Mastering Challenges in Systems Engineering

Introducing practice stages to path the way

Understanding the structure and behavior of complex systems – Transforming your Development towards the Digital Twin – Harvesting benefits from early validation

As systems are becoming ever more complex, CTOs as well as System Engineers are confronted with manifold challenges:

- determining the strategy to stay ahead of the market;
- bringing products and innovation faster to market;
- managing increasing product complexity;
- ensuring that the systems meet all functional and nonfunctional requirements;
- coping with questions about product line / reuse-oriented development;
- selecting the right tools and processes given the rapidly changing technologies;
- designing systems as a whole and integrating electrical, mechanical and software components as well as human aspects;
- dealing with the continuously growing challenges of managing projects and handling collaboration and communication issues;
- allocating dwindling resources (human, financial and time) – and many more.

This complex starting point makes it all the more important not to lose sight of the big picture in order to be successful in the long-term.

Models not only contribute to a better understanding of structure and behavior, but also help understand the system more clearly earlier on in the development phase.

Modeling – a fundamental part of engineering

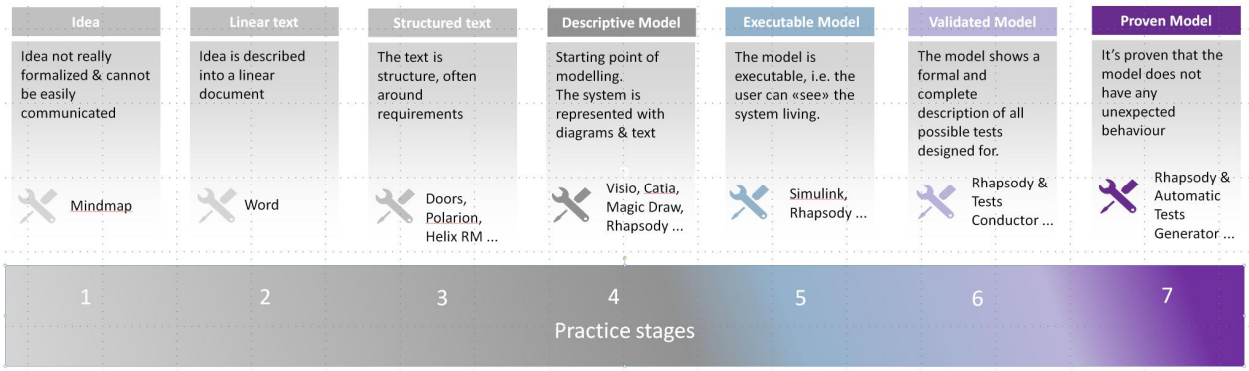
Models have become essential for understanding the structure and behavior of complex systems. Modeling helps in simulation and early validation and is a fundamental part of engineering. It is an effective way to manage the complexity of system and software development, enabling communication, design and assessment of requirements, architectures, systems and software. That's why the industry is moving towards Model Based Systems Engineering (MBSE). However, there is more to MBSE: executable models leading to Model Driven Systems Engineering (MDSE).

MDSE leverages graphical models and pre-built application components so that users not only can visually construct but also virtually interact with the system and test their complex systems through executable graphical models. As such, the models not only contribute to a better understanding of structure and behavior, but also help understand the system more clearly earlier on in the

development phase. As a result, models increase the ability to improve and validate ideas and concepts as early as possible in the process when it costs less to adapt.

An effective approach to drive the challenging tasks of improving Systems Engineering in a structured way is the Systems Development Taxonomy (refer to Fig. 1). This classification has been established based on many different domains, projects and customers. It determines seven practice stages and also helps in laying out a path for the incremental adoption of MDSE within an organization.

Systems Development Taxonomy: Practice Stages



The journey starts at Stage 1 with the simple idea and may finally lead to Stage 7 on the other end of the spectrum with the tested and proven model. The importance and value of each of these consecutive 7 stages is given in order to continuously improve the Systems Engineering capabilities.

An effective approach to drive the challenging tasks of improving Systems Engineering in a structured way is the Systems Development Taxonomy.

The System Development Taxonomy is field agnostic: It can be applied on each system development phase and is applicable to every domain – being it mechanical, software, analogue and digital electronic, optics, etc. Let us briefly go through the stages:

Stage 1 focuses on the idea, stages 2 and 3 on text. These stages are without modeling and are characterized by no or just reduced traceability.

- At Stage 1 there are loosely connected ideas which can be expressed by undemanding tools such as MindMap. Only the idea owner is able to deal with it, which hampers communication.
- Stage 2 is the result of writing such ideas in the form of linear text with text processing tools such as Word, making the idea more explicit, readable and more understandable by other people than their author. Also, it is now possible to manage document versions.

- Stage 3 is characterized by a structured set of text pieces. These pieces can be named requirements. Traceability is now possible among these pieces. Tools to improve traceability and management are e.g. Doors, Doors Next, Polarion, Helix RM etc.

Companies taking their present stage into account and taking coordinated steps to reach the next stage have shown improvement by factors.

Stages 4 to 7 are with the support of a model. It enhances the capabilities of the engineering process through the whole development workflow.

- The Descriptive Model at stage 4 is the entry point for modelling. In addition to textual, it allows a graphical representation of the system also from a behavioral and architectural point of view. It highlights the interactions between the different elements composing the system. Language specific tools are required such as Visio, MagicDraw, Enterprise Architect, Catia, ProEngineer or IBM Rhapsody support this approach. Some of these tools support you in going all the way towards Stage 7, some are limited to Stage 4, and some go some way with add-on tools.
- The Executable Model at Stage 5 allows the complete and executable description of the system in scope. The model of the system has to be executable, i.e. runnable and allowing to perform a system simulation. It enables the user to «see» the system living and can interact with Human Machine Interfaces, virtual mockups, etc.
- The Validated Model at Stage 6 includes a formal and complete description of the tests and allows to verify that the (model of the) system does everything what it is supposed to do.
- The Proven Model at Stage 7 allows to verify that the (model of the) system does not do more than it should, i.e. doing the right things right.

The potential, however, lies in Stage 5, meaning putting the executable models in the center of your activity.

An incremental evolution

It depends much on the given context which stage your development is and should aspire to be. Important is the movement from left to right, i.e. from Stage 1 towards Stage 7. However, it is imperative that this evolution needs to be taken incrementally and in small steps with a sound vision and management support.

Many companies try to skip certain stages or want to make too large steps and therefore do not reach their goals. Companies taking their present stage into account and taking coordinated steps to reach the next stage have shown improvement by factors, creating as little waste as possible while leveraging value as much as possible.

Stage 4 is the tipping point and opens the strategic opportunity to harvest from MDSE. For many companies modeling is focused on descriptive tasks, i.e. supporting better documentation. The higher potential, however, lies in Stage 5, meaning putting simulation and the executable models in the center of your activity and opening the path to early validation (Stage 6). Validating the model is by far less

expensive than using prototypes. Also, model validation allows you to validate very complex scenarios before realizing prototypes. This become s exponentially more valuable as the complexity of your system increases. This Stage 6 is required to reach the Digital Twin and thus allows you to experiment, simulate, test and develop more virtually. Moving to Stage 4 requires a change of the mindset and thus a change in the development paradigm, but it is also the first vital step towards the Digital Twin. From our experience, going from Stage 4 to Stage 5 is also a tough decision for an organization. Both steps however help substantially to reduce risks, resources and time to market.

Serenity – a cutting edge systems engineering approach

Serenity contributes to cope with the constant changes within the own ecosystem as well as in the business environment.

At EVOCEAN, we have established a model driven process framework called *Serenity*, focusing on improving systems development. *Serenity* includes a common language to communicate more effectively and coherently to speed up product and systems development and innovation enabling prosperous organizations.

Serenity is not just a technical approach to improve the working processes in Systems Engineering. It is a proven and innovative method which positively affects the entire organization and paths the way to benefit from digital twins. By enhancing multi-disciplinary interactions and collaboration, it contributes to cope with the constant changes within the own ecosystem as well as in the business environment and supports the transformation towards a lean and more collaborative organization.