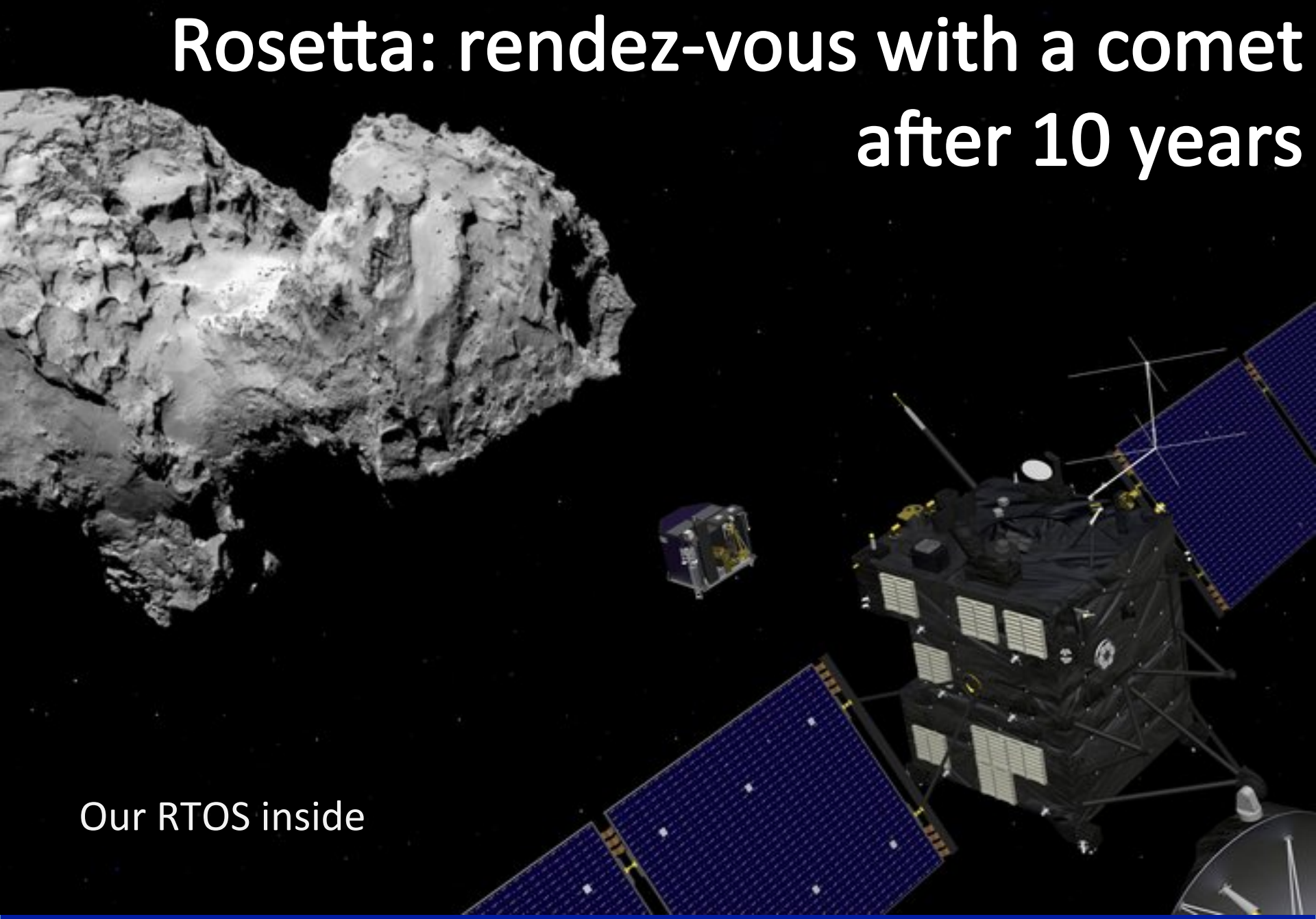
Altreonic presents:

A generic methodology for safety engineering

Developing certifiable products and systems in a cost-efficient way by generating the evidence during development with ***GoedelWorks***

Rosetta: rendez-vous with a comet after 10 years



Our RTOS inside

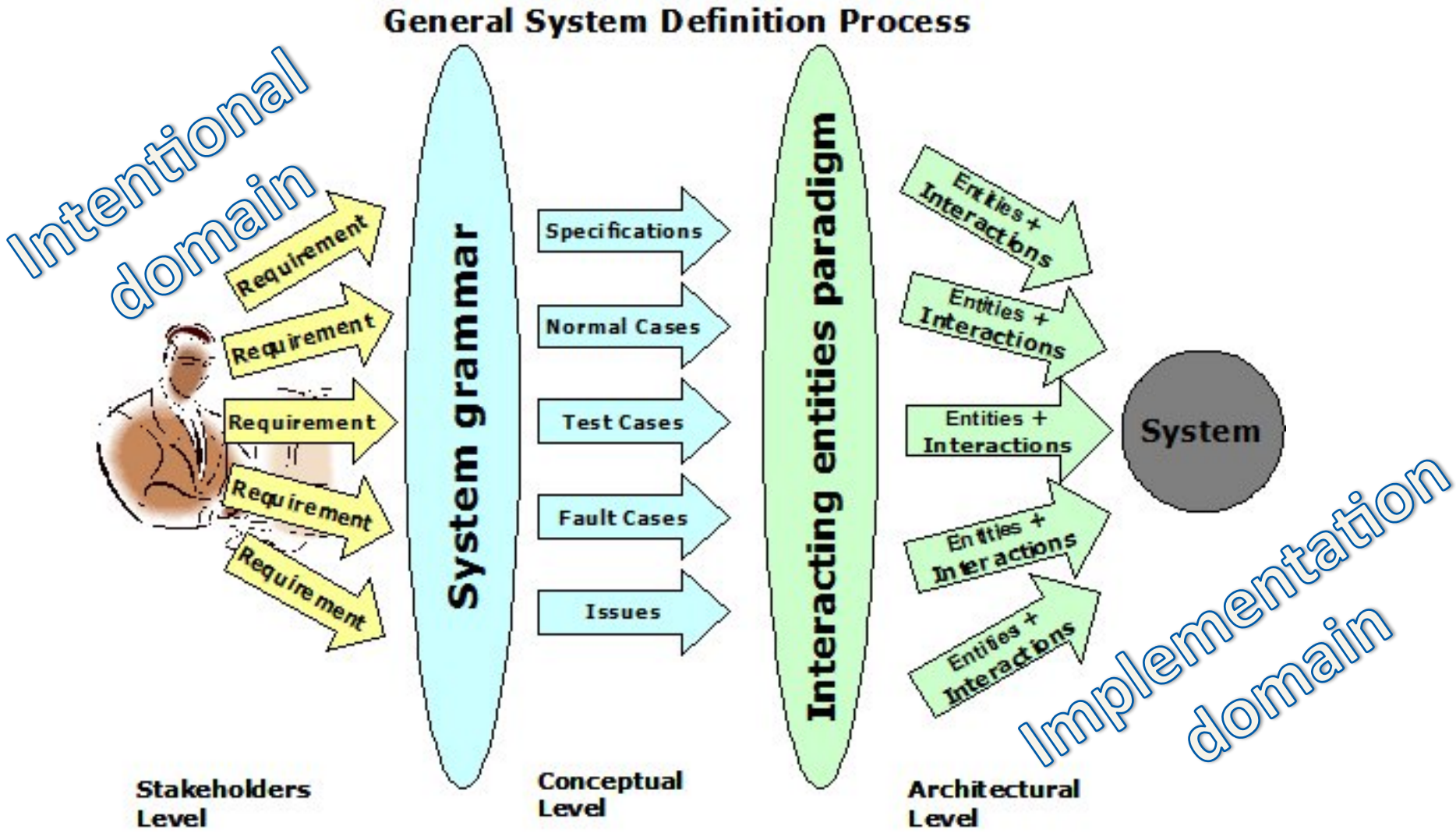
Why systems engineering?

- Systems are becoming very complex:
  - Discrete components (SW & HW): every state change is important (and can be fatal!): 10**n
  - More and more dynamic/adaptive
  - Multi-domain (mech, chem, elec, SW, …)
- Craft vs. engineering:
  - Managed development vs skills alone
  - Trust requires evidence and traceability
  - Human are creative, but error-prone
- Cost-efficiency: also for small companies?

Why (safety) standards?

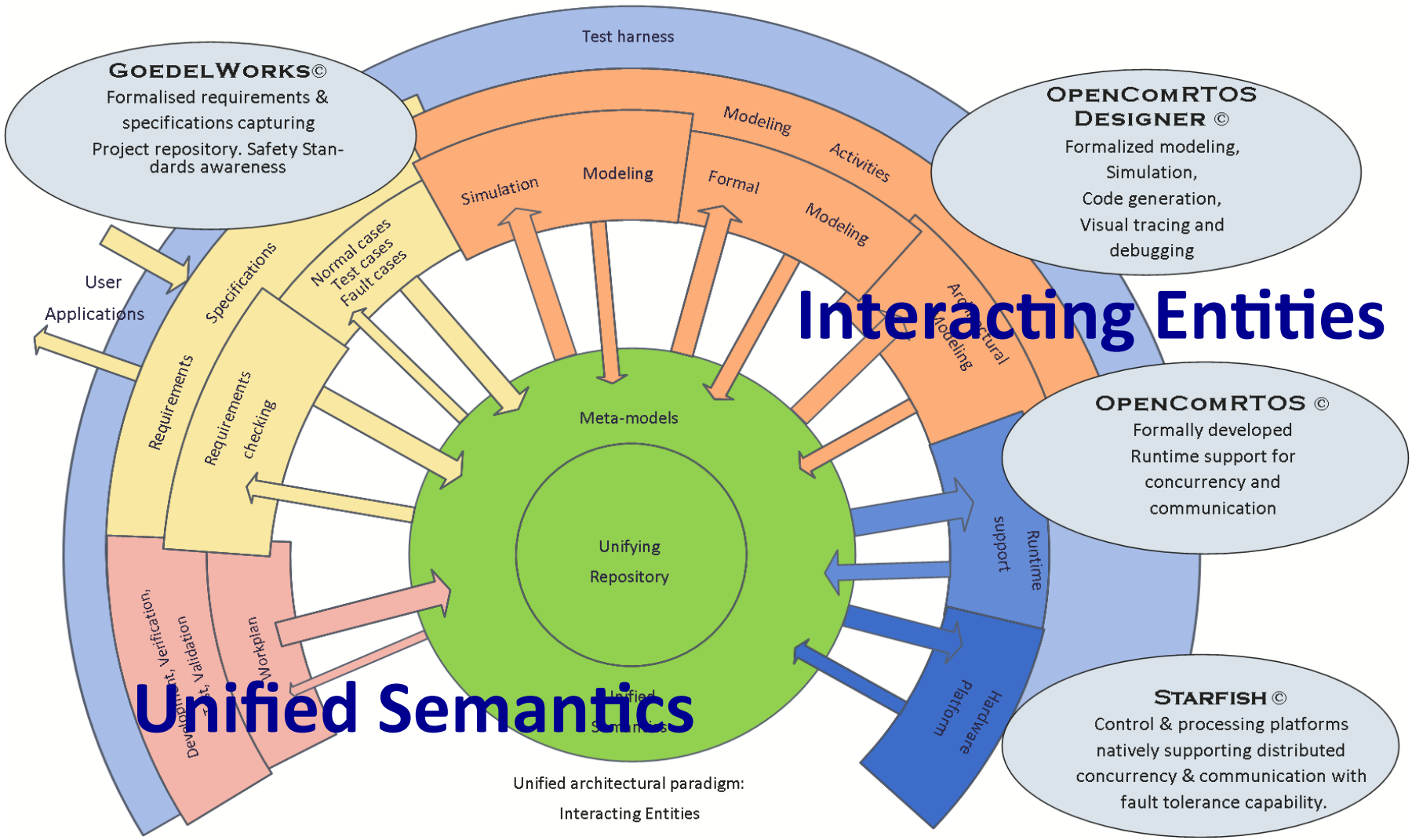
- Body of knowledge and proven practices
- Safety critical = mission critical = trustworthy
- Reduce cost of certifiable engineering
- Reduce risk of system failure
- Attempt to be normative
- **Predictability reflects our control of the engineering process.**
- **Automate if possible**

SE mapping process



Altreonic's methodology

A coherent approach to systems and safety engineering

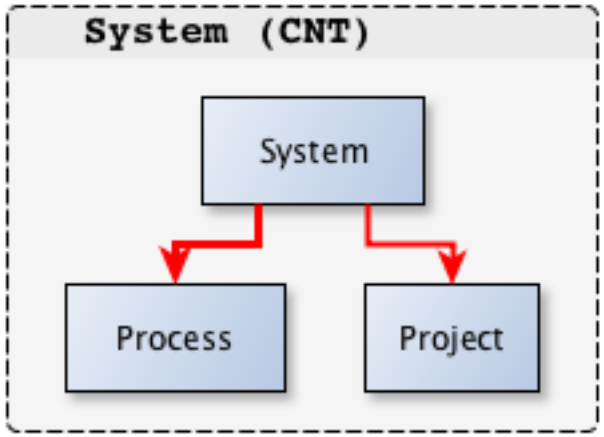


Unifying metamodel in GW 2.5

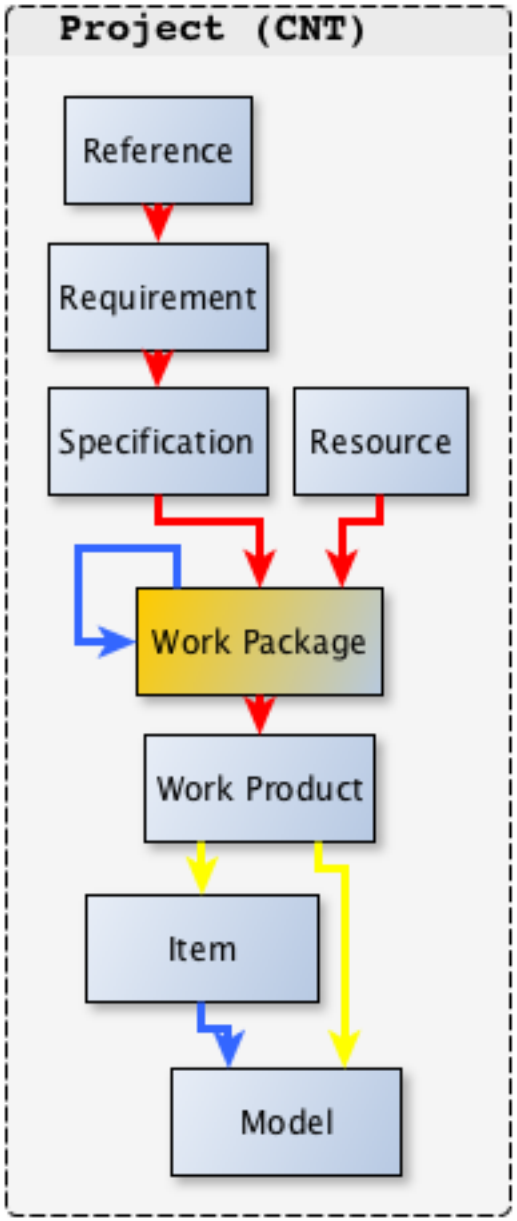
- Model is generic for all engineering domains
- Bias towards embedded systems:
  - Software
  - Hardware
  - Mechatronics
- Simple yet complete
- Customisation for each organisation:
  - Merge organisational process flow with standard's process requirement

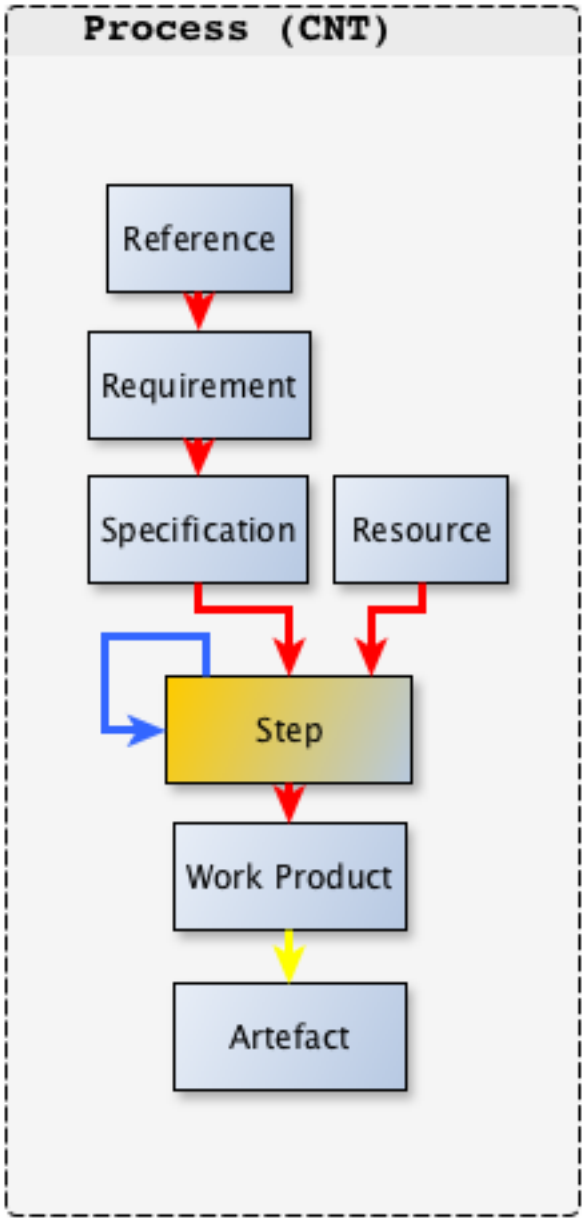
Orthogonal concepts (1)

Reference	Any relevant information, external and generic, but of potential importance for the Project: datasheet, standard, paper, …
Requirement	Any statement by any stakeholder about the system to be developed: technical as well as non-technical
Specification	A Requirement that by refinement and decomposition can be tested and verified. (requires: Test Case)
Work Package (implements Process Step)	A collection of coherent and planned Activities that result in the availability of an Item or Work Product that meets the Specifications. "Develop the right things" (what), "Develop it right" (how)
Resource	An Item or Work Product needed in a Work Package in order to execute the Activities in a Work Package or Step.
Work Product: Models and Items. + Artefact	The result of a Work Package. An Artefact is the supporting evidence.



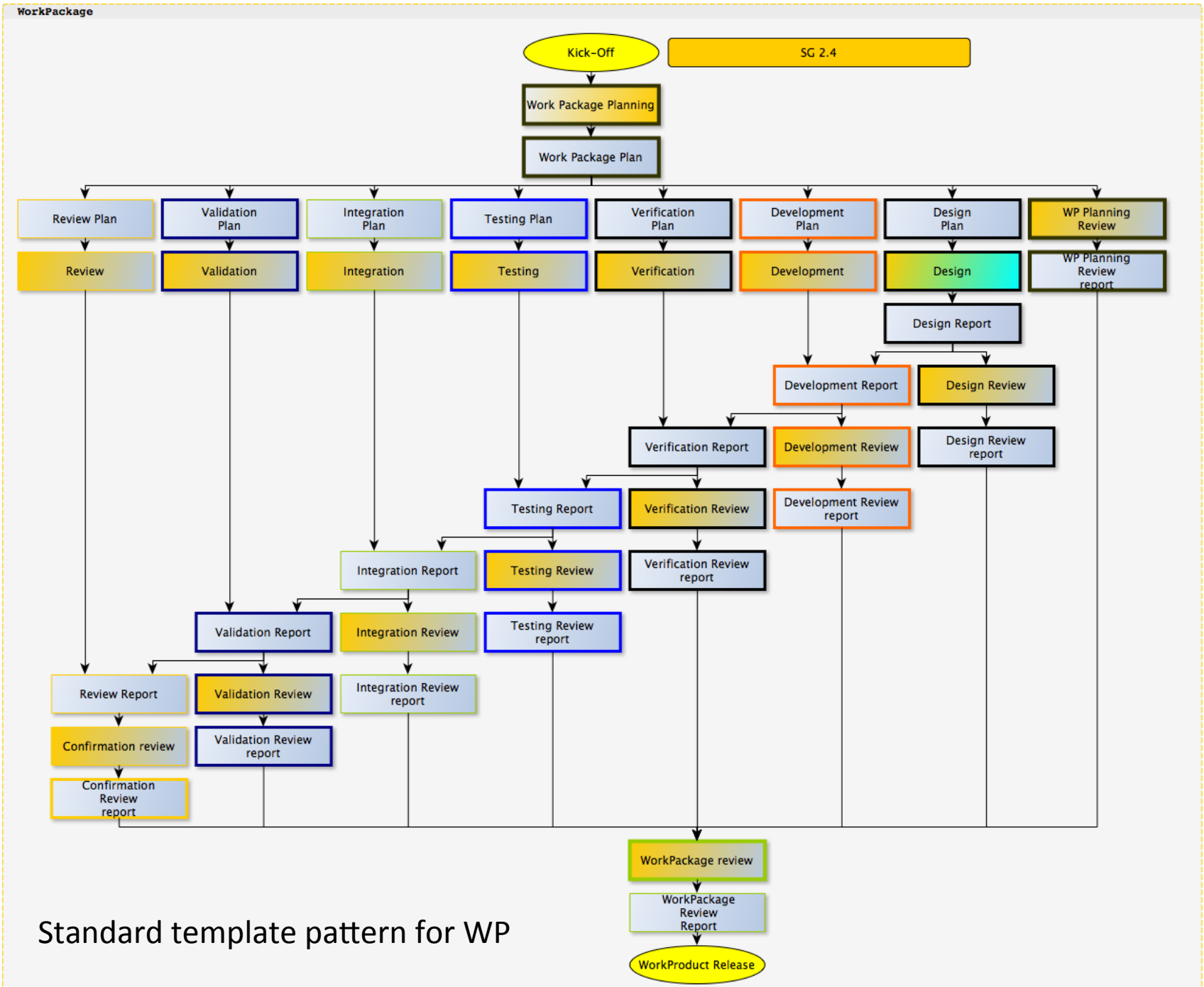
A system is the result of a **Project** (development) executed by following a (prescribed) **Process**





Work Package pattern as template

8 Work Package Activities	
Planning Design Development Verification	Testing Integration Validation Review
4 phases each	
Planning – Doing – Document - Confirmation	



Standard template pattern for WP

Orthogonal concepts (2)

Planning	Describe how an Activity will be performed. Includes what? How? When? Where? With what? ….
Design	Specify the architecture (incl. interfaces)
Development	The actual activity that takes all inputs and develops a concrete Item instance that fulfills the Specifications
Verification	Verifying that the Development was done according to the Process Specifications. “Was the work done as it should have been?”
Testing	Verifying that the Item meets its Specifications (execute Test Case)
Integration	Assemble the Items into a system or subsystem component.
Validation	Verify that the Integrated Items meet the Requirements and Specifications as a whole.
Review	Double check (using independent people)

Orthogonal concepts (3)

For each Activity:

Planning	Describe how an Activity will be performed. Includes what? How? When? Where? With what? ….
Doing	The actual activity that takes all inputs and develops a concrete Item instance that fulfills the Specifications
Document	Leaving a “trace” (evidence) for later reference.
Confirmation	Independent Review

Hence: iterative, double check at higher level

WP Internal Resources

For each WP and its Actvities:

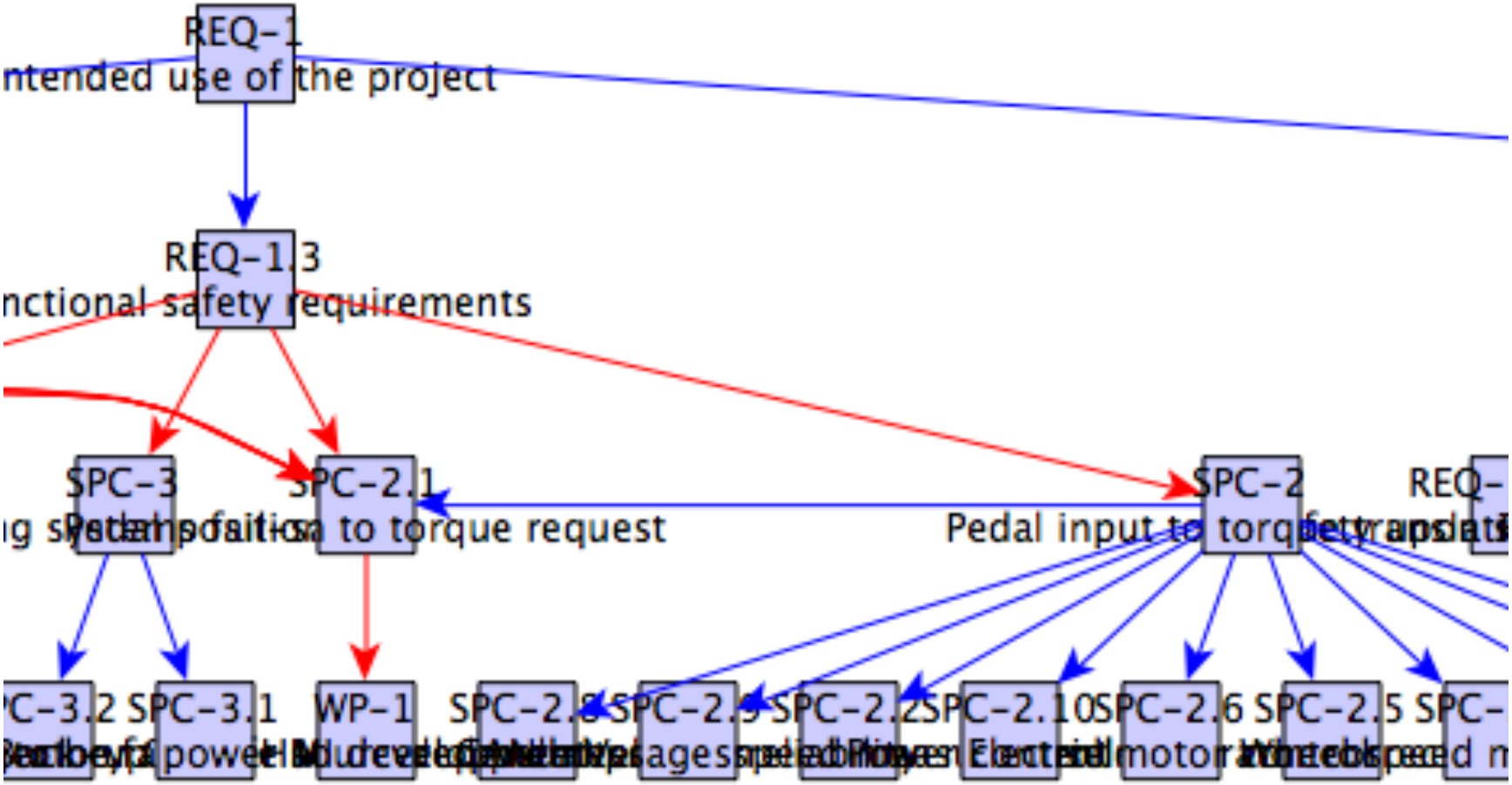
Internal Resource	Generic Resources (human, equipment, tool, financial) assigned to a WP Activity. Typically at Kick-Off.
Person has capabilities	Specified and verified/tested capabilities of a person to fulfill a specified Role: e.g. Test Engineer, Team Leader, Safety Assessor
Person has Roles	RACI: Responsible, Accountable, Consulted, Informed
External Resource	Resource that is produced outside the WP, hence creating a dependency.

Human factor made explicit

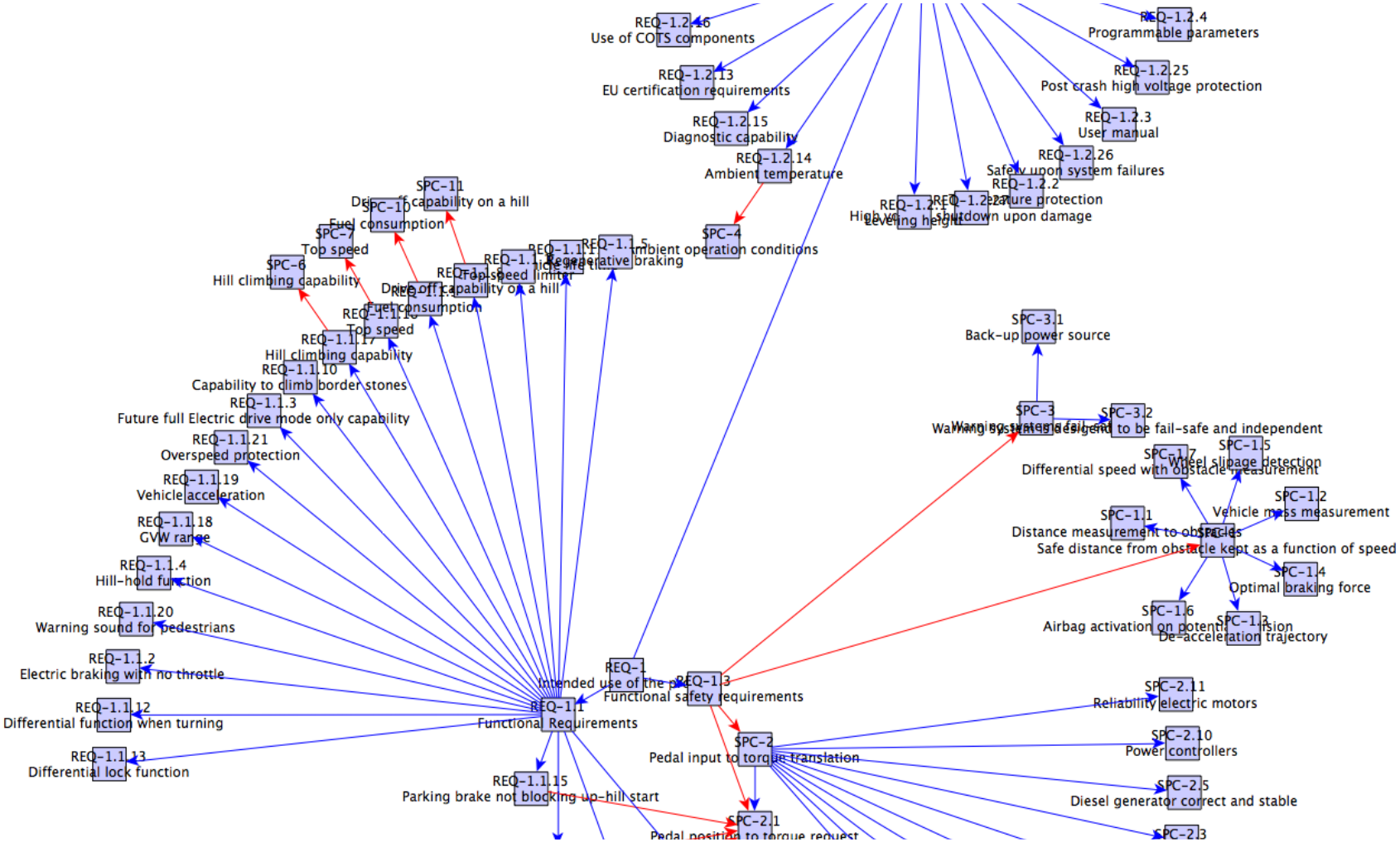
Traceability

- From any entity, dependency graph shows traceability and change impact
- Also shows incompleteness in repository (missing links, entities)
- Takes into account decomposition
- Activities can be concurrent/agile or not
- **Dependency tree used to approve entities in order of dependency**!

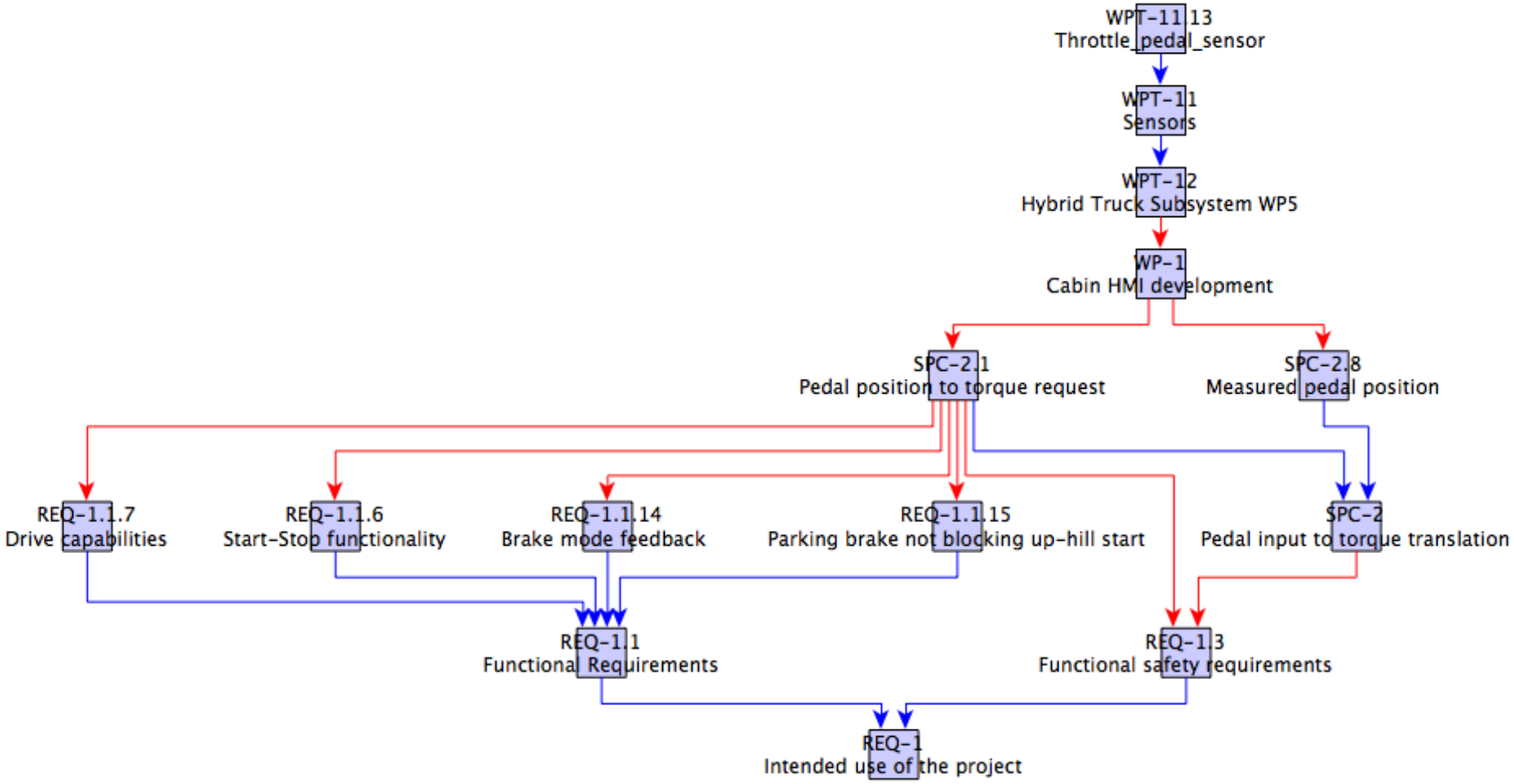
Extract from tree graph



After re-layout (see demo)



Partial trees



Other features

- Teamwork: entity locking, repository, user management
- Glossary
- “Links”: relationships between Entities reflect dependencies and decomposition
- Version management: logging of changes, undo, import/export, attaching documents, …
- Document generation: read-only snapshot
- GANTT chart: display planning

Validation of GoedelWorks

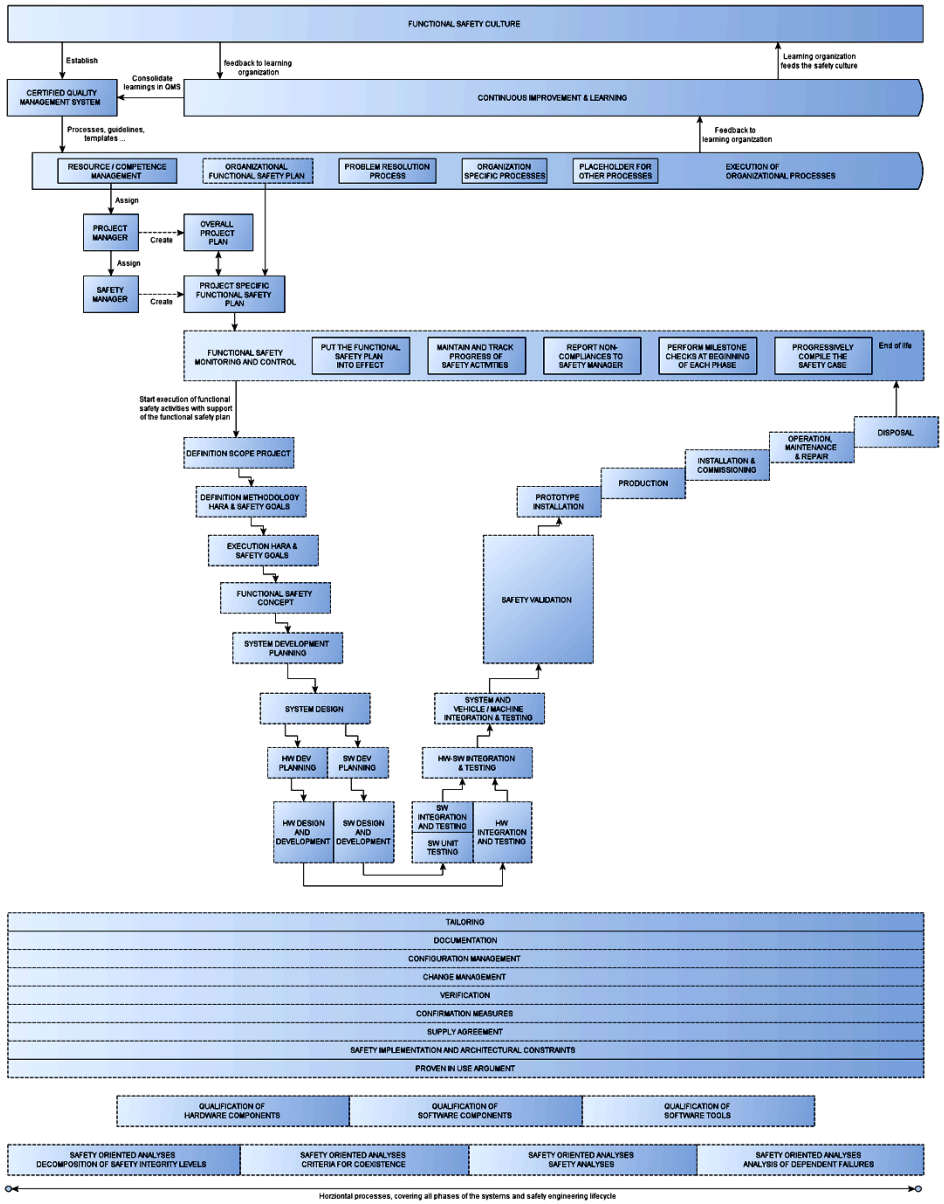
- Input: ASIL project of Flanders Drive
  - **A**utomotive **S**afety **I**ntegrity **L**evel
- Goal: develop common safety engineering process based on existing standards:
  - Automotive: off-highway, on-highway
  - Machinery
- IEC 61508, IEC 62061, ISO DIS 26262, ISO 13849, ISO DIS 25119 and ISO 15998
- 3500 Process Requirements, 355 STEPs, 100 WPTs,
- Other standards: customer specific

ASIL V-model

Organisational

Safety and Engineering/ Development

Supporting



ASIL: have the standard on-line

Word File Edit View Insert Format Font Tools Table Window Help — 100% Mon 23 Jun 11:20 Eric Verhulst

GoedelWorks | GoedelWorks | GoedelWorks

192.168.56.230/#entity=id-6840

Apps | GoedelWorks | Webmail - Emails - | Google | CADP Home page | TASTE | Model Driven Compa | Antro | GV-5-2014 | PDF Books | Reichelt | Aptina Imaging

Entities | Glossary | Query | Administration | Change Log

Entity Tree | Search Results

Find Entities

- System (192.168.56.230)
  - DELETED PROCESSES
  - DELETED PROJECTS (2)
  - PROCESSES (1)
    - PRO-1 (id-1): ASIL (5)
      - Deleted entities
      - Process Entities (8)
      - PRO-1.1 (id-6364): Supporting
      - PRO-1.2 (id-6362): Organisatio
      - PRO-1.3 (id-6363): System Safe
        - Deleted entities
        - Process Entities (8)
        - PRO-1.3.1 (id-6466): Define
        - PRO-1.3.2 (id-6572): Define
        - PRO-1.3.3 (id-6615): Execu
        - PRO-1.3.4 (id-6832): Functi
        - PRO-1.3.5 (id-6861): Syster
        - PRO-1.3.6 (id-6890): Syster
          - Deleted entities
          - Process Entities (8)
          - PRO-1.3.6.1 (id-6891): S
            - Deleted entities
            - Process Entities (8)
              - Process Referen
              - Process Resourc
              - Process Require
              - Process Specifica
              - Steps (5)
                - STP-1 (id-689
                - STP-2 (id-689

System | id-6840

STP-2 (id-6840): Create preliminary safety architecture (In Work) | Changelog | Comments | Dependency Tree | Precedence Tree

Edit Mode | Save | Access Control List | Change Navigation Parent

Name:	Create preliminary safety architecture	**Subtype:**	Process
Origin:	ASIL: 04_000_funcsafetyconcept_002_Create_PAA	**Start:**	23 June 2014 at 11:18 AM
Deadline:	23 June 2014 at 11:18 AM	**Responsible:**	Not defined
Accountable:	Not defined	**Consulted:**	Not defined
Informed:	Not defined		

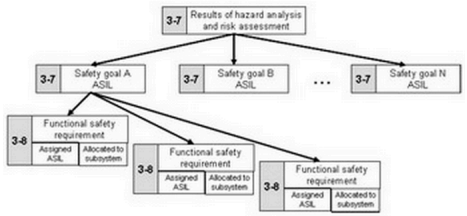
Kick-off

Description

Note that the safety architecture concept includes the redundancy and independence concept for the elements and can be given in block diagrams form. An analysis of dependent failures can be useful to check the independence concept.

Allocate the functional safety requirements

- The allocation of functional safety requirements shall be based on the elements of the preliminary architectural assumptions, functional concept, operation modes and system states, physical limits and constraints, fault tolerant time span and emergency operation for the system.
- In the course of allocation, the safety integrity level and the information given in "Derive functional safety requirements from safety goals" shall be inherited from the level above.



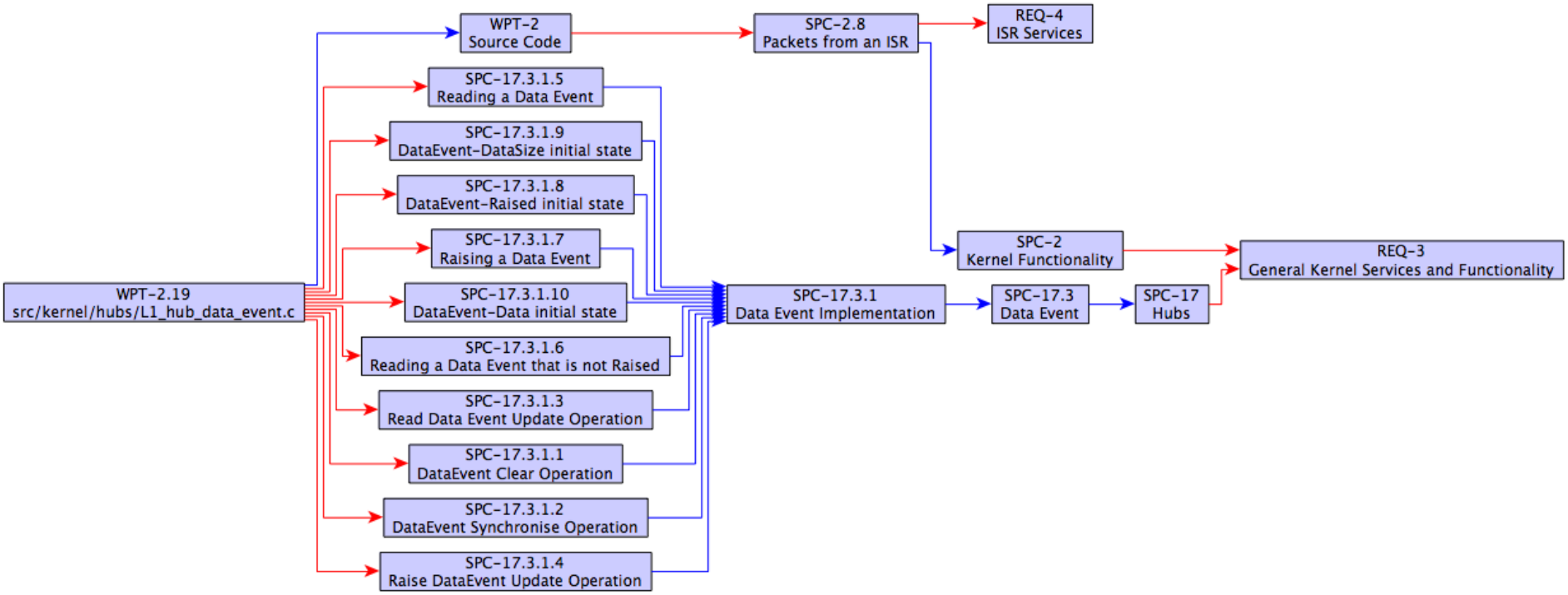
- If several functional safety requirements are allocated to the same architectural element, then the architectural element shall be developed in accordance with the

Logged in as: eric.verhulst Logout

PRJ-1: OpenComRTOS Qualification Pack

- Formally developed network-centric RTOS
- Scope: Kernel + PowerPC-e600 HAL only
- LOC of kernel: 6550 lines of C and Assembly.
- Number of Entities in Project: 1280
  - Specifications: 307
- Number of Links in Project: 2840
- Number of tests: 337
  - Unit Tests: 175
  - Functional Tests: 162 (covering 99.8% of all lines, and 99.8% of all branches, for SP)
- Number of reports: 29
- PDF output: 2043 pages (excl. source test projects).

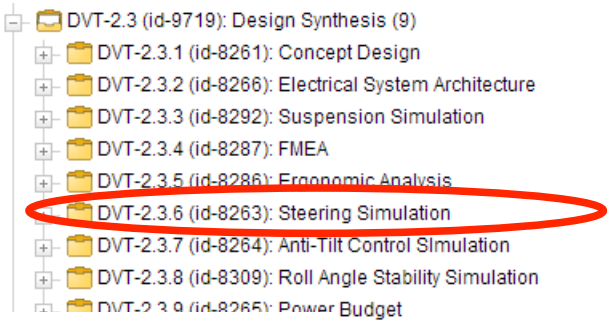
Source to REQ precedence tree

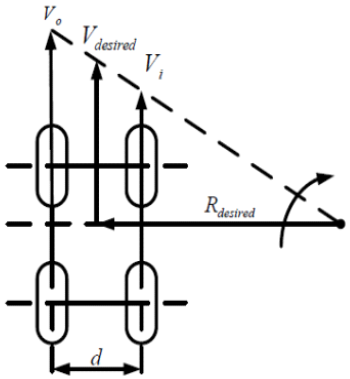


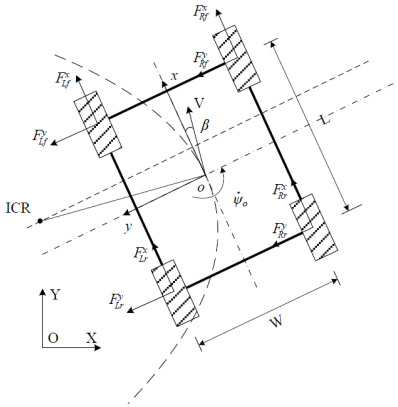
Impact analysis at the push of a button

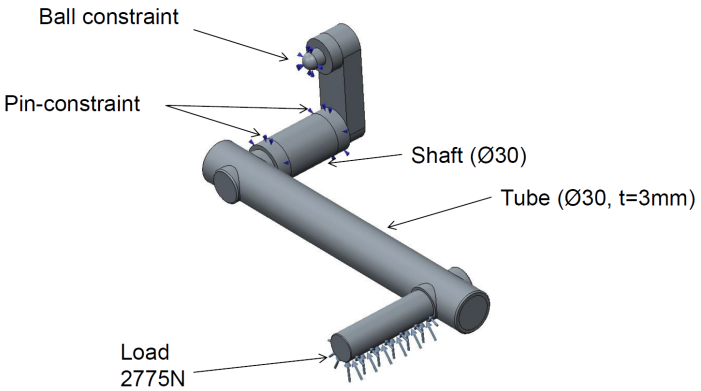
PRJ-2: KURT: Altreonic e-mobility vehicle

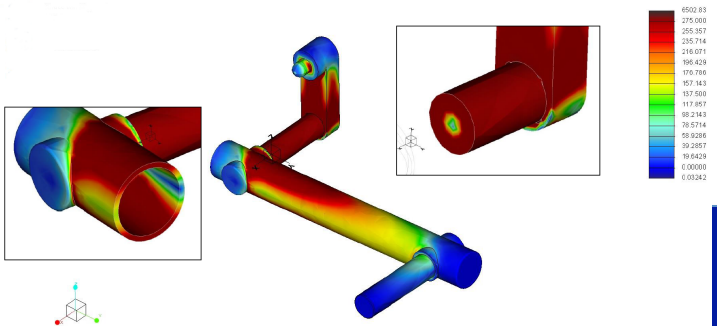
- Behavior emerges when entities interact

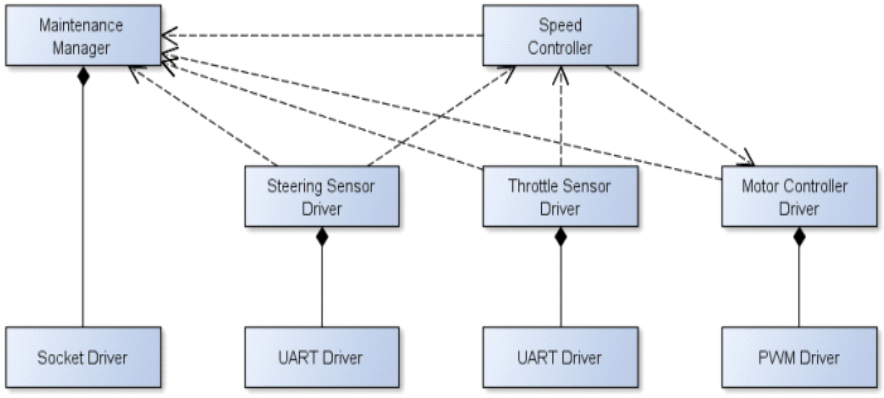






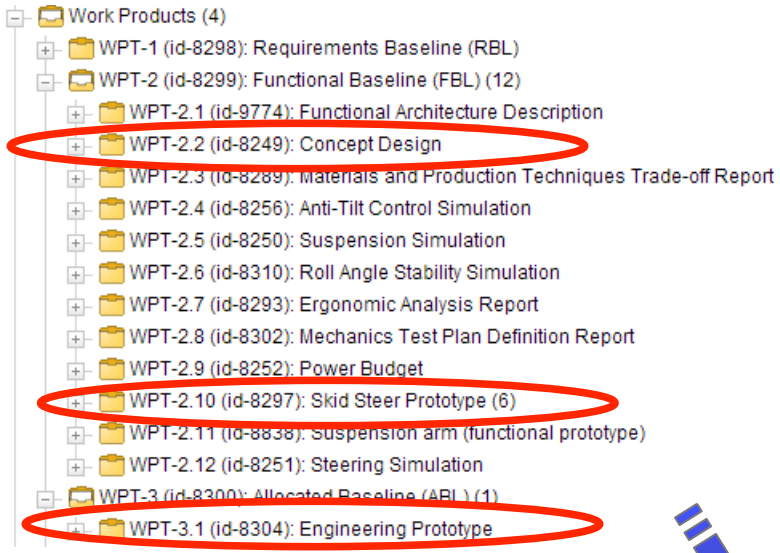






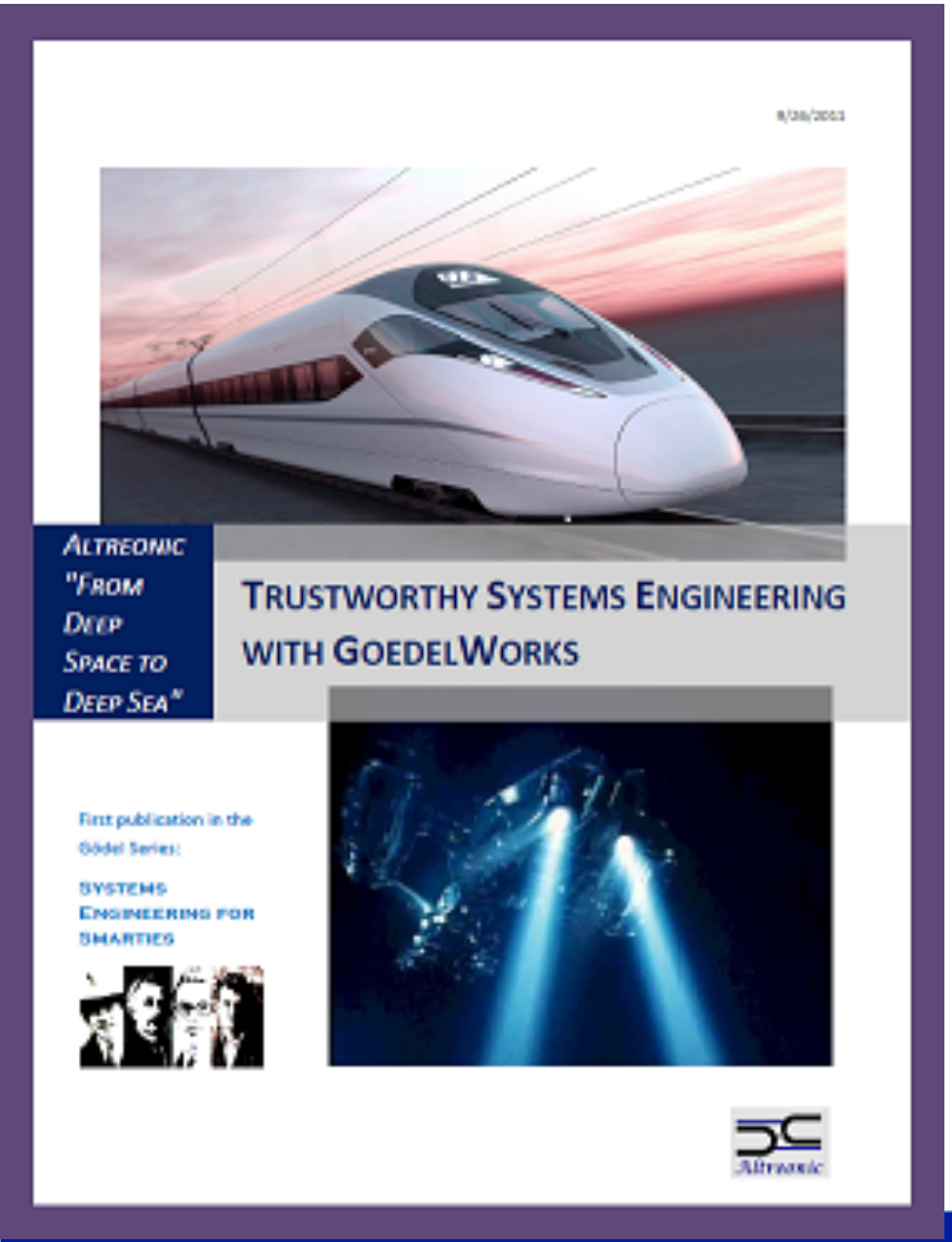
Sharing data with design partners

- Effectively communicate relationship between requirements, elements and functions



Summary

- An integrated systems engineering approach for trustworthy systems doesn't need to be complex, but complete
- Formalized when possible, formal when needed
- Accessible and cost-efficient
- Supports any size of teams



More info at www.altreonic.com

http://www.altreonic.com/sites/default/files/Systems%20Engineering%20with%20GoedelWorks.pdf

(will be updated for GW v.2.0)