

MERGING PROCESS STRUCTURE AND DESIGN FREEDOM

Systems Engineering (SE) is a methodical approach to the development of new (high-tech) products. For a long time, SE has been considered too complex and unnecessary for application in small & medium-sized enterprises (SMEs). That is now rapidly changing. In 2012 a group of Dutch universities of applied sciences and companies started a so-called RAAK-MKB project to make the tools of SE more accessible for SMEs. At the same time these tools can be used for multidisciplinary student projects, to teach students how to approach the development of a new product.

RINI ZWIKKER AND JOS GUNSING



Introduction

Developing a new product is a beautiful but at the same time almost intangible cooperative process, in which customer, technology and business meet. Various fields of expertise speaking different languages must be merged. In this process there is a continuous alternation between creativity and analysis. A discussion or creative idea can lead to an entirely new solution. But a thinking error or

simple miscalculation can make a nice idea fall flat. Cooperation is essential, but also a source of misunderstandings.

There is a common interest: realising a new product which successfully fulfils a customer's needs, and at the same time takes into account a number of constraints. But there will also be contradictions between the project contributors. One member wants to control processes and avoid risks, both financial and technical. Another member would like to keep on changing and improving, preferably up to the end of the design.

Systems Engineering (SE) can contribute to this process. It aims to provide sufficient structure, without limiting the design freedom unnecessarily. It can assure that all those involved, from manager to designer to marketer, start speaking the same language in order to unite the different interests. And it can assist in defining the functions, requirements and peripheral conditions in such a way that the new product really fulfils the user's needs.

This article gives an overview of the fundamentals of SE and its benefits, using a RAAK-MKB project as an illustration (see the box). Special attention is paid to the difference between the structure of the process, which is more or less unidirectional, and the actual way of working of the participants, which is more iterative.

AUTHORS' NOTE

Rini Zwikker is a professor of Mechatronics at Saxion University of Applied Sciences (UAS) in Enschede, the Netherlands, and a mechatronic systems engineer at Demcon. Jos Gunsing is a professor of Robotics and Mechatronics at Avans UAS in Breda, the Netherlands, and a technology innovator at MaromeTech.

They both acknowledge the input and critical review by Henk Kiela, professor of Mechatronics and Robotics at Fontys UAS in Eindhoven, Erik Puik, professor of Microsystems Technology at HU UAS in Utrecht, and Joris van den Aker, Programme Manager at TNO-ESI.

j.m.zwikker@saxion.nl
jtg.gunsing@avans.nl

Furthermore, this article is a call to teachers involved in SE at universities of applied sciences to participate in the topic group that is now being formed, and to SMEs to come up with interesting cases for projects to be executed together with teachers and students, at the same time introducing the methods and tools in the organisation. Because only with the correct application of SE structure and innovation can combined – and serious – mistakes be prevented (see the box on page 7).

Definition

The INCOSE handbook 2004 [2] defines SE as: “An interdisciplinary approach and means to enable the realisation of successful systems.” With ‘systems’ we mean complex machines, with ‘engineering’ the complete process of specification, design, manufacturing and test. In order to achieve that success the approach must fulfil a number of conditions.

The INCOSE handbook continues with: “It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements and then proceeding with design synthesis and system

validation while considering the complete problem: operations, costs and schedule, performance, training and support, test, manufacturing and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.” The ‘customers’ include all stakeholders having an interest in the system, not only the users.

Where does SE fit?

A project consists of several sequential and parallel processes with various interactions in budgets, people, facilities, planning, risk mitigation and market information (Figure 1). Moreover, there are two different but highly complementary viewpoints on these same processes: project management (PM) and systems engineering (SE). PM primarily aims at controlling the process, while SE aims at steering the content of (design) choices.

The primary overlap between PM and SE is the planning. The project manager controls the planning and the budgets. The systems engineer gives input on the costs, the quality (requirements and verification of requirements) and the translation to the planning. The common interest of both

The RAAK-MKB project

RAAK-MKB is a Dutch government funding programme for universities of applied sciences to carry out applied research projects with – preferably – SME companies, teachers and students [1]. The acronym RAAK stands for Regional Attention and Action for Knowledge circulation, and MKB is Dutch for SME.

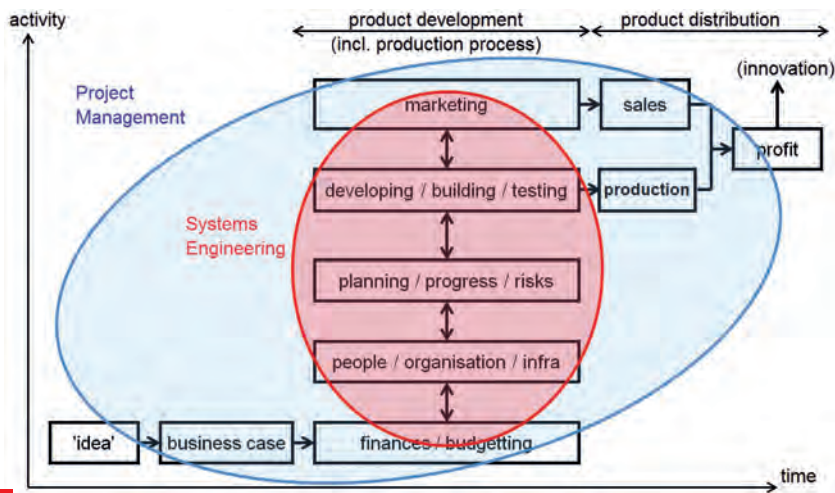
The RAAK-MKB project Systems Engineering for SMEs aims to contribute to the development of a toolkit with practical tools for the product development process, strongly connected to the (im)possibilities within SMEs. A second goal is to raise the knowledge level of teachers in the universities of applied sciences and as a spin-off the development of educational materials. The so-called V-model will be used as a basis, including the iteration loops.

Initiating partners of this project were the Research Centre for Robotics & Mechatronics (part of the Expertise Centre for Sustainable Innovation) of Avans University of Applied Sciences and Embedded Systems Innovation by TNO (TNO-ESI). Apart from Avans, leading the project, also active are the Fontys, HU and Saxion Universities of Applied Sciences and

the companies CE Masters, Ceratec, CSi Industries, Demcon Advanced Mechatronics, ESWE Technics, Fontijne Grotnes, Focal Meditech, Holmatro, Hotraco, IBS Precision Engineering, Irmato Jentjens, LAN Handling Systems, MA3 Solutions, Mecal, MTA, Nieaf-Smitt (Mors Smitt), Philips Innovation Services, Q-Sys, Tegema, Vanderlande and Wijdeven plus the industry organisations FEDA and Brainport Industries. Contacts are meanwhile also being made with the universities of applied sciences of Arnhem and Nijmegen, The Hague and Windesheim plus several newly interested companies.

The project started in September 2012 and will end in September 2015. Introduction of SE in companies had a much bigger impact on the organisation than originally expected. The board, directly involved employees and other departments must be included in the process. Within universities of applied sciences, convincing teachers and boards, and setting up educational material in combination with suitable projects, is also taking more time than anticipated, but progress is there.

WWW.AVANS.NL WWW.SAXION.NL WWW.FONTYS.NL
WWW.HU.NL WWW.ESI.NL



1

1 Systems Engineering and Project Management as complementary processes.

the project manager and the systems engineer in this process is to minimise project risks at the earliest possible stage.

PM includes more aspects than SE: business case, support staff, (after) sales, advertising and production. Setting up the production process is again a complementary interest of PM and SE, because of choices on production facilities, manufacturability and production costs. PM is not (or should not) be leading over SE or vice versa. Both are equally important to achieve a good result, commercially and technically.

Benefits and limitations

SE can be a powerful tool that creates clarity and understanding for the different phases of the product development process. It can increase its efficiency and effectiveness. It can support the development of a helicopter view for important decisions.

Examples of recent projects where serious and expensive mistakes were made

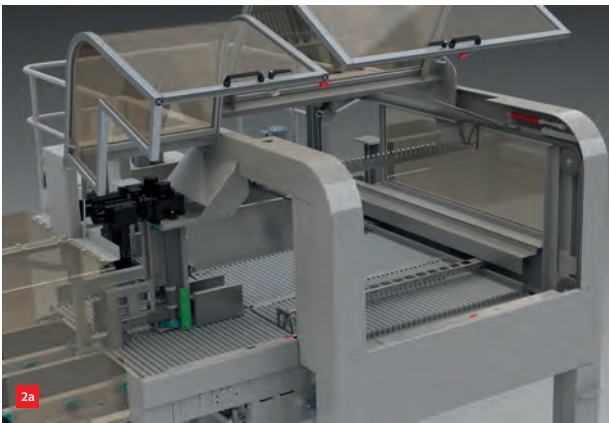
- The French aircraft carrier which was too short.
- The Fyra train (photo) which could not cope with snow.
- The French regional trains which were too wide for the platforms.
- The small Mercedes A which failed the moose evasion test.
- The Ariane 5 (photo) that crashed due to software copied from Ariane 4 without adaptation.
- The O-ring of a Space Shuttle booster fuel tank which could not cope with low temperatures.
- The iPhone antenna shielded by user's hand.
- Numerous car recall actions to correct errors with airbags, brakes or electronics.



The Fyra train. (Photo: Maurits90/Wikipedia)



Ariane 5 launch. (Photo: ESA/CNES/Arianespace)



2 Examples of complex products that required multidisciplinary engineering from RAAK-MKB participants.
 (a) Palletizer (CSi Industries).
 (b) Robot food packaging handler (LAN Handling Systems).
 (c) Bead optimisation system (Fontijne Grotnes); the bead is the edge of a tire that sits on the wheel.

development process. All these trends require a common information structure with associated data flows.

These issues were more than enough reason to step into the RAAK-MKB project for companies like CSi Industries, LAN Handling Systems and Fontijne Grotnes. Some of them have already been working with the SE concept for some time. One of the main conclusions in the early stage of the project was that organisational changes were necessary as well, such as setting the projects for module development (for CTO) more apart from the customer-specific system integration projects.

But SE is no substitute for solid technical knowledge, experience, understanding and creativity. It cannot match a close-knit team of professionals fully overseeing their assignment without documentation. It does not transform an engineer into an architect. It is no guarantee for a good cooperation. It is no cure for a lack of time, money, people and means. It does not automatically lead to an innovative and successful result.

Why SE?

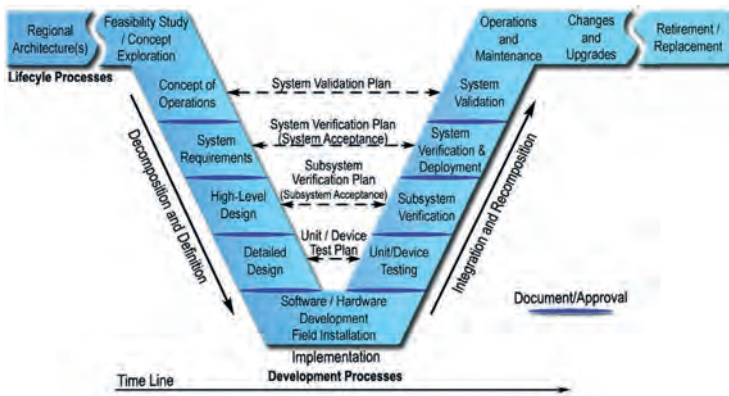
Related to product development, there are a number of well-known trends that increase the need to explicitly use methods from SE, also for SMEs. Products have become more complex and multidisciplinary over the last decades (Figure 2). Requirements on functionality, safety, environment and energy increase. In contradiction, the time and costs of development must be reduced. Series become smaller, with product variants created by reconfiguration: from build-to-order (BTO) via engineer-to-order (ETO) towards configure-to-order (CTO). This fits well in Industry 4.0 / Smart Industry trends. Products are being developed by various cooperating teams, in parallel or in series. A chain of companies is often involved. Direct integration of standard modules must be possible. The legislator demands traceability of the

Even with a strong belief in SE, which most of the product engineers /developers involved have, it takes quite some time to set up SE-based ways of working. It has the character of a culture change. All the companies involved have a strong belief that adopting SE will give them a stronger position in the market, more employee work satisfaction (less reinventing the wheel, more time for real innovation) and as such will become more future-proof.

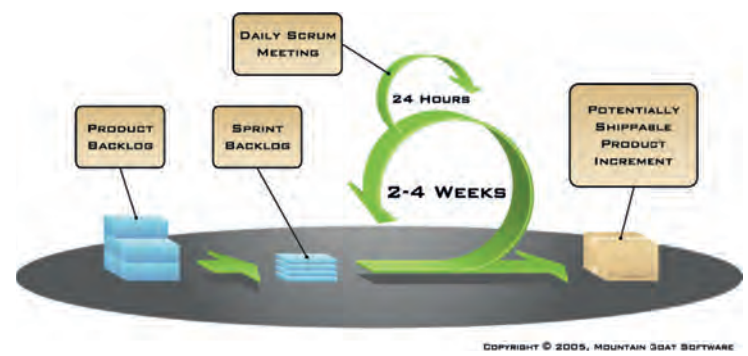
Different methods – one umbrella

Systems Engineering includes many different methods and tools to support the development process. The most well-known are Plan-Do-Check-Act (PDCA), the V-model, Agile, Spiral Design, Unified Modelling Language (UML), Methodical Design, ISO 2221, CAFCR, and the 3-Cycles Model; see Figure 3.

The methods available all have their own advantages and disadvantages, and are either more or less suitable for specific application areas, because all development projects are different: from simple to complex, from evolutionary to disruptive, from hardware-driven to software-driven. For software-driven projects the structure is often more cyclical, for hardware-driven projects more sequential: in software it is easier (and better for managing complexity) to add

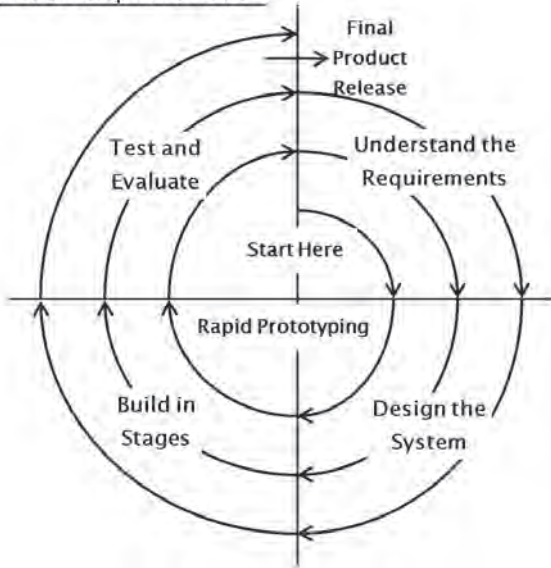


3a

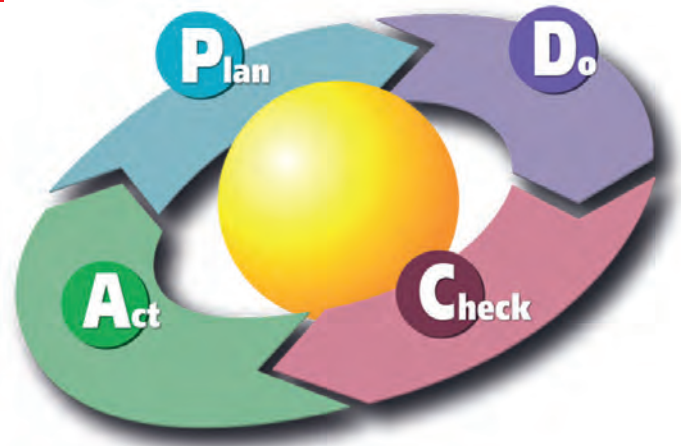


3b

Spiral Development Model



3c



3d

- 3 Various methods for Systems Engineering.
 (a) V-model [3].
 (b) Agile [4].
 (c) Spiral Design [5].
 (d) PDCA [6].

interviews, simulations and look-and-feel models to arrive at a consistent set of user requirements that is accepted by the customer and technology and business stakeholders. It is important to uncover the real user requirements together with the customer, and to formulate them at the right level of abstraction.

modules or functions during development than in hardware.

This stepwise addition of functions is the basis of the Agile method. But all SE methods must allow a combination of sequential and cyclical processes, as will be explained below. Especially for multidisciplinary (mechatronic!) systems the challenge is to merge and align the different approaches.

Common phases

The goal of all methods is a more structured course of the development process, by dividing it into a number of phases with well-defined interactions. Four phases can be found in almost all methods.

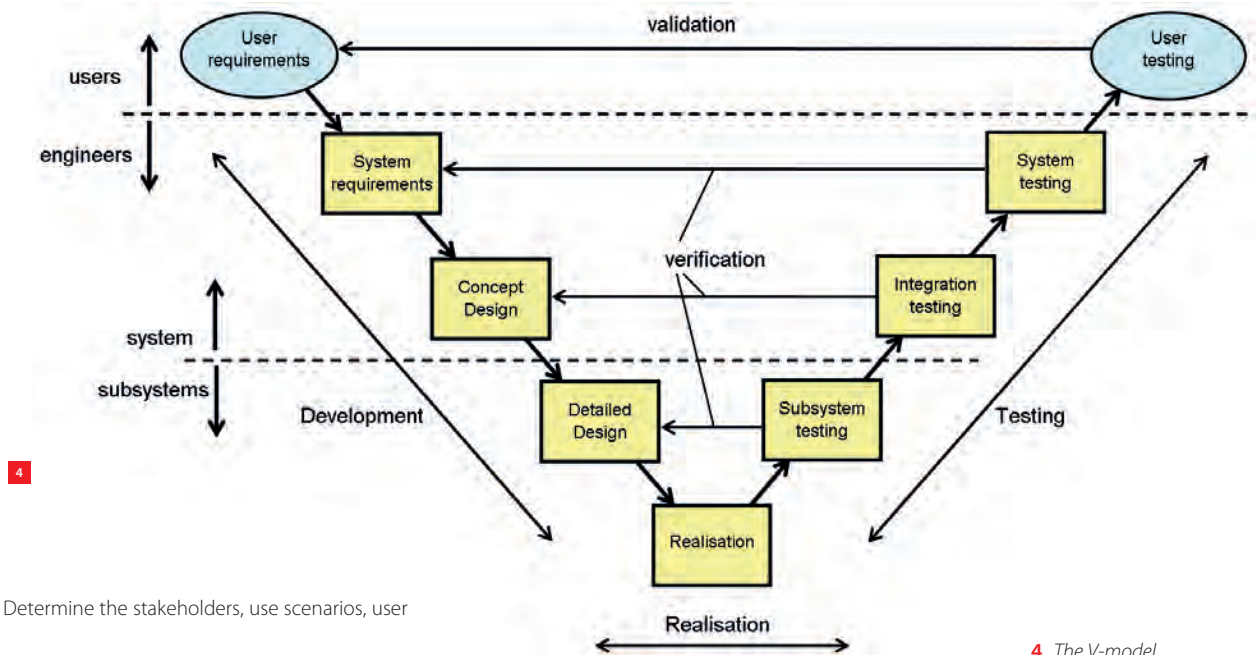
Phase 1 – Compilation of a consistent set of requirements

This starts with the user requirements: “What does the customer need?” These are the functions for the user process. Customers can often not yet formulate this precisely at the start of a new development. It is a time-consuming and iterative process with experiments,

Partly in parallel with the formulation of the user requirements, the translation and extension to the system requirements needs to be done: “What must the product do?” This is also the transition from marketing to engineering. This set of requirements must be complete and consistent. Make sure they are formulated as SMART (Specific – Measurable – Attainable – Relevant – Time-bound). Formulate the constraints to the design as well: costs (of product, tools & project), legislation, conditions of use, production process, standardisation, design, etc. This is a very important phase in the development process, which strongly determines the result. Adaptations to the requirements still have a low impact on budget and planning, while the effect on the design can be high.

Phase 2 – Generation of a number of concepts

Several concepts are generated in a creative process and compared against a limited list of primary requirements.



4 The V-model.

- User requirements: What does the user need? Determine the stakeholders, use scenarios, user functions and constraints.
- System requirements: What must the product do? Determine system functions and constraints with testable requirements.
- Concept design: Generate and evaluate possible concepts; choose and elaborate best concept to proven feasibility.
- Detailed design: Design and dimensioning of chosen concept to component level of hardware, and module level of software.
- Realisation: From drawings, diagrams and software architecture to testable subsystems.
- Subsystem testing (verification): Test the behaviour of separate subsystems at their interfaces. Subsystems are often mono-disciplinary: mechanical, electronic, software.
- Integration testing (verification): Do the subsystems work together as intended?
- System testing (verification): Test functions and constraints of integrated system against requirements.
- User testing (validation): Are the needs of the user in his/her application fulfilled? This is also called acceptance testing.

Tools that can be used in this phase are a morphological overview and a table of comparison. Some calculations are often needed to prove the feasibility of a concept. The compliance with the other requirements needs to be checked. From this follows the choice of the most promising concept. This phase is closed by a concept design review, where the results are presented and discussed.

Phase 3 – Realisation of the design

The chosen concept is elaborated to a detailed design, and documented in models, drawings, diagrams and purchase specifications. The parts are purchased or produced with this documentation. All hardware is assembled. If something does not fit properly this is modified and the documentation is updated. Parallel to this hardware process, the software is developed and implemented, from modules to code.

Phase 4 – Testing of the system with respect to the requirements

The subsystems are often verified first, then the complete system, and finally the validation in the customer process is done. Validation means determining the value of the system for the user.

The V-model

The V-model is a much-used and well-structured method for the development of products in which hardware (mechanical or electrical) plays an important role, like mechatronics. Figure 4 makes the name V-model directly clear. The left leg of the V is development, the lower end is realisation, and the right leg is testing. Moving downwards in the V means decomposition from system to subsystem to part level. The four basic phases mentioned above are included, with ‘Requirements’, ‘Design’ and ‘Testing’ divided into several blocks.

A strong advantage of the V-model compared to other methods lies in the horizontal relationship between the development of decomposed system elements and the acceptance criteria that are derived from the subsystem specification. These acceptance criteria will determine the acceptance of the subsystem before it is admitted in the next-level assembly. The other methods leave acceptance criteria more or less open. Practice of development of complex systems teaches us that anything that can go wrong on a subsystem level will be found once the system is assembled. Debugging at that stage can be costly, time-consuming and eventually disappoint the customer.

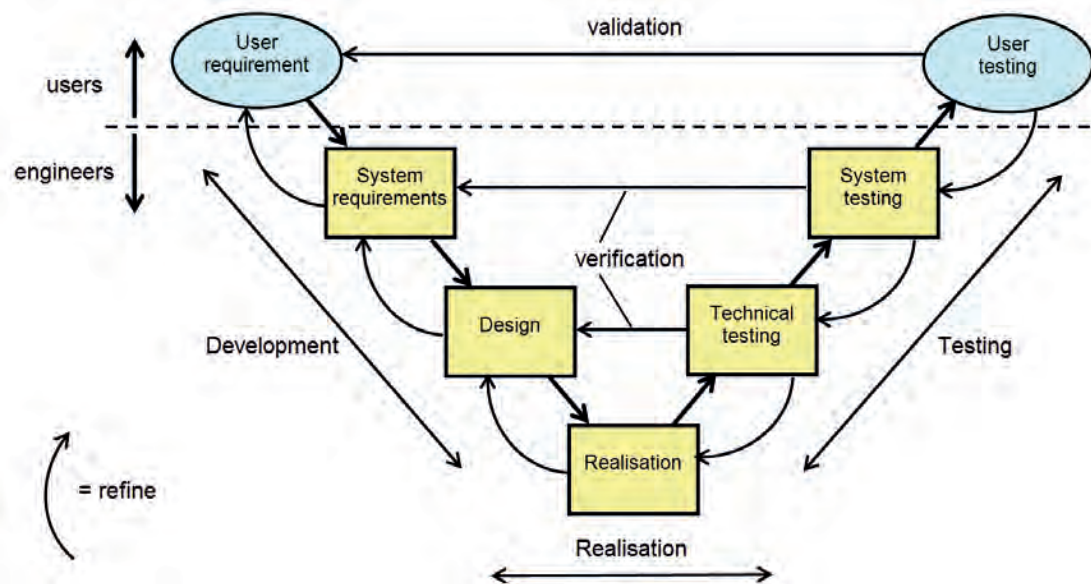


Figure 4 shows nine blocks, with a separation line between the system and multiple subsystems. These subsystems are developed in parallel in the lower part of the V. Horizontal arrows indicate verification and validation. These compare the results at different levels with the requirements (verification) and the use (validation). In Figure 4, the arrows are drawn to the left to indicate the reference for the testing. INCOSE draws them either to the right or bi-directional (Figure 3a), but the meaning is the same.

The V-model has proven to be a successful method and is well-suited to get acquainted with SE for both SMEs and higher education. Every phase results in a specific intermediate product, which is very useful for planning and assessment of educational projects. However, there is a significant risk that the process becomes document-driven, which could hamper creativity and therefore innovation. The documents are a means, not a goal!

Sequential or iterative?

Here, a distinction has to be made between the structure and the way of working. The structure of the process described above appears to be sequential. It has a direction and a goal: from user need to product. It can be divided in a number of well-defined steps. This is the way of thinking of the manager. But the actual way of working of the engineer is more iterative. There must be room for an alternation between creativity and analysis. This especially applies to more innovative developments, where the outcome is more or less uncertain.

All SE methods must allow and merge this sequential and iterative thinking, to unite manager and engineer. In some

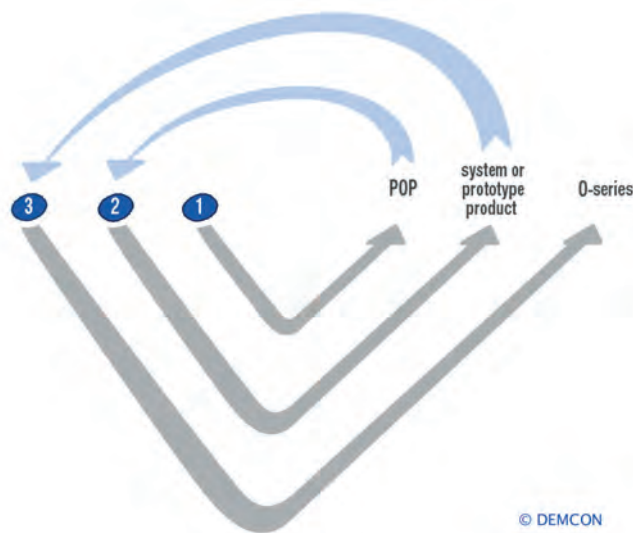
methods, like Spiral Design (Figure 3c), iterations are the core of the process. An objection sometimes made against the V-model is that it is (too) sequential, which is actually not true: the V-model was initially conceived as a combination of the linear 'waterfall' model and the spiral model.

The validation and verification arrows form several inner loops, as shown in Figure 4, but actually doing a redesign must be avoided where possible. The smaller the loop, the better it is. Most loops even take place within a phase: one must often elaborate a solution to some extent to be sure it will fulfil the requirements. If not, one must take a step back and try something else. But iterations may also take place between subsequent phases, as shown in Figure 5:

- While determining the system requirements, it may be discovered that more information is needed from the customer.
- During concept design, it may be concluded that the system requirements are incomplete.
- During detailed design, flaws in the concept may be discovered and fixed.
- While documenting, it may be necessary to improve the detailed design.
- During production and assembly, the product documentation may have to be corrected.
- During testing, the assembly and integration may have to be optimised (also of the software).

A special note can be made on the fusion of Agile software development, which is a stepwise incremental approach [4], and the V-model. The subsystem development in Figure 4 consists of three steps: detailed design, realisation and subsystem testing. For hardware development the

5 V-model, with iteration loops between subsequent phases.



© DEMCON

‘verification’ arrow does not imply that a loop is made. This would happen only when such a large error was discovered during subsystem testing that it necessitated a (partial) redesign.

6 Triple V-model (Demcon [7]).

For a software subsystem these same three steps are named design – implementation – testing. And these can be very well repeated in a number of incremental loops according to the Agile principles, adding functions one by one, in parallel with one ‘pass’ of the hardware development. This

is the way in which mechatronic development with the V-model is implemented at the mechatronics company Demcon [7].

Taking the combination of iterative and sequential approach one step further, the complete V can be passed through several times (Figure 6). Especially for more innovative products this leads to the triple V-model [7], with proof-of-principle, prototype and 0-series. These can also be characterised as research – development – engineering.

In the first (inner) ‘proof-of-principle’ loop of the triple V-model the main goal is to prove the feasibility of the main function(s), in order to reduce the technical risks. Usually this is done with a laboratory set-up using standard components where possible.

When this has been successfully tested the second loop is entered: to build a prototype that incorporates all functions, requirements and design features of the final system as much as possible.

In the third and outer ‘0-series’ loop some modifications from prototype testing may be made, changes may be made to improve manufacturability, and tooling for production and calibration is developed.

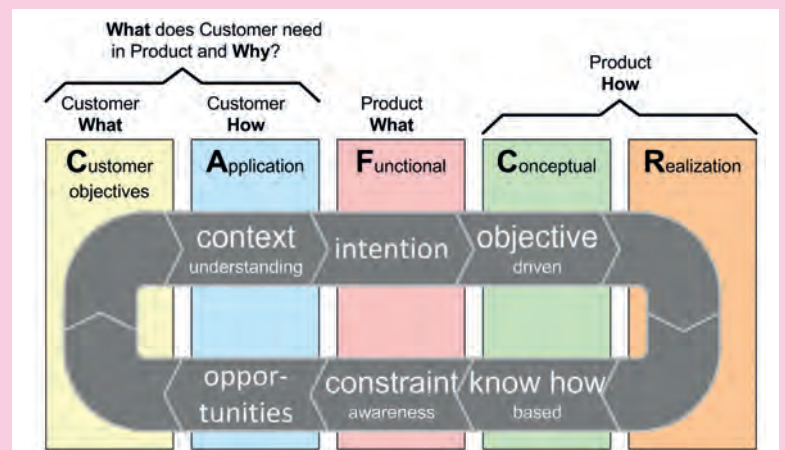
During the three loops the risks decrease, the number of requirements increases, the design freedom is reduced. This process can be depicted as three nested Vs for compact visualisation.

CAFCR

CAFCR is a method that has been developed by Gerrit Muller [8], connected to TNO-ESI. It is the acronym for the steps Customer – Application – Function – Concept – Realisation (Figure 7). It is, again, an iterative process that needs to be repeated a number of times during the definition phase to achieve a valuable, usable and feasible product. The functions defined in this loop are the pivotal point between the user (What?) and the product (How?).

It does not matter what the starting point in the loop is, as long as the correct chain of questions is asked. Who is the customer? What is his application? What is the function of the product in the application? With what concept(s) can this be achieved? Is this concept feasible? What are the constraints? Are there new opportunities?

This iterative CAFCR-process is a good method to determine a complete and consistent set of functions of the product that is to be developed, together with the customer. It also aims to find a set of key drivers for the use of the product. It uses techniques like interviewing, storytelling and creating use scenarios.



7 The CAFCR-model (Gerrit Muller, TNO-ESI [8]).

The aim must be to limit the amount of iterations as much as possible, and also the number of phases involved. Iterations between the loops in the triple V-model from outside inwards are certainly not desirable. It sounds contradictory, but a way to achieve this is to go through the complete development process up to the realisation several times in the mind during the determination of the system requirements. This is the essence of the CAFCR model of TNO-ESI; see the box.

SE in education

What becomes more important in enterprises must also get more attention in education. Working together in projects can only be learned in practice. Therefore the execution of group projects is becoming an increasingly important part of student training. Preferably, these projects should be performed by multidisciplinary teams, with students from different disciplines. This accounts in particular for students in mechatronics, who often form the link between mechanics, electronics and software engineering. SE will become a competence just as important as knowledge of,

for example, machine components, programming and control engineering.

At all universities of applied sciences involved in the RAAK-MKB project, educational tracks are being set up in which working in projects is an important learning goal. This includes both Systems Engineering and Project Management. Tutorial material is being developed. By using this in student projects it can be improved iteratively. But the importance of this educational track is not yet recognised everywhere; better integration is necessary. ■

REFERENCES

- [1] www.regieorgaan-sia.nl
- [2] International Council On Systems Engineering, www.incose.org
- [3] Federal Highway Administration, www.fhwa.dot.gov
- [4] www.mountangoatsoftware.com
- [5] www.hyperthot.com
- [6] www.bulsuk.com
- [7] www.demcon.nl
- [8] www.gaudisite.nl



MIKRONIEK

Mikroniek is the professional journal on precision engineering and the official organ of the DSPE, The Dutch Society for Precision Engineering.

Mikroniek provides current information about technical developments in the fields of mechanics, optics and electronics and appears six times a year.

Subscribers are designers, engineers, scientists, researchers, entrepreneurs and managers in the area of precision engineering, precision mechanics, mechatronics and high tech industry. Mikroniek is the only professional journal in Europe that specifically focuses on technicians of all levels who are working in the field of precision technology.

Publication dates 2015

nr.:	deadline:	publication:	theme (with reservation):
3.	22-05-2015	26-06-2015	Optics & Optomechanics
4.	31-07-2015	04-09-2015	
5.	18-09-2015	23-10-2015	issue before the Precision Fair
6.	06-11-2015	11-12-2015	Robotics

For questions about advertising please contact Gerrit Kulsdom

Dial: 00 31(0)229-211 211 ■ E-mail: gerrit@salesandservices.nl