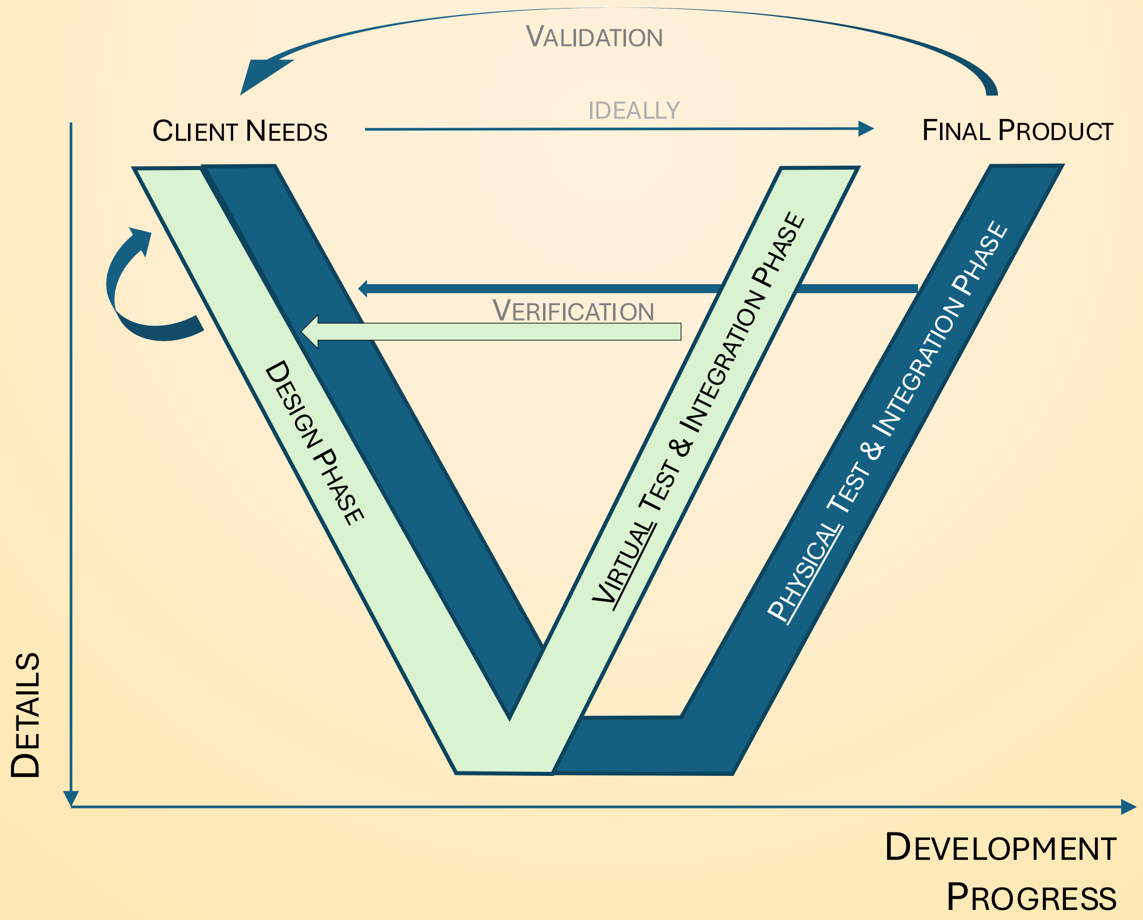
How is the System Engineering V-model evolving?

See here the V-model explained



Dr. Clément Coïc

How is the System Engineering V-model evolving?

First, let’s clarify one thing…

The concepts that the V establishes remain unchanged.
These are almost “universal truth”:

- Collect client needs,
- capture them into system requirements,
- decompose your system into subsystems,
- decompose the subsystems into components,
- Design and size the components,
- Test the components (and verify),
- Integrate and test the subsystems (and verify),
- Integrate and test the system (and verify).
- Validate that the needs are fulfilled

Before getting the critics: the “V” does not say how you should do it. If you want to be agile in the process, do it. It does not matter.

What matters is that you fulfill a need.

And complex systems can’t hardly be developed without decomposing them. Software are also developed this way.

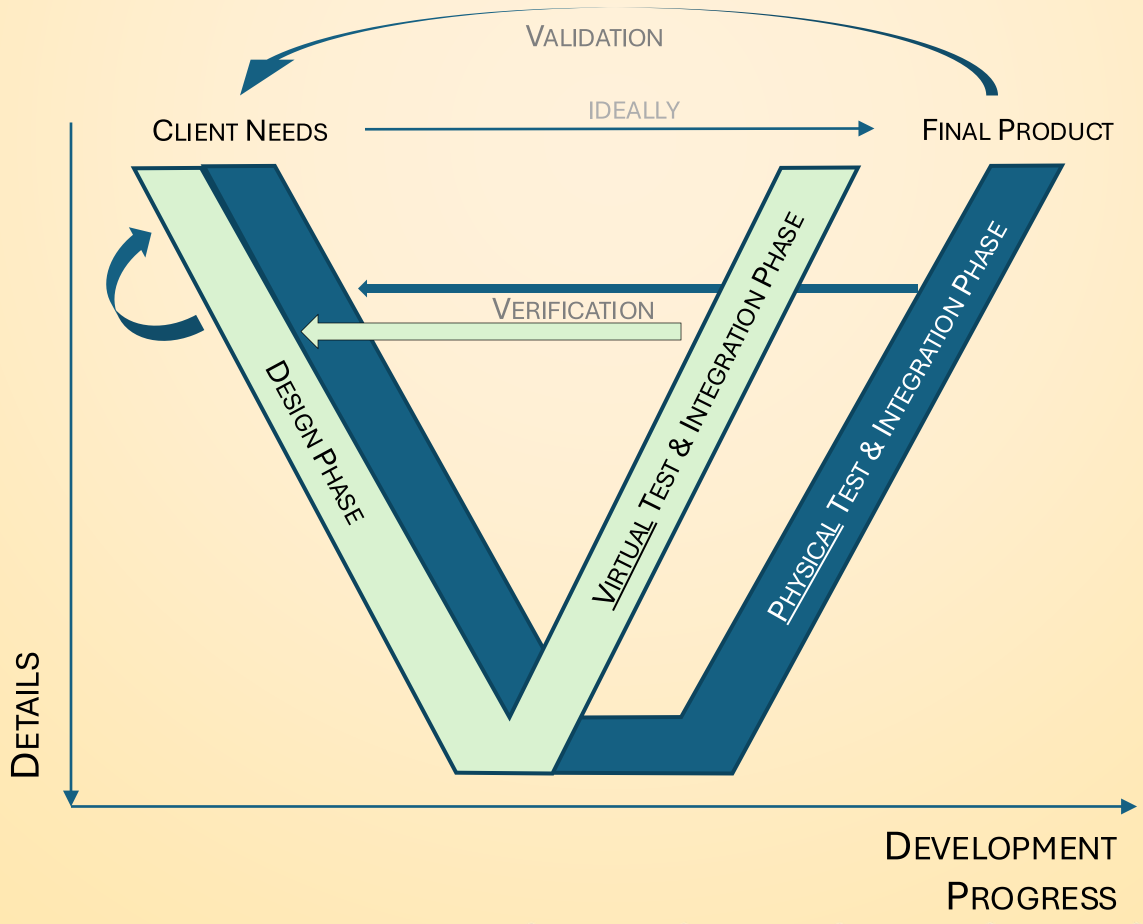
How is the System Engineering V-model evolving?

So, what is new?

Tool capabilities have evolved, and we can now develop a full virtual model of the system.

In practice, this means that we can follow a double-V model, where a first virtual V precedes the physical V.

Physical implementation only starts when the virtual V is completed.



Dr. Clément Coïc

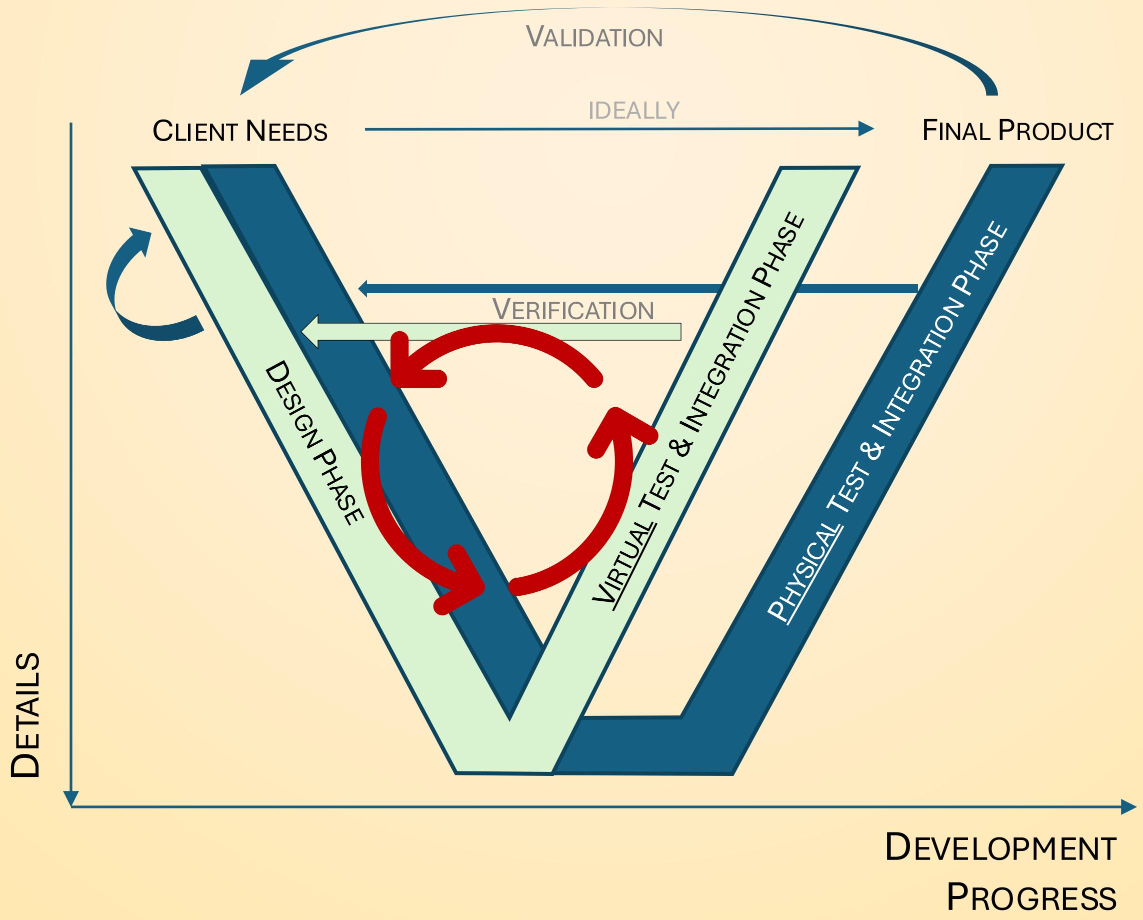
How is the System Engineering V-model evolving?

Design iterations happen virtually, which is faster, less expensive and can be more thorough than physical testing.

This can be done at all level: component, subsystem and system level.

Virtual Verification is conducted – and can be automated. The full system performance is thus verified before implementing the physical prototype

The goal is to be "right the first time" with the physical tests.



Dr. Clément Coïc

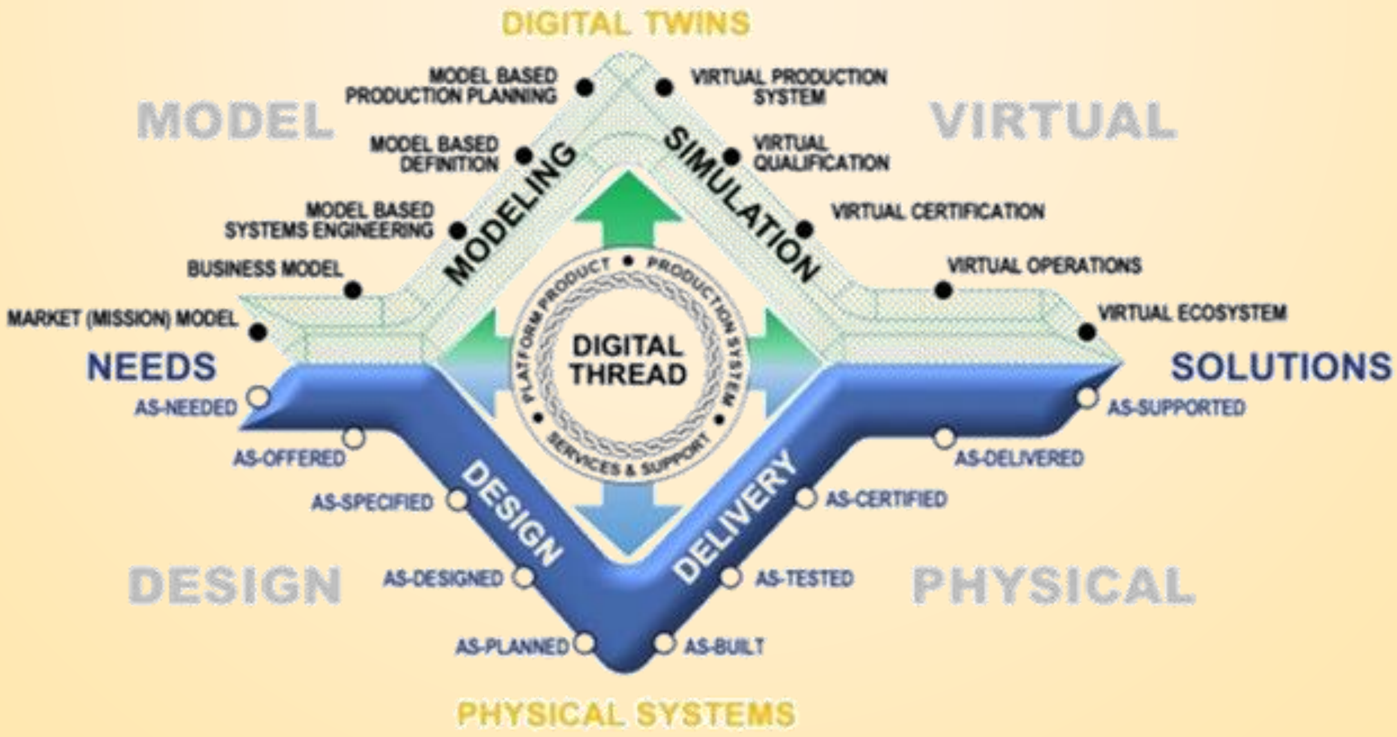
How is the System Engineering V-model evolving?

Are there more representations?

The MBE Diamond, proposed by Boeing, illustrated below, shows a similar view where the virtual V is a symmetry of the physical one.

While it adds to the double-V some key insights, it can also introduce some confusions if wrongly read. The stages of physical V and virtual V are not synchronized, even if aligned.

The virtual V shall happen first to bring value.



What are the key blockers to a Virtual V?

1. Often models are not trusted compared to physical tests. This is almost ironic when one considers all the potential sources of errors in a physical test and the measurement of the results. Both physical and virtual prototypes and test benches should be thoroughly verified and used to their full potential.
2. Digital continuity – the continuity of data – throughout the virtual V is today a strong limitation to virtual development. Many models are only snapshots of a given design, and virtual iterations require redoing the work.
3. Design continuity is also not happening due to different roles (persons / departments) working with different tools and potentially lacking the skills for virtual development.
4. Test-Driven Development for virtual models of physical systems is still complex to put in place and this prevent a direct feedback from failing designs.

…

Also, we are running a poll to understand which development cycle you are mainly using.
Please vote to let us know!

Comment if you need any further clarifications or insights.

Dr. Clément Coïc