DOPAS

# DOPAS
# Training Workshop 2015

## Requirements Management
### General Introduction

Marjatta Palmu, Posiva Oy

14 September 2015

D1 1.2.2

*The research leading to these results has received funding from the European Union's European Atomic Energy Community's (Euratom) Seventh Framework Programme FP7/2007-2013, under Grant Agreement No. 323273 for the DOPAS project.*

update 2016

# Sources of Requirements

- Highest level requirements are derived from policy decisions and strategies, legislation and other regulations, owners, other major stakeholders
- System requirements are derived from safety objectives to be fulfilled and related safety functions of subsystems and components
- Design requirements are also derived from standards and rules, and from industry conventions
- Forming statements about what a system, component (design of a component) has to do (shall...), how does it need to perform, and what it must be like (its characteristics e.g. not harmful...), what type of conditions it needs to tolerate, what it cannot be?

Source: adopted from DOPAS D2.4

# Expected Learning Outcomes from Topic 1.2

- Identify and list major sources of requirements for geological disposal and for closure
- Understand and be able to describe the major elements of the general concept of requirements management (various elements of it) and its objectives orally or in written form/figures
- Discuss the collection of identified requirements and their different hierarchy
- Understand how requirements are applied in the design of plugs and seals using iteration cycles in interaction with compliance management

# List of contents (1)

- What is a requirement, what makes a requirement?
- How to write and interpret requirements;
- What does one do after requirements have been identified?
- Models for requirements management – describing the different levels of the requirements, their content and source/links
- Setting a baseline for requirements and managing change
- Tools to assist in the management; Related concepts, and short examples

- How does one translate identified requirements into practical designs and solutions? Some case examples and DOPAS experiment examples to follow in today's other presentations.

# What is a requirement?

- an objective or a need of an end-user
- expressed in general with the verb “shall”
- requirement itself is not the solution to the objective or need

# What makes a requirement?

- a single statement with defined attributes,
  - an absolute requirement (applies always), “shall” verb in the statement
  - target (requirement sets a target, but can be optimised or negotiated)
  - expectation or expression of need (requires modification into a requirement)
- preferably numerical
- requirements can further be
  - functional
  - non-functional requirements
  - constraints/boundary conditions
- have different priorities in relation to each other

# Writing requirements

**Requirements need to be**

- correct (use a competent team)
- consistent (cross check)
- complete (need has to be covered)
- realistic (technical feasibility)
- necessary
- verifiable
- traceable to source (change management and updates)
- can be prioritized in connection with other requirements

**Expressing priorities**

- shall (absolute)
- should (iterate, negotiate)
- nice to have (optimisation)
- may

# Requirement attributes (examples)

| | |
|---|---|
| **Source** | Who has placed the requirement, origin? |
| **Lifecycle stage** | In which lifecycle stage does the requirement apply |
| **Justification** | Reason, why this requirement is needed |
| **Priority** | In relation to other requirements (e.g. classification) |
| **Urgency** | At which stage of the system design or engineering is the requirement based information (when is it needed for the work) |
| **Verifiable** | Can the compliance to the requirement be tested, verified or validated (important to consider when writing a requirement) |
| **Approved** | Has the requirement been approved as part of the design basis |
| **Inspected** | Inspection status |
| **Value** | Target/validation value |
| **Range of values** | Acceptable range of values or tolerances for a numerical requirement |
| **Safety** | Is the requirement safety or production critical? |
| **Other comments** | Other necessary complementary comments |
| **Open issues** | Open questions prior a requirement can be accepted |

# Overall disposal <u>system</u> objectives or requirements

- **<u>Safety</u> and <u>robustness</u> of system**
  - The disposal system has to ensure that the waste is secure and that human beings and the environment are protected from the effects of radiation for the time period of about one million years during which the wastes (especially spent fuel) pose unusual hazard
  - Robust: Performance may not be unduly affected by residual uncertainties from realistic future scenarios regarding its evolution …
- **<u>Reduction</u> of <u>likelihood</u> and <u>consequences of human intrusion</u>**
  - Measures should be taken to minimize the risk of human intrusion. Should intrusion nevertheless occur, the repository should be designed in such a manner that degradation of performance after intrusion is limited.

=> Safety functions of disposal system = functional objectives with key relevance to long-term safety and security (to comply with the requirements).

Source: adapted from Nagra 2005. NTB-05-02, p. 34

# What to do after requirements are identified?

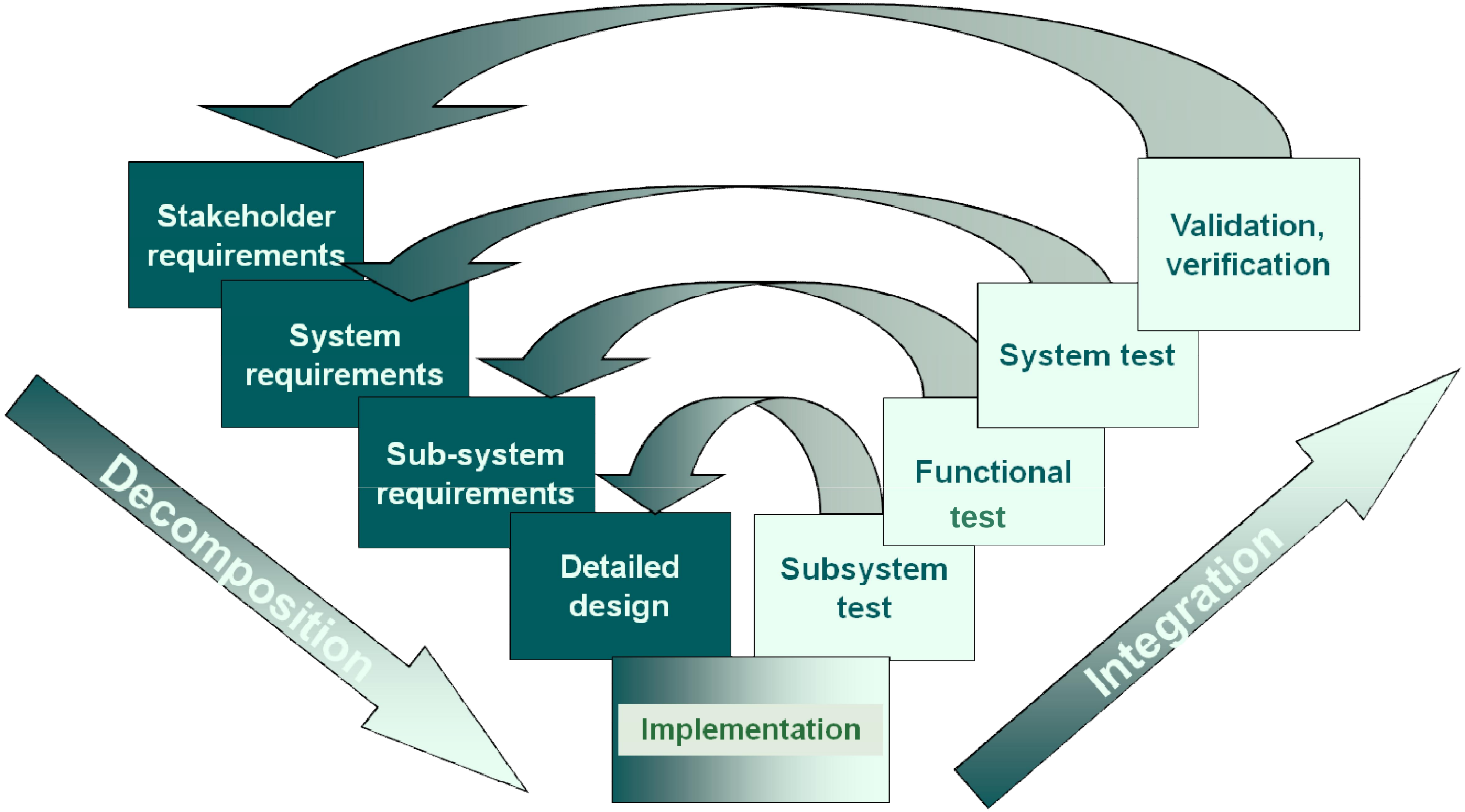
- Requirements themselves <u>form a complex information structure</u>, that increases in complexity as the disposal project advances to specification level.
- Within this system the number of relationships increases and adds further to complexity and knowledge management challenges.
- Simultaneously the requirement changes need to be managed and the status of the requirements updated according to each project stage.
- Requirements change in an iterative manner (iteration cycle)
- As a starting point for managing requirements (i.e. setting up a requirements management system (RMS)) <u>the first baseline of requirements needs to be set. The changes are compared against this baseline and traced</u> with the assistance of a requirements management tool in most cases.

# Requirements Management (RM)

- a way of including the customer’s voice into the design process by
  - stating what a system is supposed to do
  - instead of how it is supposed to do it.
- according to Hoffman & al. (2004) :

  ”**Requirements management** is the structuring and administration of information from elicitation, derivation, analysis, coordination, versioning and tracking requirements during the complete product lifecycle”
- The origin of requirements management is in Systems Engineering
- An alternative concept meaning almost the same ”Configuration Management”

# A system of requirements – V-model




A hierarchical system to link higher level requirements into lower level requirements for operationalization => designing functionality to meet the identified needs with **practical solutions from alternative options**, and verifying these against the set (of) requirements

*Example source: Posiva*

# Requirements

- make a hierarchy of increasing detail when moving from the top level requirements to the component level specifications.
- They are developed in an iterative manner and
- intended to ensure traceability and control the impact of changes.
- They shall/ cannot be in conflict with each others (links between requirements need to be identified).
- The source/s of a requirement needs to be identified in a transparent, traceable manner (especially the underlying assumptions) – e.g. numbering of requirements, level of requirement, and further attributes of each requirement.

Source: adopted from DOPAS D2.4

# Requirements Management (RM)

- In the disposal system's requirements management
  - approval,
  - inspection,
  - prioritization and
  - verification of compliance of solutions to requirements

  make a crucial part of the RM system.
- Requirement attributes are used for the purpose as part of the Requirement Management Tools. RM Tools are software databases assisting in managing the complex requirement infrastructure.

# The hierarchy levels in Posiva's V-model (VAHA)

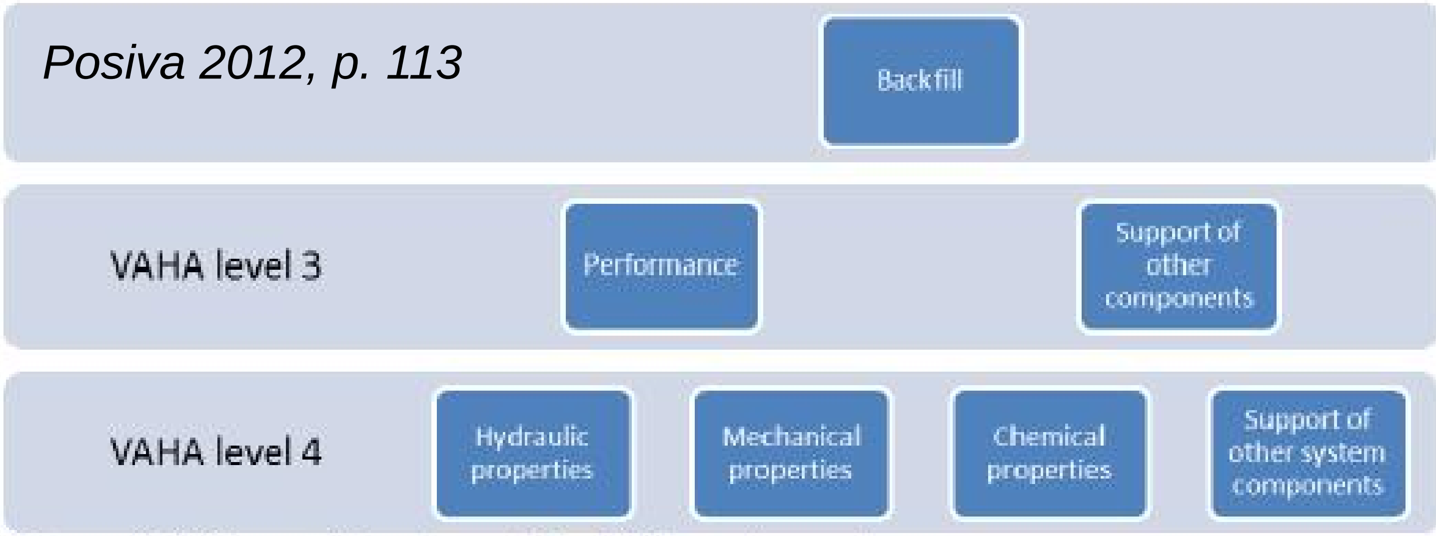


*Figure 9-1. General structure of backfill requirements.*

- Stakeholder requirements (Level 1) [SHR]
- System requirement (Level 2)
- Subsystem requirement (Level 3) – Backfill (incl. plug)
- Design requirement (Level 4)
- Design specification (Level 5)

*Source: POSIVA 2012-03, pp.113-118*

# A set of examples contributing to Level 3 performance targets (incomplete set)

- Hydraulic isolation (Level 4)
  - *The plugs shall isolate the deposition tunnels hydraulically during the operational phase of the repository.*
  - *solution?*
- EBS compatibility
  - *The chemical composition of the backfill and plugs shall not jeopardise the performance of [other barriers] the buffer, canister or bedrock.*
  - *solution?*
- Ability of plugs to keep backfill in place
  - *The plugs shall keep the backfill in place during the operational phase.*
  - *solution?*

*Source: POSIVA 2012-03, pp.113-118*

# RM Tools to manage requirements

**Complex information structure**

# Suitable software (RM tool) is necessary (1)

- The work on draft documenting and managing the requirements often starts with e.g. MS Excel, but for keeping track of all requirements and their links and managing changes and their impact, a requirements database is most useful.
- Tool is needed because in the development process, the requirements are becoming more detailed at the lower requirements levels and finally when translated into specifications their relations become more complex and numerous. The requirements also change and the requirements need to be handled, their status updated and traced. (e.g. Hoffmann & al. 2004).
- In a recent project, the Finnish Radiation and Nuclear Safety Authority STUK is developing a RMS to include the current Finnish Nuclear Safety Guides (45 YVL-guides in total):
  – Related Data Volume: approximately **9500 requirements with 19 attributes** including defined value/s for each of these requirements!
  – This volume represents only the regulatory requirements!

# Suitable software (RM tool) is necessary (2)

- Comparisons between databases/tools are available on the web, popular software includes e.g. comprehensive enterprise software like SAP or dedicated software like DOORS (IBM).
- Custom-made Access® or other database based software also exist.
- One link to the current software listing is available at http://www.capterra.com/requirements-management-software/

# Related concepts to systems engineering and RMS

- **Configuration management** (overall architecture of the disposal system for design) a process including verification and validation of selected design bases and designs. Looks at the V-model over the whole project's lifetime.
- **Functional analysis** – also originating from systems engineering (Andra's presentation)
- **Requirements engineering** – related to RM
- **Quality Function Deployment (QFD)** – "House of Quality" for customer driven product development – use of ratings to come up with the optimum solutions in terms of conformance, performance and image. Developed by Japanese Y. Akao since 1960s.

# Some V-model application examples

**Canister quality assurance**

**Code verifications**

# It's all about iteration

NAGRA NTB 02-05 10

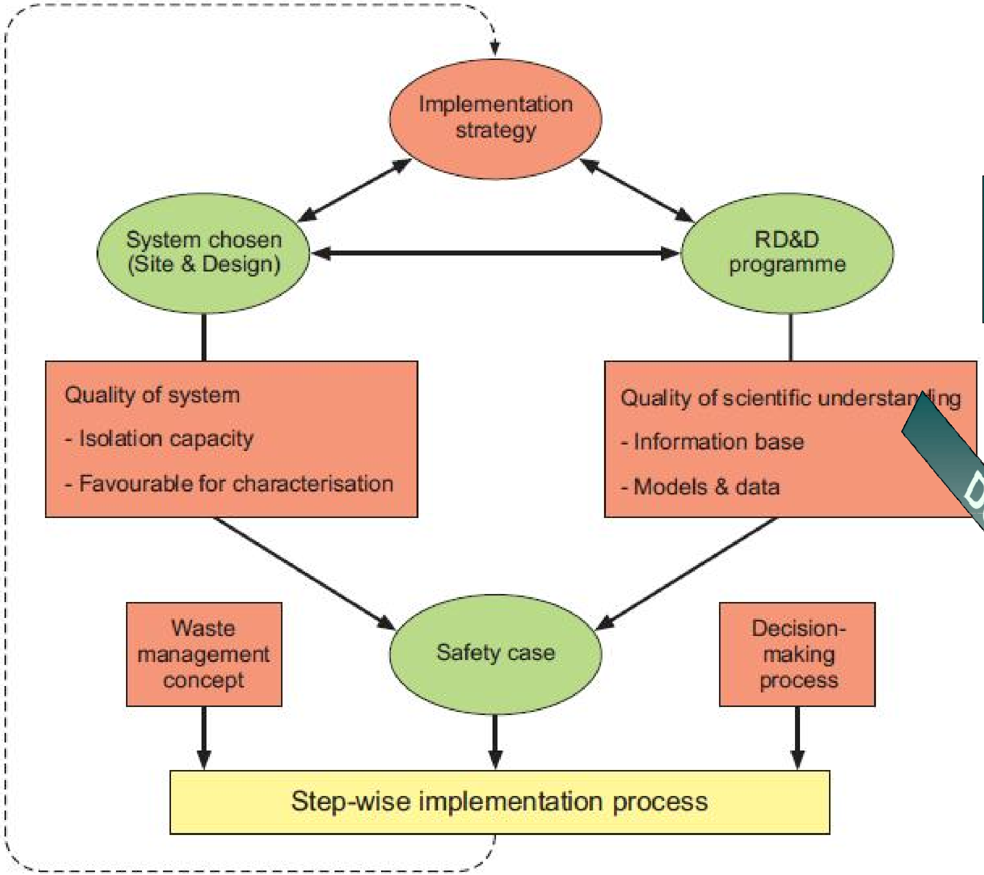
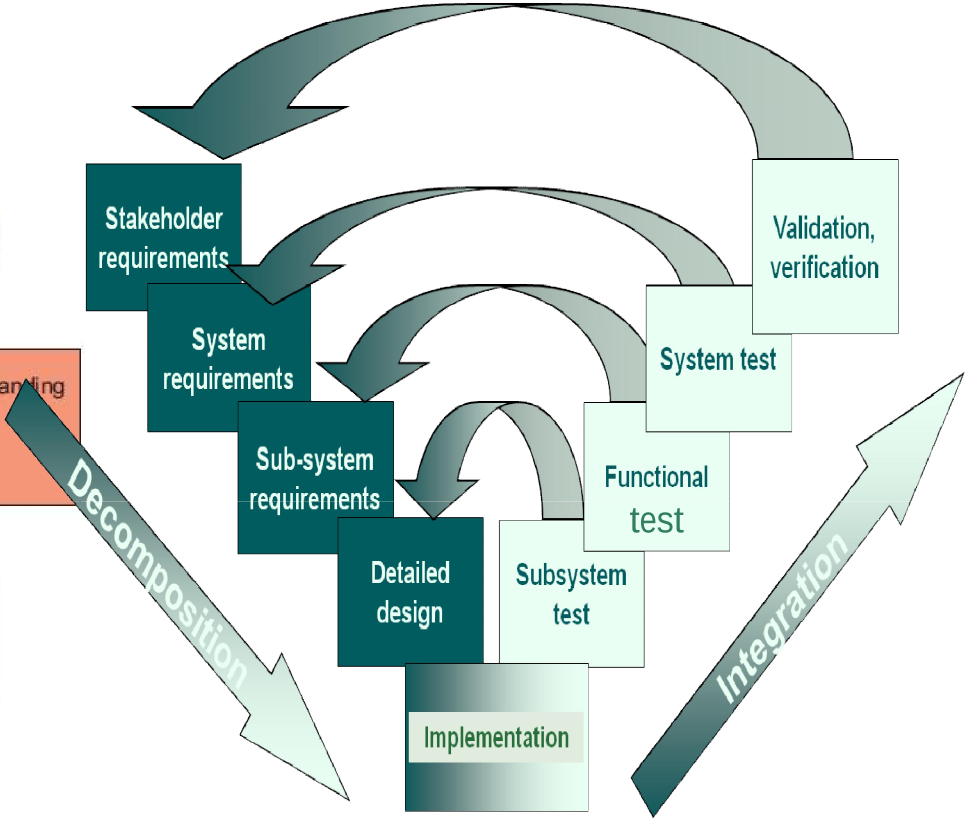



Fig. 1.2-3: Important elements of decision-making for the step-wise and iterative repository implementation process

© Nagra

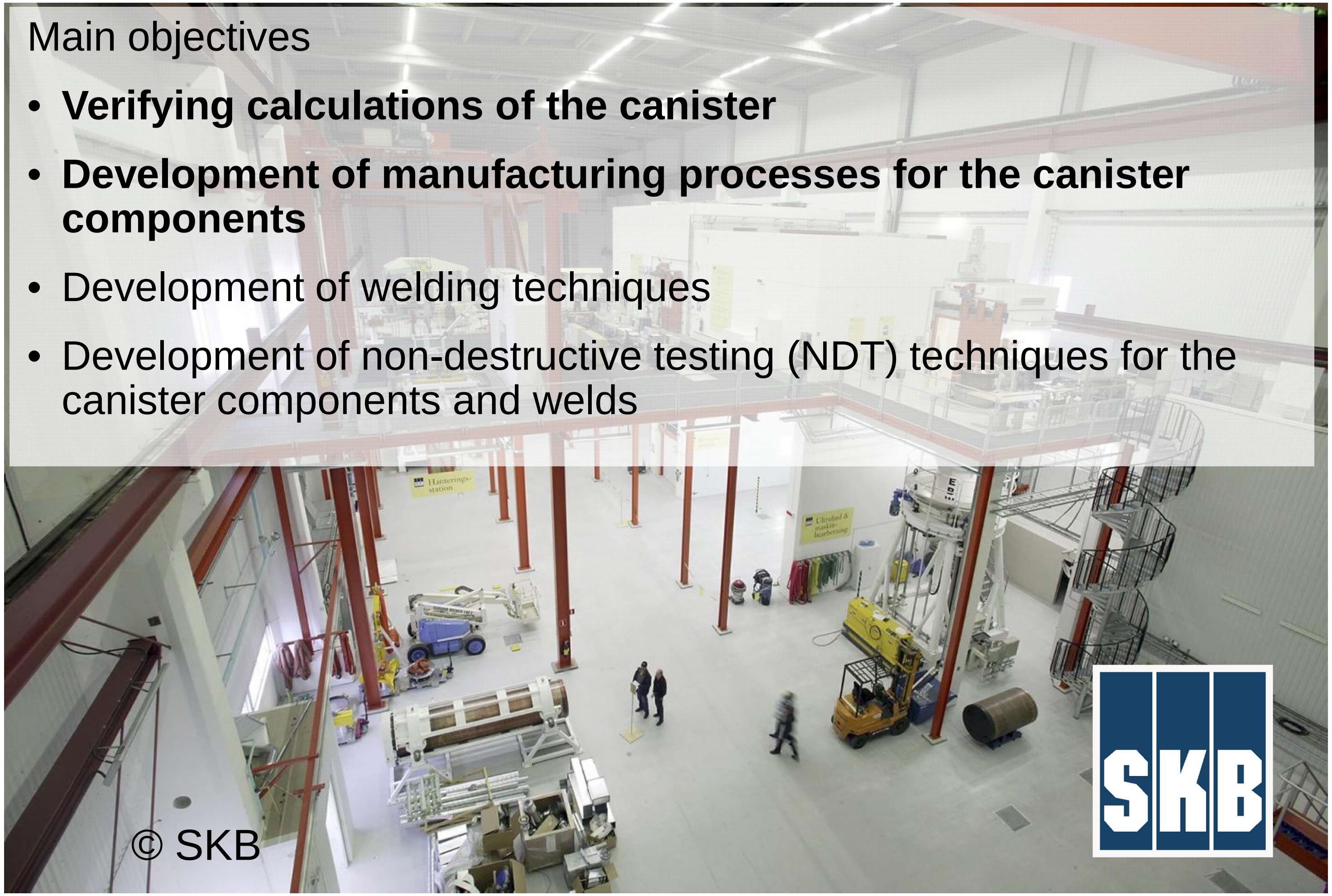



Source: Posiva

# Examples from the SKB's Canister laboratory

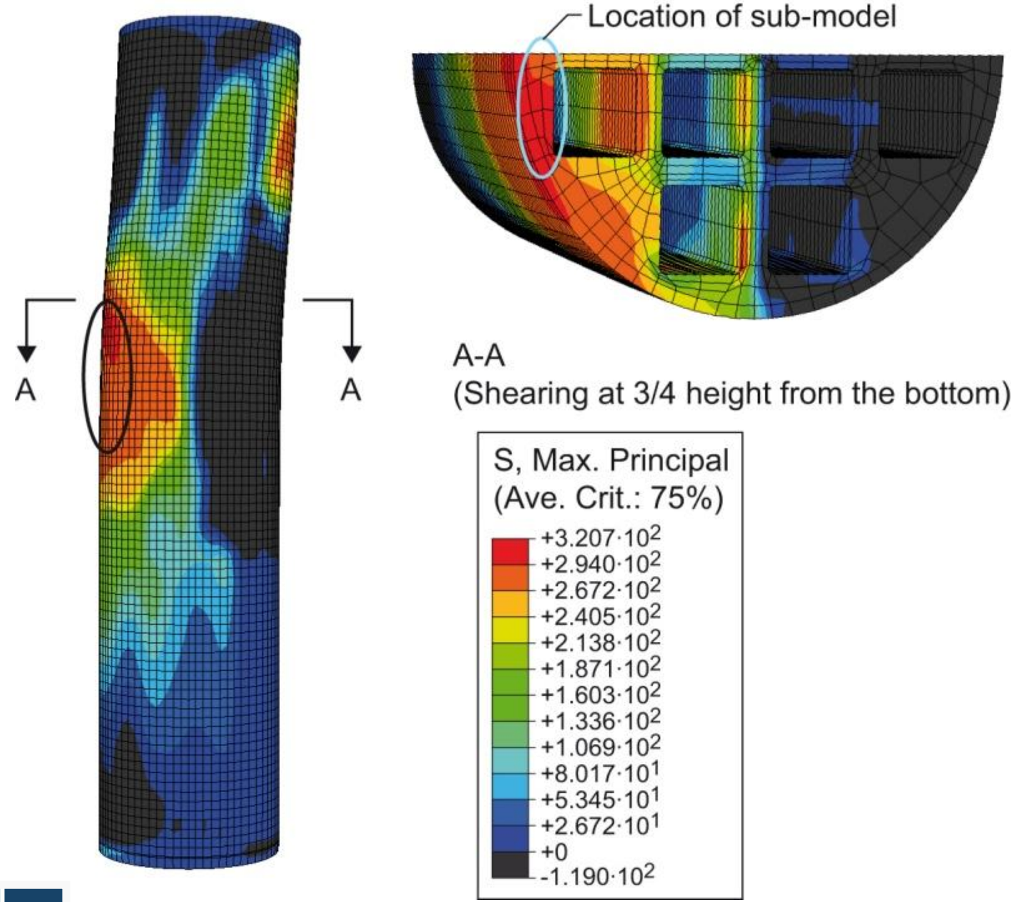
Main objectives

- **Verifying calculations of the canister**
- **Development of manufacturing processes for the canister components**
- Development of welding techniques
- Development of non-destructive testing (NDT) techniques for the canister components and welds



# Verifying calculations of the canister

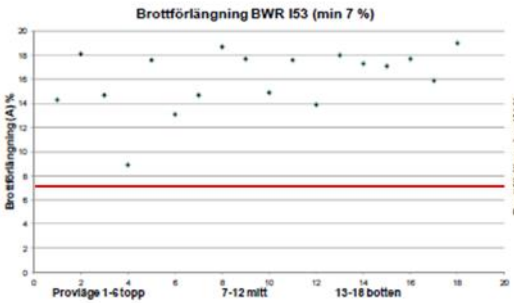
- **Shear load case = highest demands / functional demands/ requirements/**
  - **Global load analyses**
  - **Local load analyses**
  - **Highest strains close to the surface**
  - **Give high demands on acceptable defects**

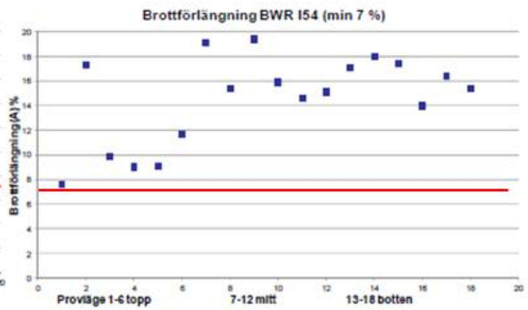


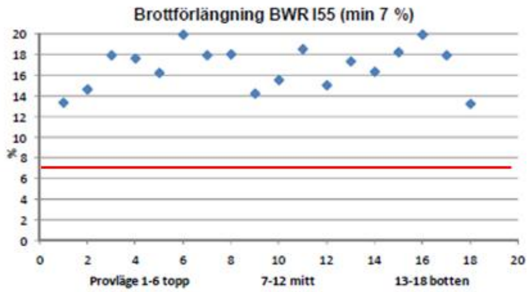

A means of verifying compliance with the requirements

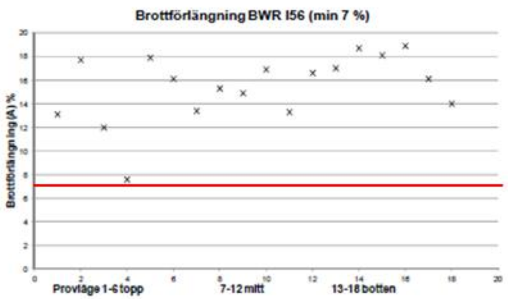
# Elongation of insert material – Process quality improvement

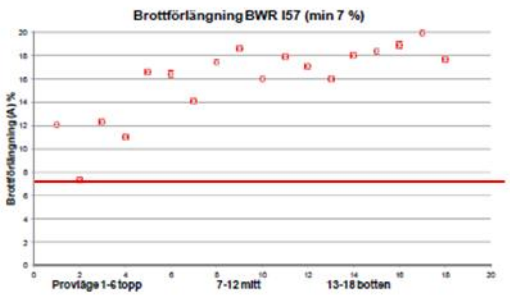
- Improve the casting process for minimizing deviations of mechanical properties in insert – improved process control increases confidence in sampling results

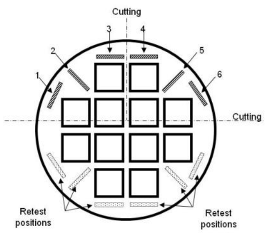


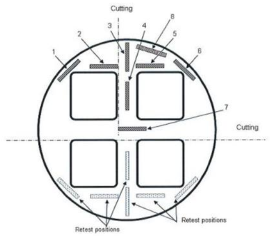



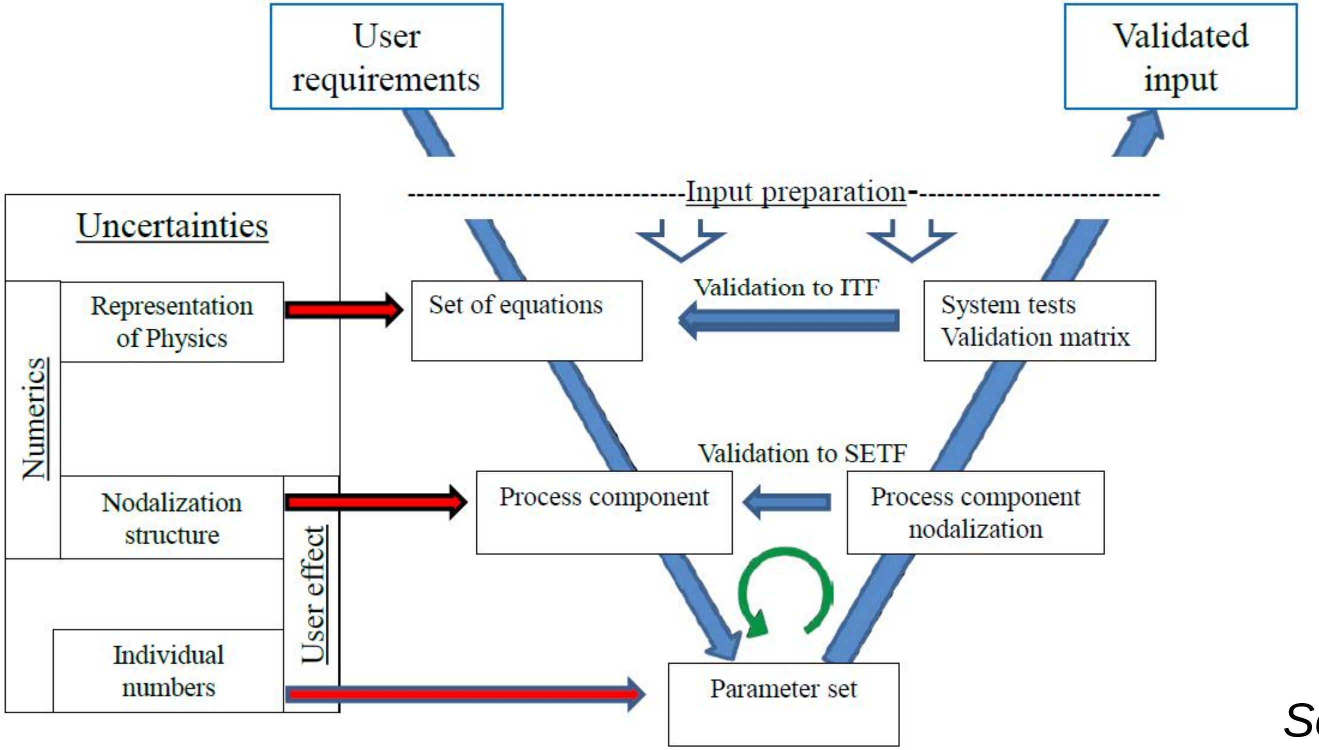



# A recent dissertation: VVER-440 Thermal Hydraulics as a Computer Code Validation Challenge

## Novel Method for Code Input Validation, V-model approach




ITF = Integral Test Facility
SETF = Separate Effect Test Facility

*Source: J.Vihavainen 2014, Fig.14, p. 51*

# References


- Hoffmann, M. et al. 2004. Requirements for Requirement Management Tools. Proceeding so fthe 12th IEEE International Requirements Engineering Conference (RE’04).
- White M. & Dodou S. (eds.) 2015. Deliverable D2.4: WP2 Final Report Design Basis for DOPAS Plugs and Seals. DOPAS project. (*to be published*)
- Nagra. 2002. Project Opalinus Clay. Safety Report. Demonstration of disposal feasibility for spent fuel, vitrified high-level waste and other long-lived intermediate-level waste Nagra Technical Report NTB 05-02.
- Posiva. 2012. Safety Case for the Disposal of Spent Nuclear Fuel at Olkiluoto. - Design Basis 2012. POSIVA 2012-03. (pp. 113-122).
- Vihavainen J. 2014. VVER-440 Thermal Hydraulics as a Computer Code Validation Challenge. Doctoral Thesis. Acta Universitatis. Lappeenrantaensis 618. Lappeenranta University of Technology, LUT School of Technology, LUT Energy, Energy Technology. http://urn.fi/URN:ISBN:978-952-265-717-6 (pp.47-53)

# List of recommended further reading:


- OECD/NEA. 2004. Engineered Barrier Systems (EBS): Design Requirements and Constraints – Workshop Proceedings, Turku, Finland 26-29 August 2003. available on https://www.oecd-nea.org/rwm/public-documents/ (nea-4548)
- OECD/NEA: 2003. Engineered Barrier Systems (EBS) in the Context of the Entire Safety Case - Workshop Proceedings, Oxford, United Kingdom 25-27 September 2002. available on https://www.oecd-nea.org/rwm/public-documents/ (nea-4308)
- NASA. 1995. Systems Engineering Handbook. SP-610S. e.g. on http://web.stanford.edu/class/cee243/**NASAS**E.pdf or newer version (2007)


# Web resource:

Rational DOORS

https://www.youtube.com/watch?v=qYK7_g4Fy44 (12 min demo)

http://www-03.ibm.com/software/products/en/ratidoor

This presentation provides also a short introduction to the V-model.

# Conditions for use of this training material

The training materials for the DOPAS Training Workshop 2015 have been produced partly with the European Commission's financial support. The materials can be downloaded from the DOPAS WP7 webpage and used in general freely without a permission for non-commercial purposes providing the source of the material and Commission support is referred to.

The figures and pictures in each presentation originate from the organisation that has produced the specific training material unless mentioned otherwise.

Some photos and materials in the presentations present prior knowledge (background information) of the consortium partners. This information is marked with © and requires a permission for all uses from the copyright owner.

Non-commercial use means that if this training material is used e.g. in education, training, or consulting no fee may be collected from using this material.

For other uses, please contact the DOPAS project.

DOPAS

www.posiva.fi/en/dopas

The research leading to these results has received funding from the European Atomic Energy Community's (Euratom) Seventh Framework Programme FP7/2007-2013, under Grant Agreement No. 323273 for the DOPAS project.