties						✓					✓	✓	✓		✓	✓						
		Extract relations: hierarchical, aggregation						✓				✓	✓	✓	✓	✓	✓	✓	✓			✓		
		Extract info from state machines											✓	✓	✓		✓	✓						
		Extract states										✓	✓	✓	✓		✓	✓						
		Extract transitions										✓	✓	✓	✓		✓	✓						
		Capella Operational Architecture										✓												
		Capella Capabilities										✓												
		Capella Dataflows										✓												
		Capella Architecture										✓												
		Capella Trees										✓												
		Extract info from sequence diagrams										✓	✓	✓	✓		✓	✓						
		Extract info from packages elements											✓	✓	✓		✓	✓						
		Extract info from use cases and actors											✓	✓	✓		✓	✓						
		Extract info from activities											✓	✓	✓		✓	✓						
	Physical models	Extract signals											✓	✓	✓		✓	✓						
		Signal values											✓	✓	✓		✓	✓						
		Simulink Blocks														✓								
		Simulink Common Block Properties														✓								
		Simulink Block-Specific Parameters														✓								
	Others	Safety case diagrams																	✓					
		Feature models vocabulary																			✓			
		Export future models																			✓			
		Ontological information																				✓		

The TRC Systems Engineering Suite Interconnectivity

- The Systems Engineering Suite tackles requirements quality management by offering a set of tools and processes
- Automatic measurement of requirements quality metric
- Support to Requirements Authoring
- SES Suite models requirements quality metrics using the CCC approach (Correctness, Consistency and Completeness)

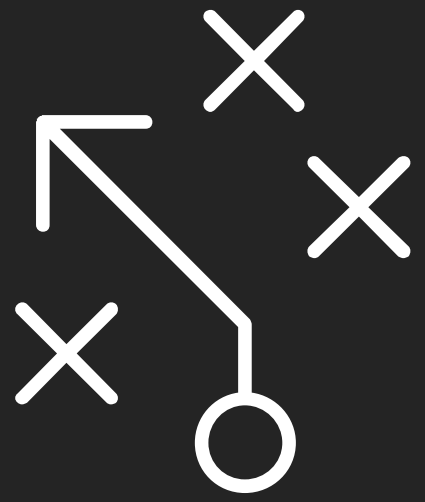


- **RQA Quality Studio / V&V Studio:** to setup, check and manage the quality of a requirements specification
- **Rich Authoring Tool (RAT):** to assist authors while they are creating or editing requirements
- **Knowledge Manager (KM):** to manage knowledge around a requirements specification: dictionaries, glossaries, concept maps, knowledge models, ontologies, patterns…



https://youtu.be/O6lql1R-vaw (Rat4Capella)

https://www.youtube.com/playlist?list=PLSf7bEUEIX8am7AzkKfEdD6eSZDjPyX1n (RAT for IBM, PTC, etc..)



Traceability of work products

What is traceability?: INCOSE

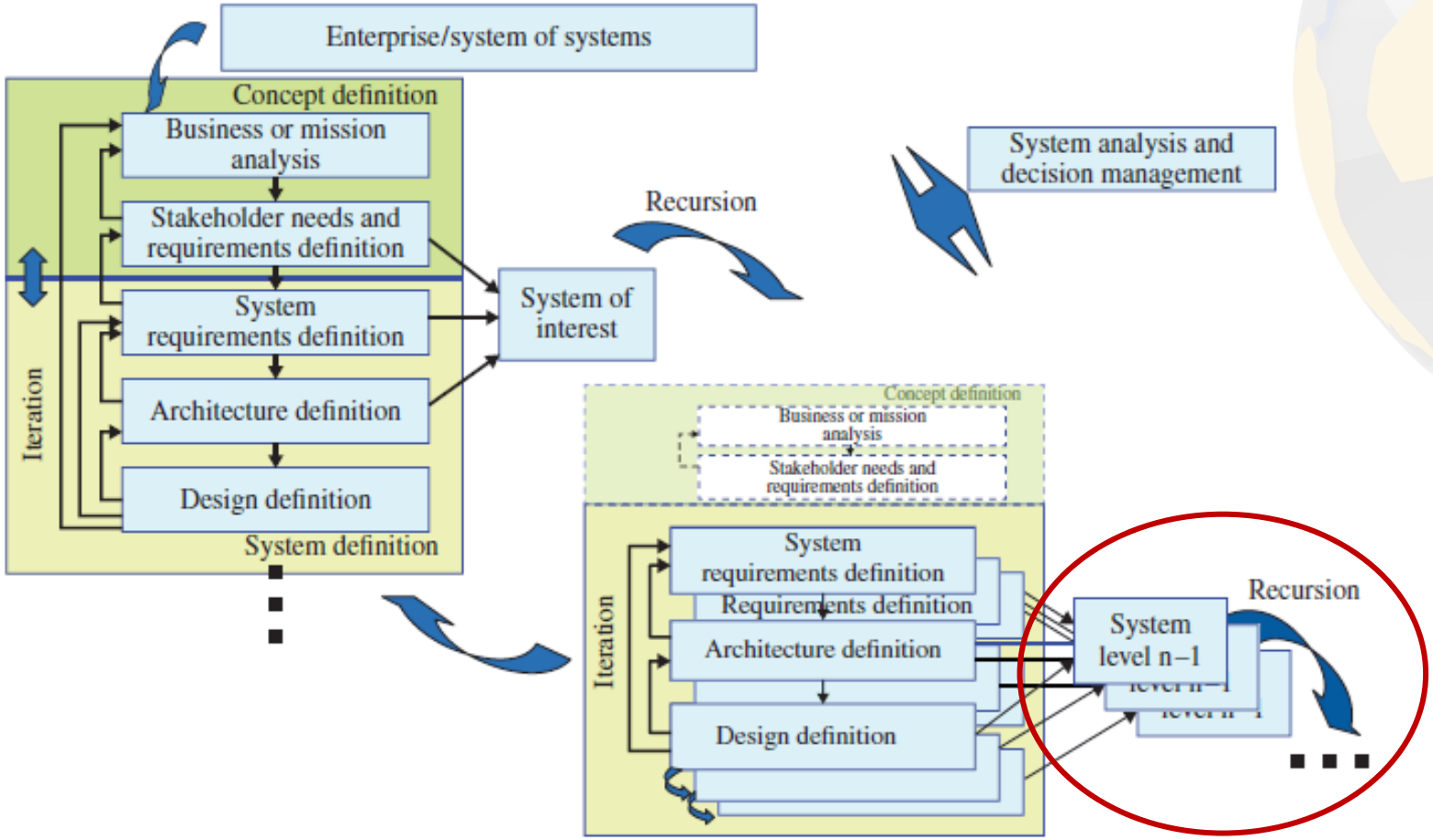


FIGURE 3.5 Iteration and recursion. Reprinted with permission from Garry Roedler. All other rights reserved.

Source: INCOSE Systems Engineering Handbook, Ed. 4

What is traceability?: INCOSE

"**Manage System Requirements:** Establish and maintain traceability between the system requirements and the relevant elements of the system definition (e.g., stakeholder requirements, architecture elements, interface definitions, analysis results, verification methods or techniques, and allocated, decomposed and derived requirements."

Source: INCOSE Systems Engineering Handbook, Ed. 4

"**Requirement's traceability** is the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases)."

Source: Gotel and Finkelstein

- Project quality:
  - Are all the requirements properly tested, verified and validated?
  - Completeness:
    - Have we considered every high level requirement?
    - Have we created all the expected work products following requirements
- Scope management (project control):
  - Clear reference to source…
  - … to avoid Gold plating / scope creep
- Visibility:
  - Impact analysis
- Collaboration:
  - Among different roles: requirements manager, architects, designers, testers and… above all, Project manager

What is traceability?: Basic approach

Sample traceability matrix

Requirement Identifiers	Reqs Tested	REQ1 UC 1.1	REQ1 UC 1.2	REQ1 UC 1.3	REQ1 UC 2.1	REQ1 UC 2.2	REQ1 UC 2.3.1	REQ1 UC 2.3.2	REQ1 UC 2.3.3	REQ1 UC 2.4	REQ1 UC 3.1	REQ1 UC 3.2	REQ1 TECH 1.1	REQ1 TECH 1.2	REQ1 TECH 1.3
Test Cases	321	3	2	3	1	1	1	1	1	1	2	3	1	1	1
Tested Implicitly	77														
1.1.1	1	x													
1.1.2	2		x	x											
1.1.3	2	x											x		
1.1.4	1			x											
1.1.5	2	x												x	
1.1.6	1		x												
1.1.7	1			x											
1.2.1	2				x		x								
1.2.2	2					x		x							
1.2.3	2								x	x					
1.3.1	1										x				
1.3.2	1										x				
1.3.3	1											x			
1.3.4	1											x			
1.3.5	1											x			
etc....															
5.6.2	1														x

- ✓ Might be good as a first step
- ✓ Allows to check where every requirement comes from
- ✓ Allows you to check completeness of tests
- ✓ Ensures that implementation meets specification
- ✓ Enough for some projects (e.g. SW)
- ✗ Not enough in complex projects
- ✗ No support to req. decomposition or design
- ✗ Doesn't satisfy standards like: ARP4754, DO-174, DO-254, ISO26262…

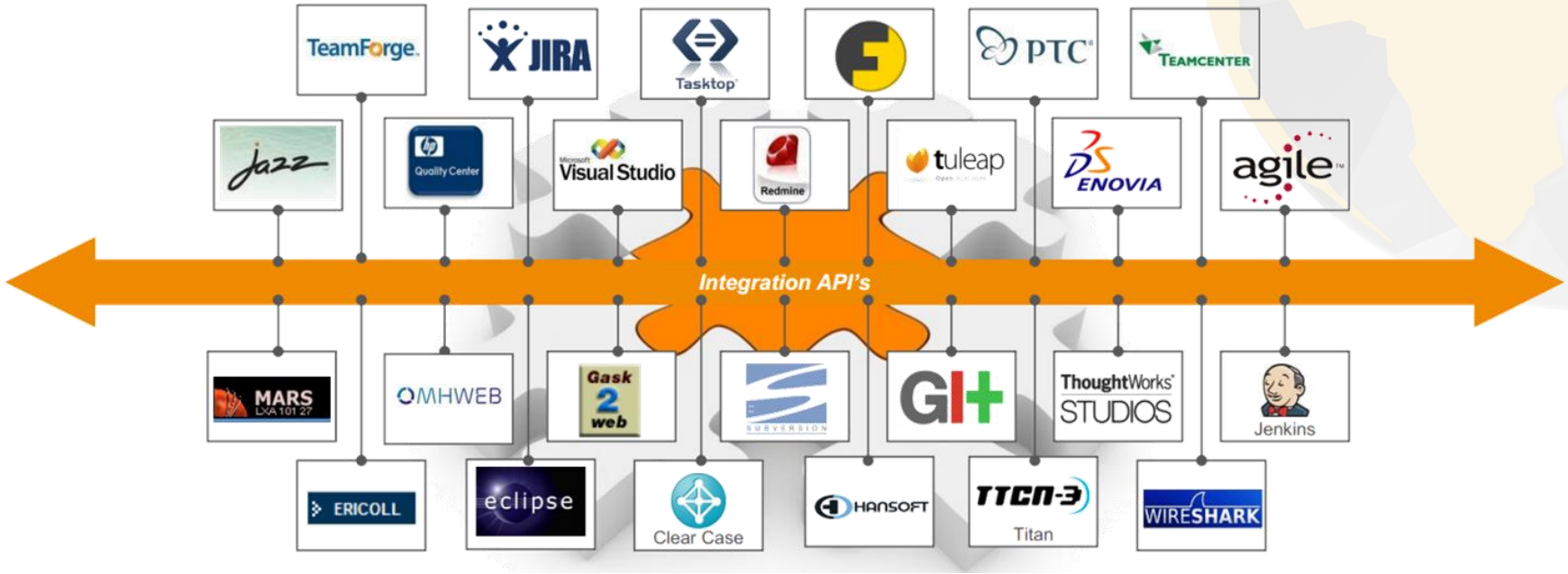
"Safety Requirements shall be traceable with a reference being made to:

- a) each source of a safety requirement at the next upper hierarchical level;
- b) each derived safety requirement at the next lower hierarchical level, or to its realization in the design; and
- c) the verification specification."

Source: ISO 26262

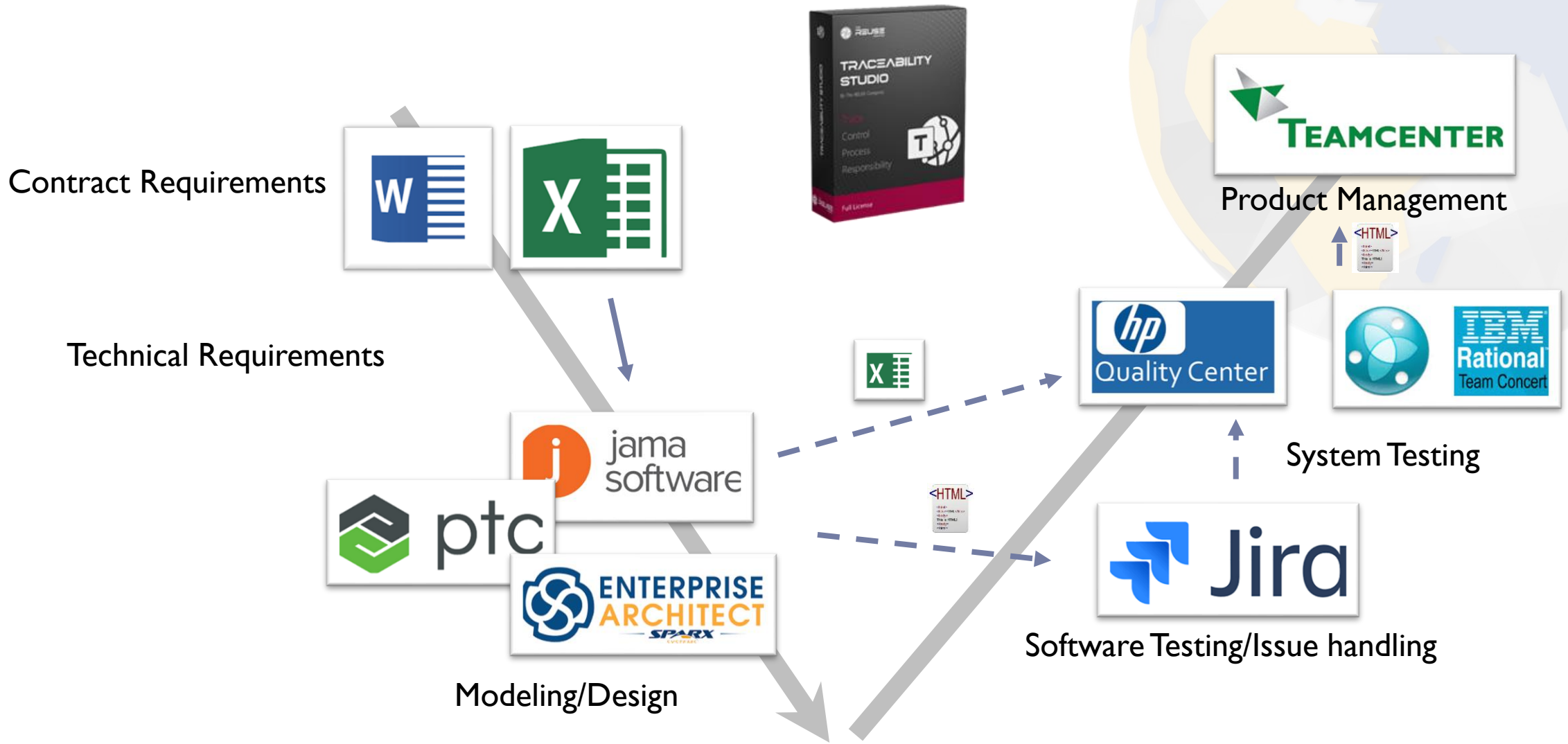
Traceability in complex projects: complex ecosystems

Mats Berglund (Ericsson) http://www.ices.kth.se/upload/events/13/84404189f85d41a6a7d1cafd0db4ee80.pdf

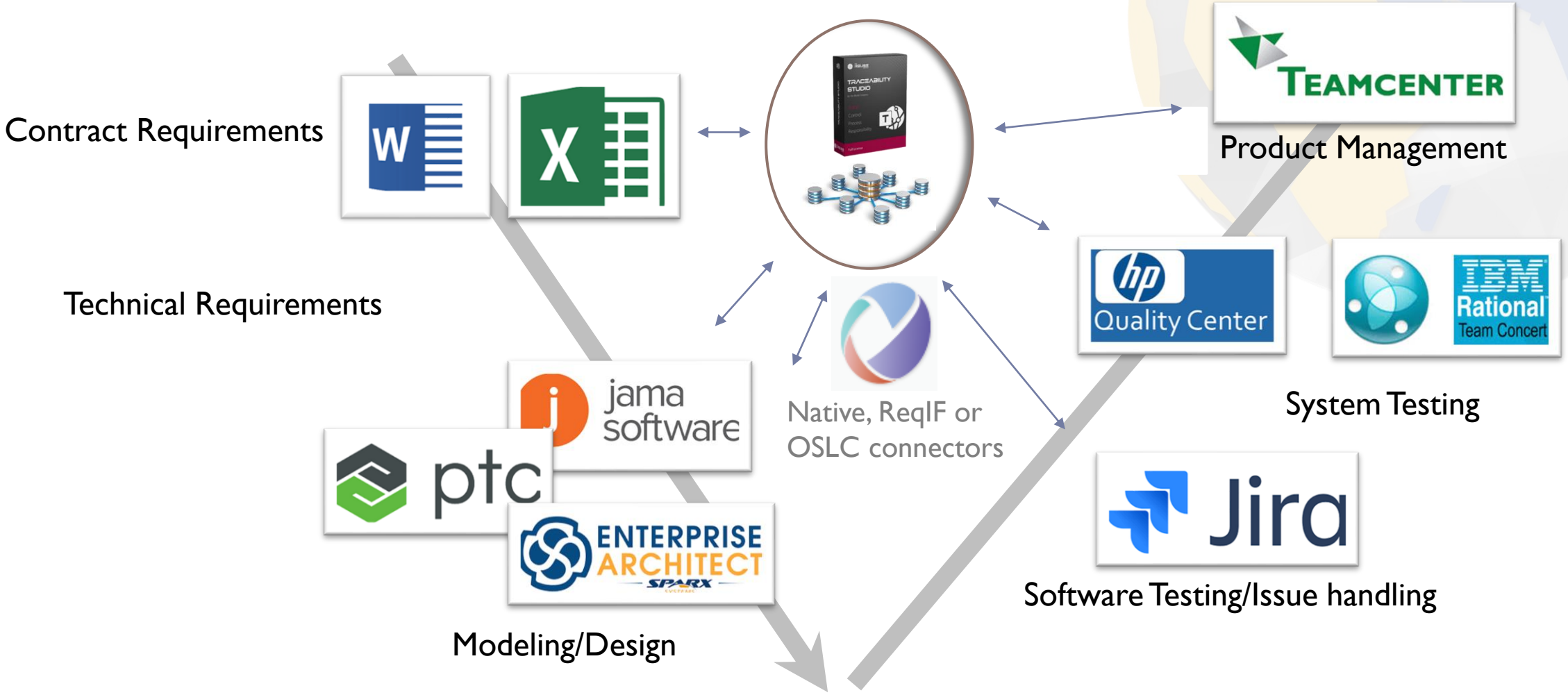


- Multiple **domains**
  - Different **types of artifacts**
- Need of **intra-operability**
  - Intra-domain
- Need of **interoperability**
  - Inter-domain

System quality depends on Traceability



Traceability Studio - Traceability between all relevant work products



P

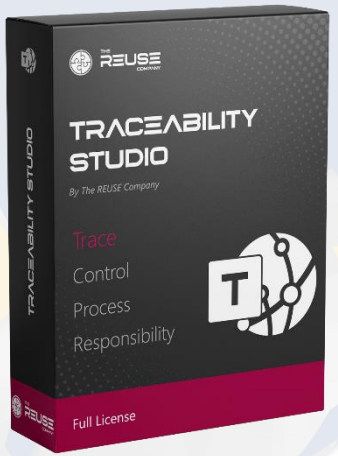
Pros

It's a **must** in complex and safety critical projects
Requested by **standards** and **good practices**: ISO26262, ARP4754…
Provides **visibility** in complex projects

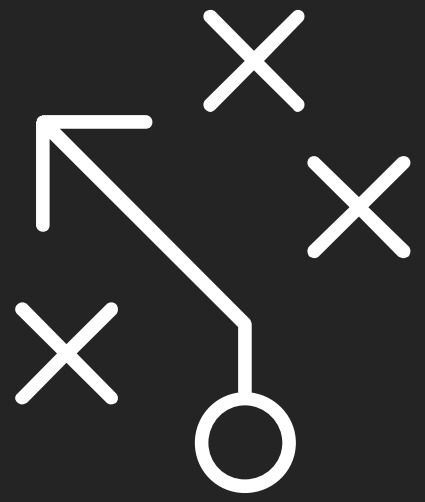
C

Cons

It's normally a **tedious** and **manual** task
Involves **connection** of tools that are not easy to connect



https://youtu.be/WQAslzsYDV8



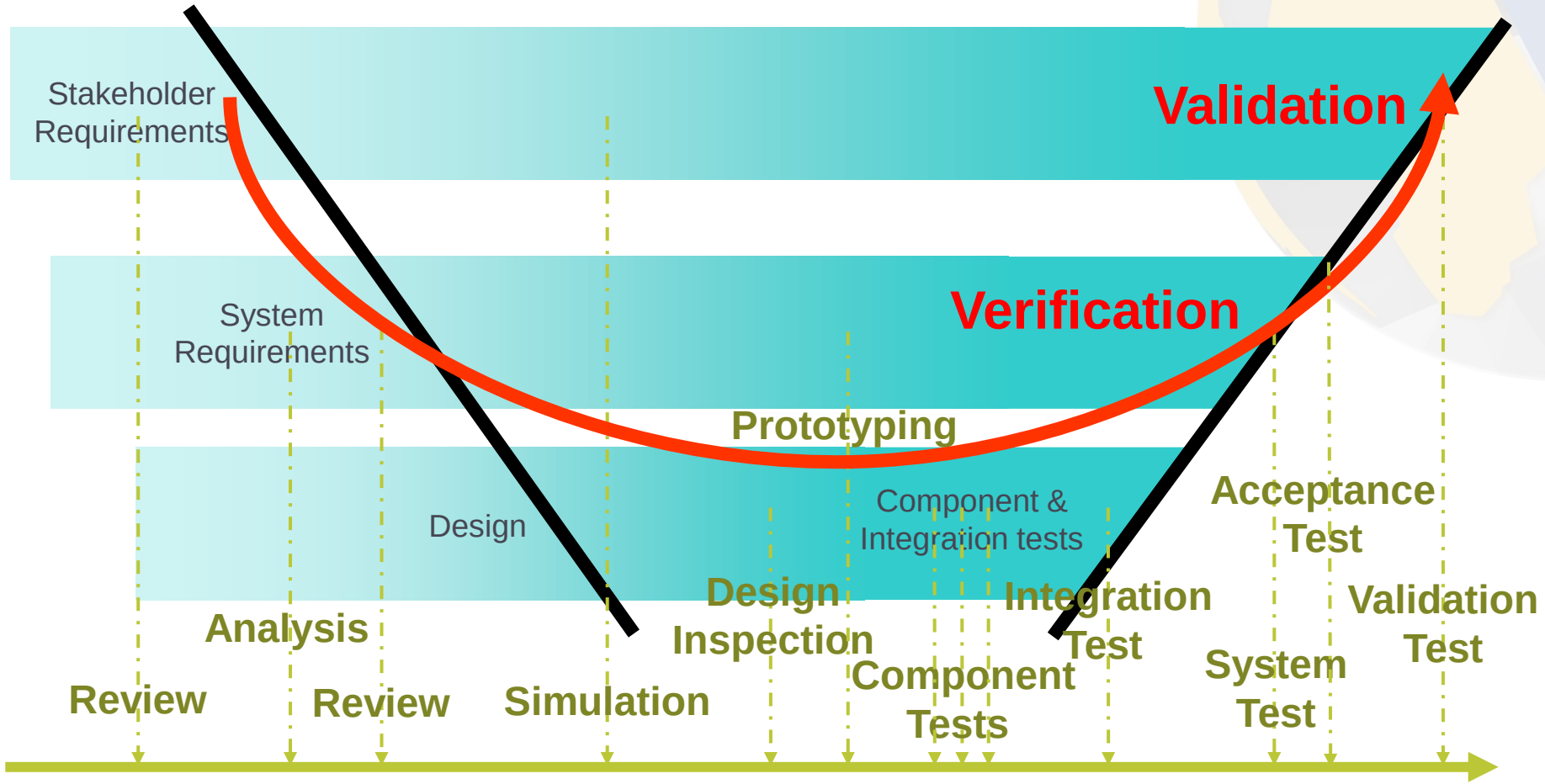
Verification & Validation

Why Verification and Validation (V&V)?

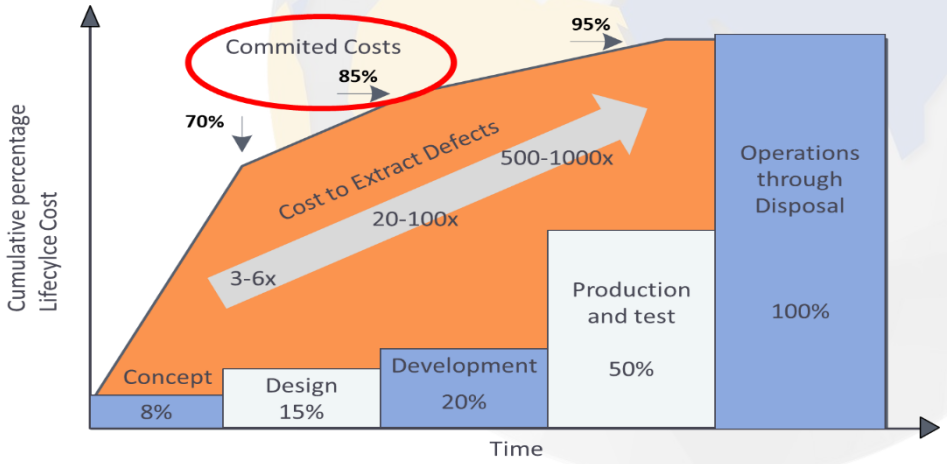
- To obtain a useful system
  - The issue of paramount importance to the user is system performance; i.e., will it fulfill the mission. The verification and validation process provides data to tell the user how well the system is performing during development and if it is ready for fielding.
- To reduce Risk
  - The program manager must balance the risks of cost, schedule and performance to keep the program on track to production and fielding. The responsibility of decision-making authorities centers on assessing risk tradeoffs.

- Verification – Have we built the system right?
- Validation – Have we built the right system?

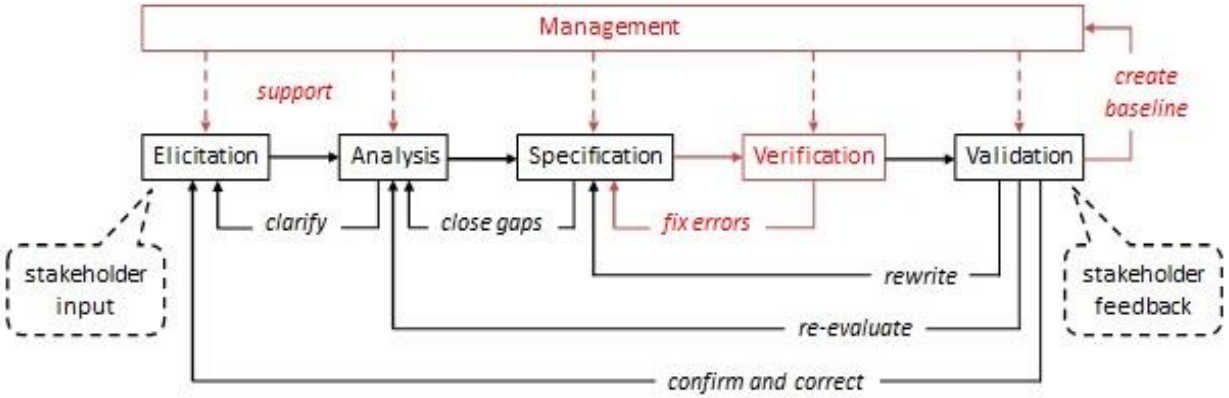
Verification &Validation Strategy – Risk Reduction



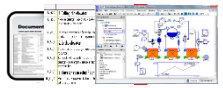
- The universalization of the Verification process can be found in the notion of quality
  - the less defects of any type produced during all the different life-cycle stages …
  - the better it will be to reduce cost, calendar time and quality… and V&V



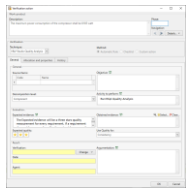
- Combine Verification with Quality control + assurance is the kernel of this approach
  - One example of this combination is to apply the verification process to requirements:, requirements must be validated, and (now recently) verified



Prepare for Verification

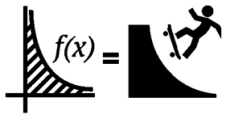


Select the Items to be Verified



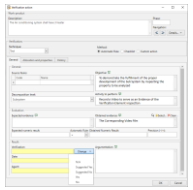
Define a Verification Action for each Verifiable Item

Prepare Resources of all types, Identify constraints, Enabling Systems, etc.



Define a Verification Technique for each Verifiable Item

Perform Verification



Execute the Verification Action for each Verifiable Item. If possible, the computer gathers the Information automatically

Computer applies the OK / KO decision process based on the standard guidelines.

Engineer defines the final state of the Verification process

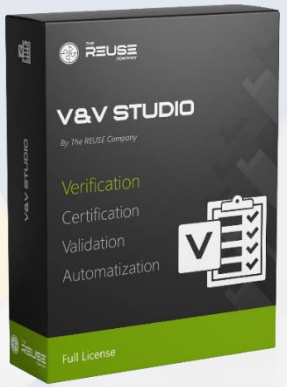
Manage Results of Verification

Manage and record discovered anomalies and evidences

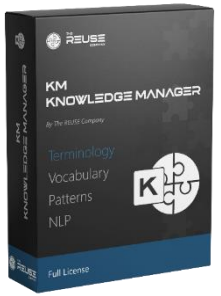
Track the Verification Process and manage Configuration

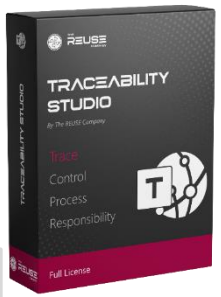
Build and maintain the RTVM

Provide proper reports



https://youtu.be/ZvsFYKE5Fbg





Need

System requirements

Architecture & Design

Component Design, Implementation & test

Integration

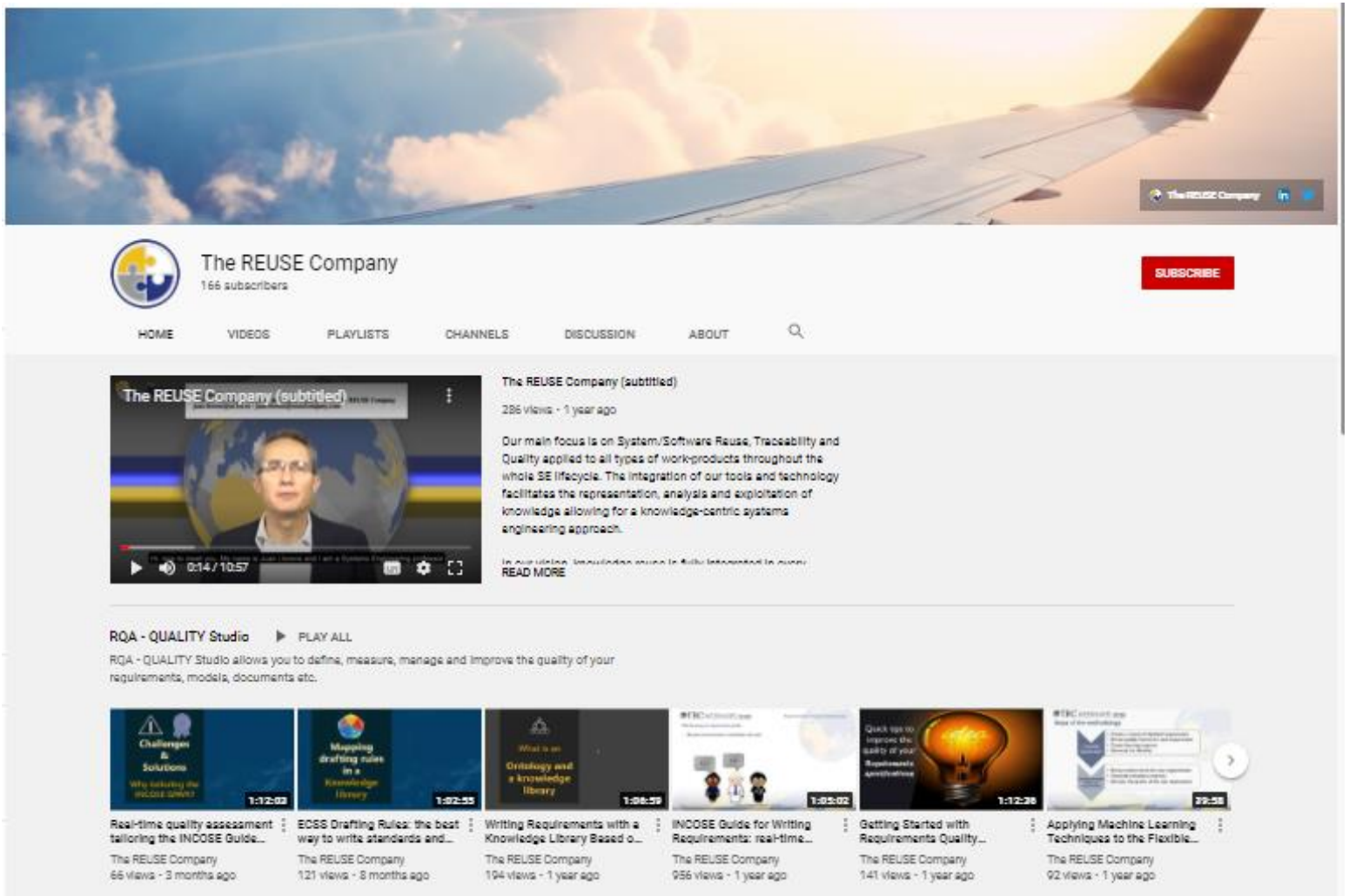
System verification

System validation & Acceptance

time

The End

- **Tailoring and Deployment of SE Suite in F4E**
- F4E is the European Organisation managing Europe's contribution to ITER, the world's largest fusion experiment. F4E is responsible for 45% of the ITER project and authored 700+ technical specifications with 80,000+ requirements in total.
- The Systems Engineering Group adopted the SE Suite from The REUSE Company to improve requirements quality and support the Requirements Management & Verification (RMV) process.
- This presentation will focus on the steps that were taken to tailor the SE Suite to F4E's needs, lessons learned and the results from the pilot projects which have started using the SE Suite.
- **Date:**
  - June 29, 2021



The REUSE Company in Youtube:

https://www.youtube.com/user/TheREUSECompany


Q&A

Contact information

Christer Fröling – Reuse Company Scandinavia

christer.froling@reusecompany.com

+46 (0)72 232 24 63

@ReuseCompany

www.linkedin.com/in/christerfroling

THE REUSE COMPANY