Fontys UNIVERSITY OF APPLIED SCIENCES › FOR SOCIETY

BEYOND THE V-MODEL

INTEGRATING GOVERNANCE, DESIGN LOGIC, AND ITERATION

Erik Puik
Fontys University of Applied Sciences Eindhoven, NL
INCOSE IS, Yokohama, June 13th 2026

1

Agenda

- Reality Can Be Harsh and Troublesome
- Basic V-Model
- Layered V
- Axiomatic Design
- Iteration & Agile
- Practical Application
- Exercise: What to do in your next project
- Summary

Fontys UNIVERSITY OF APPLIED SCIENCES

2

2

1 REALITY CAN BE HARSH AND TROUBLESOME

3

1 Reality Can Be Harsh and Troublesome

Your requirements pass review. Teams build subsystems in parallel. At integration, one design choice affects three others. Nobody knew they were connected. By the time testing catches it, repairs cost ten times what prevention would have. Rework eats your budget, deadlines slip, teams stop trusting each other.

Beyond the V-Model gives you one method:

- connect what the system must do to how you build it to what you can measure and control
- Make those connections visible
- When tests fail, follow the trail back
- Changes stay small instead of spreading

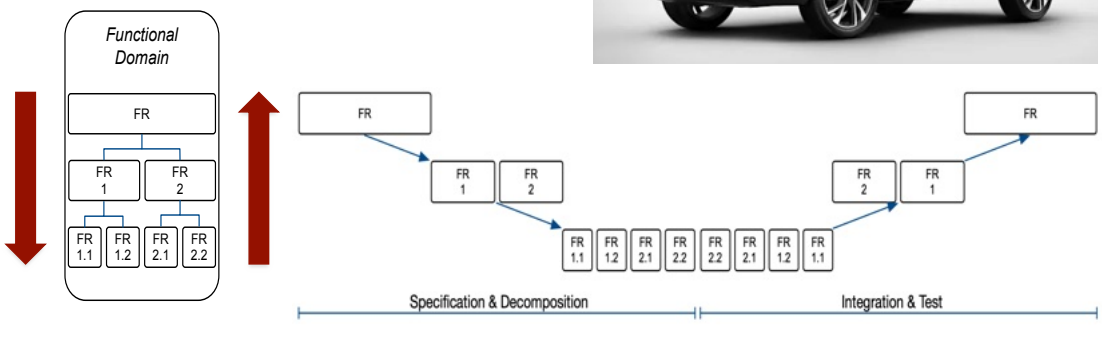
4

2 V-MODEL

5

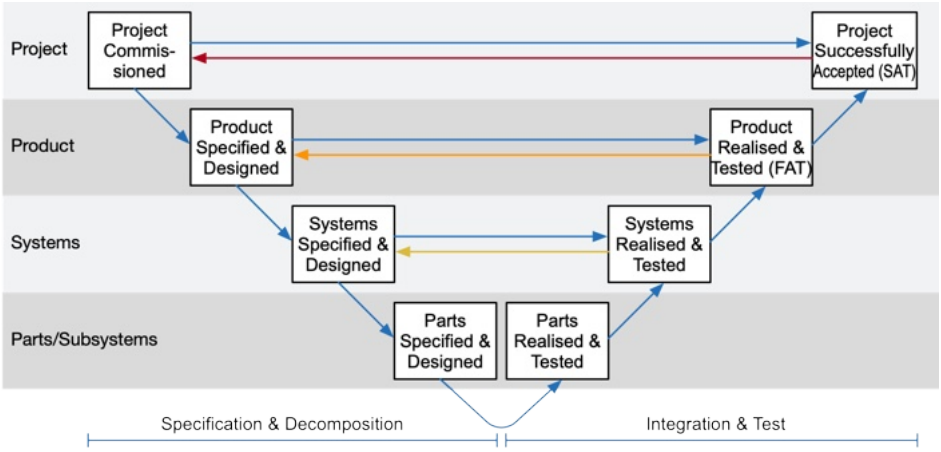
2 V-Model: Reinventing the V-Model

- Can you functionally decompose a car and then rebuild it
- Can you align your activities chronologically with respect to the hierarchy?



6

2 V-Model: A Basic Layout



Fontys UNIVERSITY OF APPLIED SCIENCES

7

7

2 V-Model: Exercise (10 min)

1. Try to specify the 'secret object' in the 3-5 most optimal requirements
   - do not mention the name of object
   - Do not use specific keywords

E.g., if the Object is 'a Train'
- *Transport many people at once*
- *Stop at designated places*
- *Operate environmentally friendly*
- *Support people during operation*

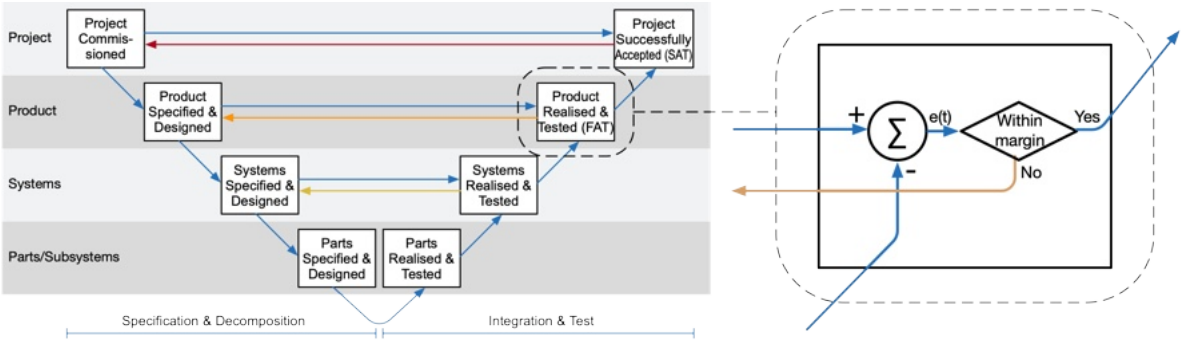
2. Hand your requirements to the next group (clockwise) and try to guess the object of the former group

Fontys UNIVERSITY OF APPLIED SCIENCES

8

8

V-Model: Verification/Validation

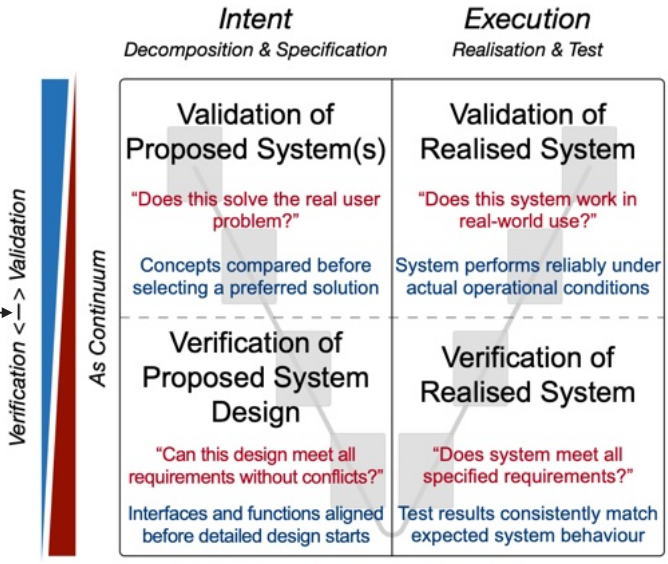


9

2 V-Model: Intent & Execution

Problems arise when one quadrant answers another's question

Leads to Tension



10

2 V-Model: Equivocality

- Equivocality refers to the presence of multiple plausible interpretations within a single statement. It is distinct from uncertainty: uncertainty stems from missing information, while equivocality persists even when more data is available. Resolving it requires interpretation, dialogue, and structured clarification.
- In the V-Model, equivocality often appears early, during requirements definition, and can persist unnoticed until integration. Because the model links artefacts, not meanings, it may pass formal reviews while concealing significant differences in interpretation between disciplines.

Example: A requirement reads, “The system must respond in real time.”

- A software engineer may interpret this as a response within 10 milliseconds
- A mechanical engineer may assume 1-2 seconds is acceptable
- A clinician may simply expect no perceptible delay

Fontys UNIVERSITY OF APPLIED SCIENCES

11

V-Model: Actors with Legitimate Logic

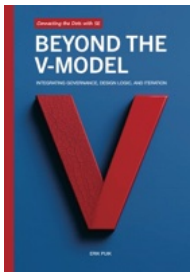
Dominant Group	Primary Logic	View of Uncertainty	Primary V-Model Concern
Managers, sales, product planners	Direction, value, commitment	Uncertainty is a business risk to be bounded	Governance
Exploratory Engineers	Learning, structuring, solution finding	Uncertainty is the normal medium of early development	Design Logic
Consolidation Engineers	Control, repeatability, execution	Unresolved uncertainty is a risk to be reduced before work continues	Agile / Realisation

Fontys UNIVERSITY OF APPLIED SCIENCES

12

12

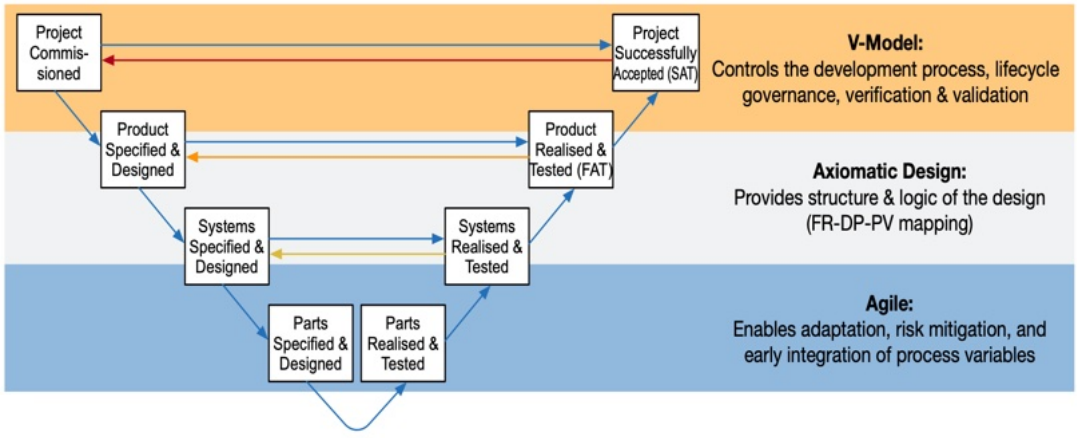
3 LAYERED V-MODEL



13

13

3 Layered V: Working Space in Layers



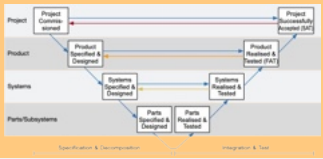
14

14

3 Layered V: Different Way of Looking

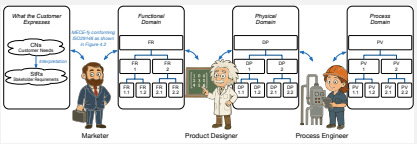
LAYER 1: GOVERNANCE: *When do we decide?*

- Phases and milestones (FAT, SAT)
- Stage-Gate decision points
- Risk management and project control



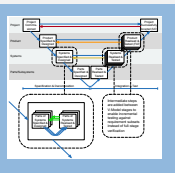
LAYER 2: DESIGN LOGIC: *How do we design?*

- Structured transformation
- Axiomatic Design principles
- Independence and traceability throughout



LAYER 3: ITERATION: *How do we learn?*

- Agile sprints embedded within V-phases
- Early feedback loops
- Continuous validation and improvement



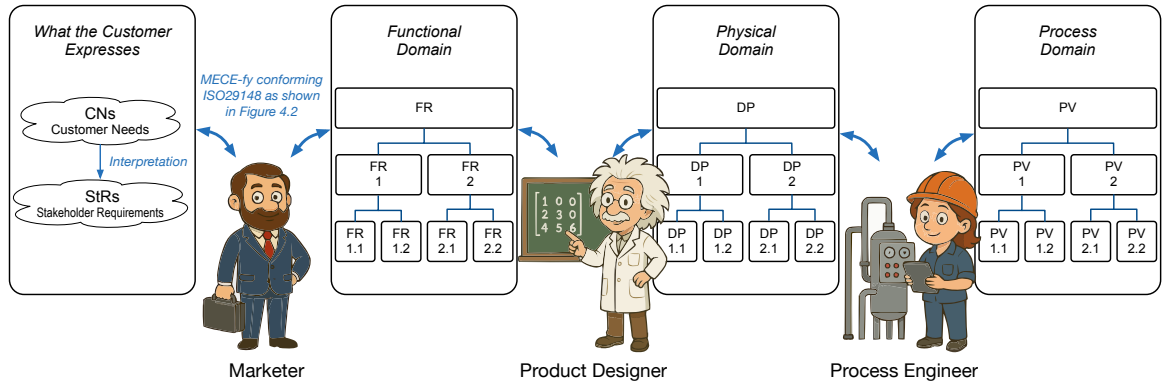
15

4 AXIOMATIC DESIGN

Fontys
UNIVERSITY OF APPLIED SCIENCES

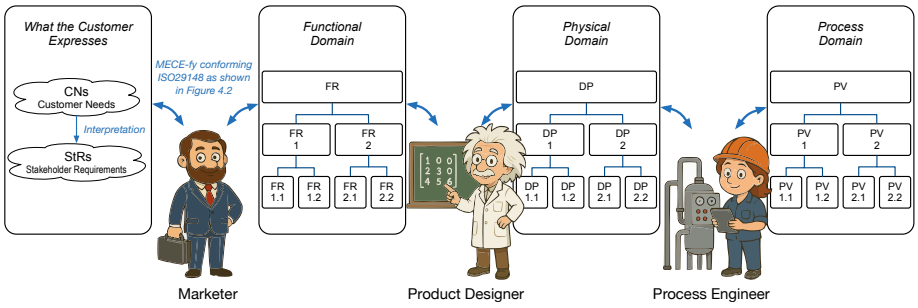
16

4 Axiomatic Design



17

4 Axiomatic Design: Logic in Design



- Customer Domain (CN) → What does the customer want? The original need or problem statement
- Functional Domain (FR/nFR) → What should the system do? Functional requirements (FR) define behaviour; non-functional requirements (nFR) define constraints and quality attributes.
- Physical Domain (DP) → How do we realise that? Design parameters that implement the functions.
- Process Domain (PV) → How do we make it? Process variables that control manufacturing and operation.

18

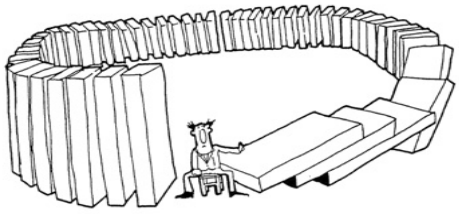
4 AD: Introducing the Matrix

Axiom 1: Doing the Right Things

Basic Idea
- Map Requirements to Solutions
- Make all relationships explicit
- Visual representation of logic

What Matrix Shows
- Which DP affects which FR
- Strength of relationships
- Pattern reveals quality

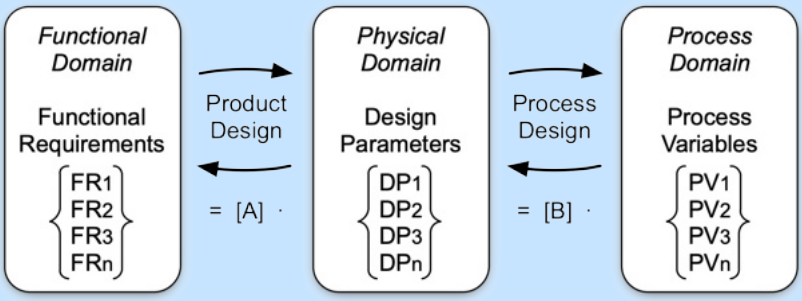
The Core Principle:
- Axiom 1: Maintain FR independence
- Each function adjustable separately
- Prevents cascading changes



19

4 AD: Design Matrix

Axiom 1: Doing the Right Things



- [A] = Design Matrix
- [B] = Process Matrix

20

20

4 AD Example: Faucet (1)

- Define matrix for adjustment of warm & cold water
- Is it coupled or decoupled?

Fontys UNIVERSITY OF APPLIED SCIENCES

21

4 AD Example: Faucet (2)

- Coupled design; temperature and flow cannot be adjusted independently

FRs

DPs

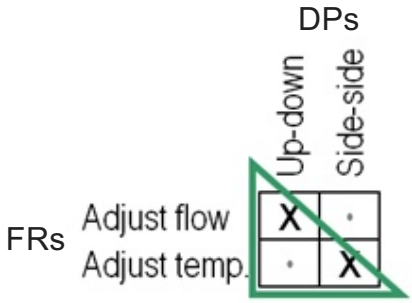
	Hot valve	Cold valve
Adjust flow	X	X
Adjust temp.	X	X

Fontys UNIVERSITY OF APPLIED SCIENCES

22

4 AD Example: Faucet (3)

- Improved design



23

23

4 Axiomatic Design: Design Matrix

	DP1	DP2	DP3
FR1	X	0	0
FR2	0	X	0
FR3	0	0	X

Uncoupled Design

	DP1	DP2	DP3
FR1	X	0	0
FR2	X	X	0
FR3	X	X	X

Decoupled Design

	DP1	DP2	DP3
FR1	X	X	X
FR2	X	X	X
FR3	X	X	X

Coupled Design

24

24

4 Axiomatic Design: Exercise (10 min)

Discuss in groups, 3-4 persons per group:

- Do you recognise coupled designs in your work
- Did you succeed in decoupling them
- Can you visualise in a simple matrix (e.g. 3x3)
- Put your results on paper sheet

Fontys UNIVERSITY OF APPLIED SCIENCES

25

25

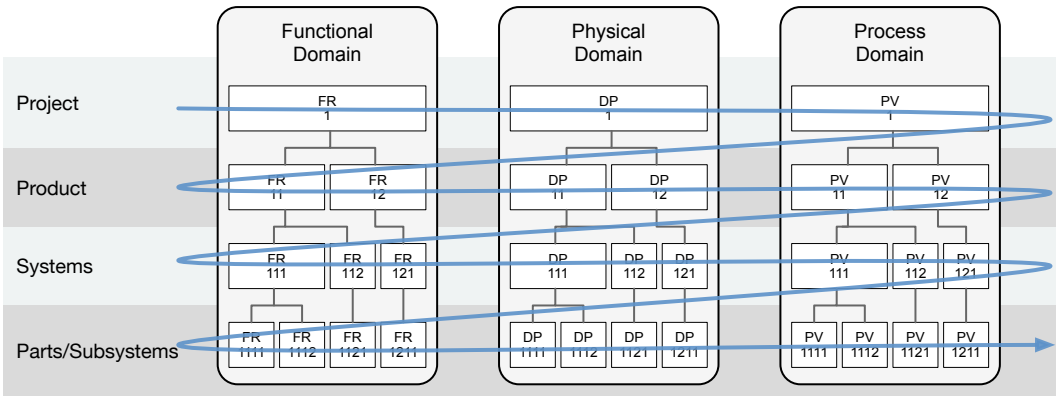
Short break (15 min)

Fontys UNIVERSITY OF APPLIED SCIENCES

26

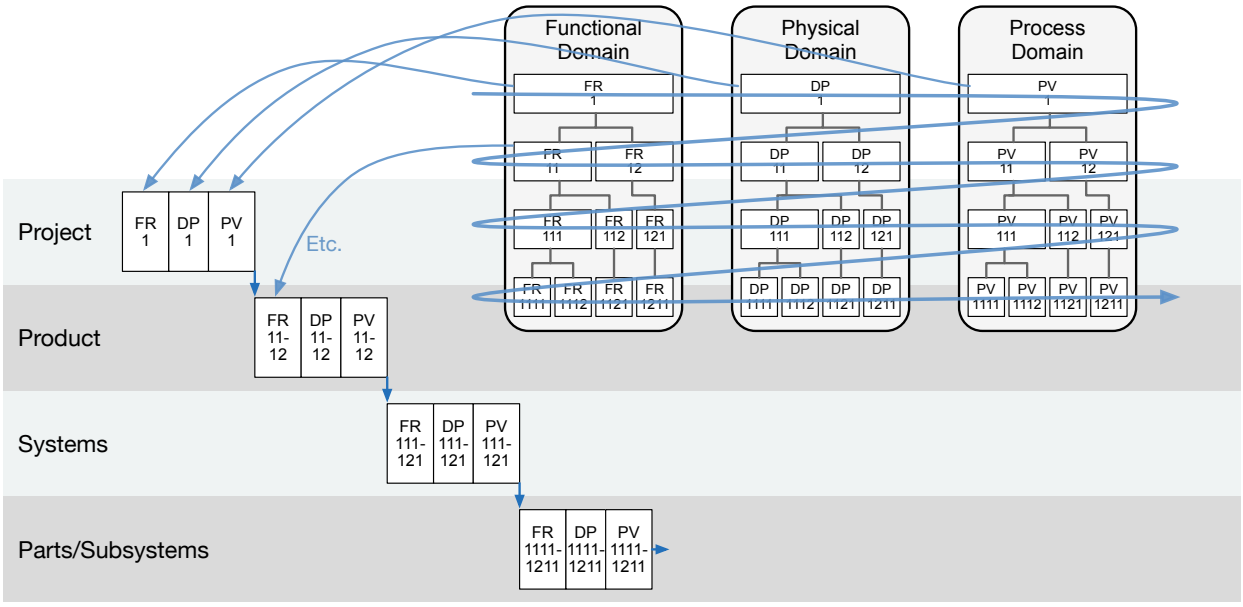
26

Zigzagging



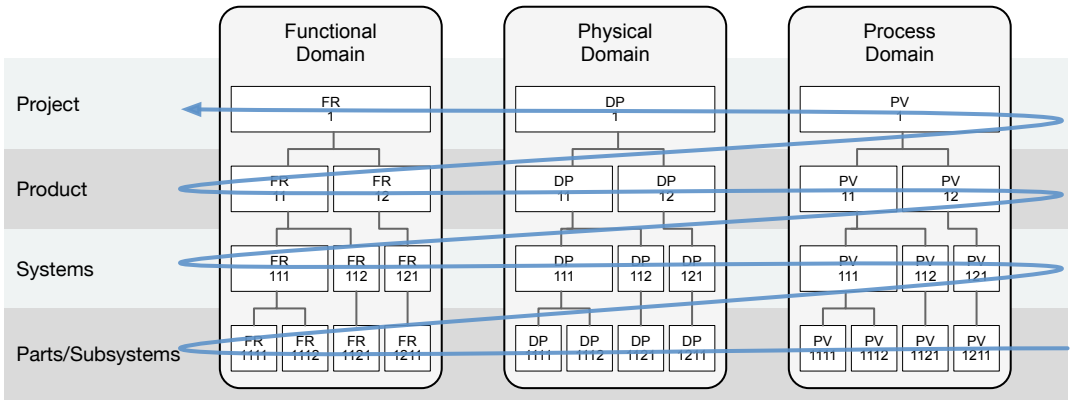
27

Zigzagging and the V-Model



28

Reverse Zigzagging

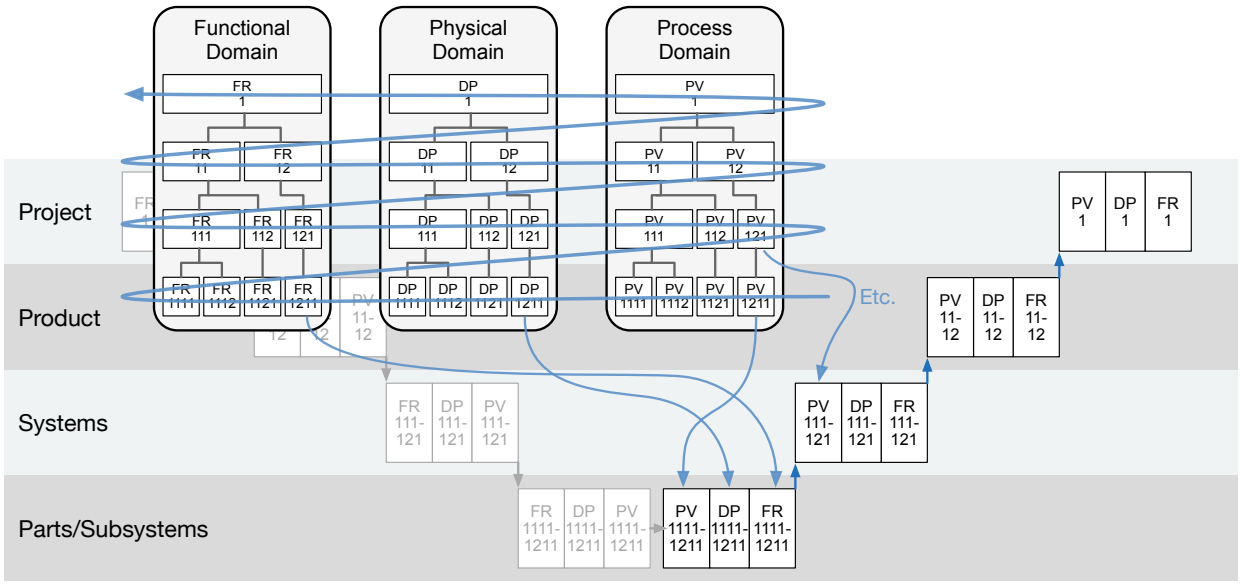


Fontys UNIVERSITY OF APPLIED SCIENCES

29

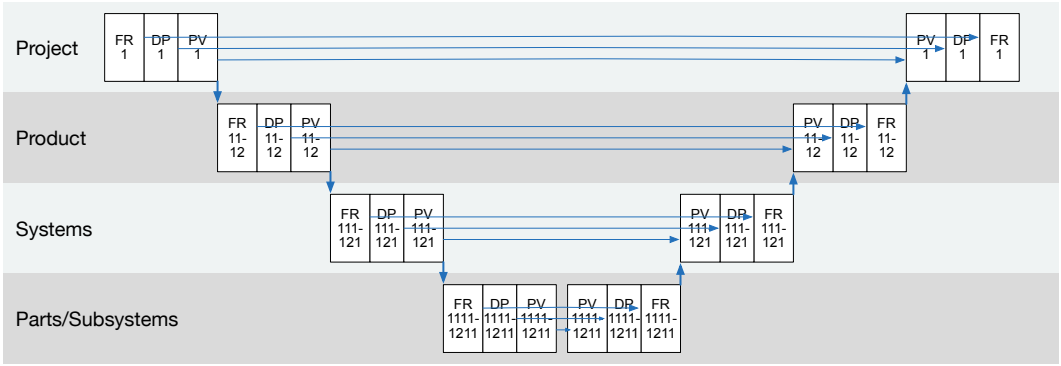
29

Reverse Zigzagging and the V-Model



30

Flow through the Domains

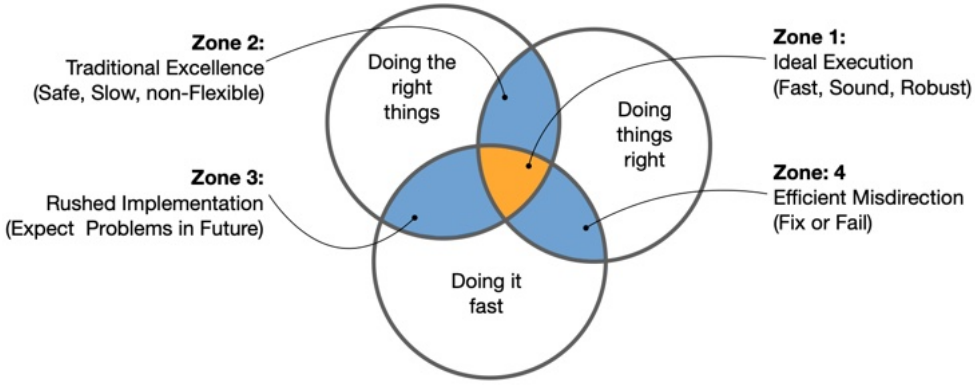


31

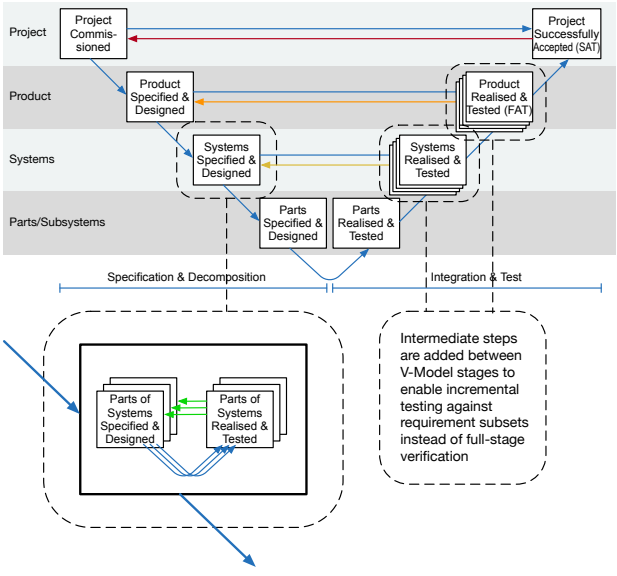
5 ITERATION & AGILE

32

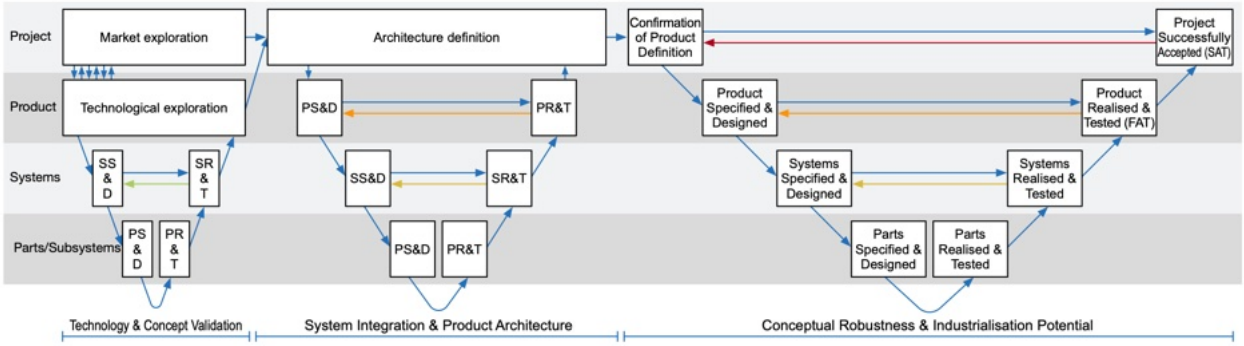
5 Iteration & Agile



5 Iteration & Agile in the V-Model



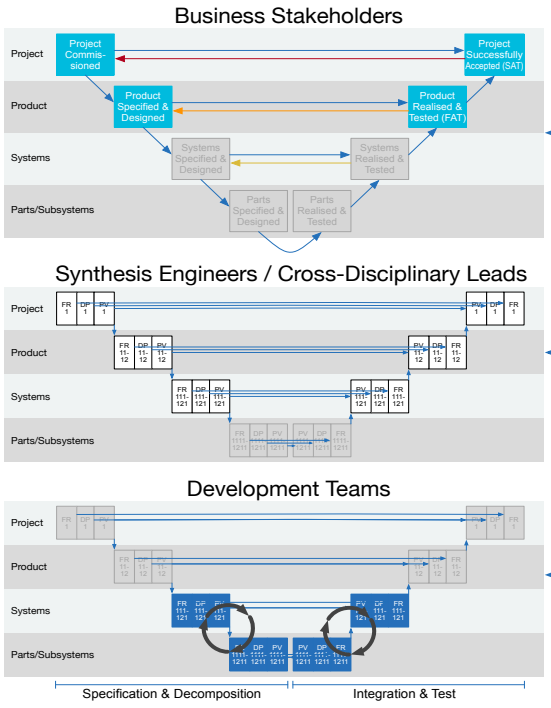
5 Iteration & Agile: Nested V-Model



6 Application in Practice

Three steps to integration:

- Step 1: Establish Governance; plan your phases and decision points
- Step 2: Apply Design Logic; structure your design process using Axiomatic Design
- Step 3: Build In Iteration, embed learning loops throughout



Define business goals, market needs, and product-level requirements. Provide strategic direction, boundary conditions, and initial constraints that shape the scope and high-level functional intent of the system

Integrate functional, design, and process domains into coherent system structures. Ensure FR–DP–PV alignment across levels. Drive design synthesis through abstraction, decomposition, and cross-disciplinary reasoning to maintain system consistency

Design and realise subsystems or components. Select methods based on task nature: iterative for exploration, Axiomatic Design for risk-critical elements needing functional clarity, early validation, and controlled complexity

7 Exercise: What to Do For Your Next Project

Now it's your turn:

- Take the Quick Guide, Chapter 7 is your worksheet
- You'll work on your own current project for 30 minutes
- Goal: go home with one concrete intervention for next Monday
- Work individually or pair up with a colleague from your company

37

7 Your Project, Three Steps (30 min)

1 Sketch your project as a V
   - Where are you right now: requirements, design, realisation, verification?

2 Find your weakest layer (checklist, QG §7)
   - Governance; unclear milestones, scope creep, unmanaged risks?
   - Design Logic; coupled functions, unclear requirements-to-design link?
   - Iteration; problems discovered late, rigid process?

3. Define ONE intervention for next Monday
   - Small, concrete, within your own control.
   - Name the artefact: which matrix, which milestone, which test cycle?

4. Pitch your intervention to the group

38

8 Summary

The V-Model is not a linear process but a layered framework; master the layers and you will master your project:

- Governance provides structure and control
- Design Logic ensures robust, traceable solutions
- Iteration enables learning and adaptation

Together, these three layers transform the V-Model from a simple process diagram into a complete systems engineering framework.

39

Thank you for your attention

› FOR SOCIETY

Erik Puik, erik.puik@fontys.nl

40