INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

SPALTEN PROBLEM SOLVING METHODOLOGY IN THE PRODUCT DEVELOPMENT

A. Albers, N. Burkardt, M. Meboldt, M. Saak

Keywords: methodology, development process, troubleshooting, management, system analysis, systems Engineering

1 Introduction

Product development is characterised by numerous problems, for which at the time of their appearance there are often no solutions. In this context the entire product development process as well as its individual phases can be considered as a problem. The required transferring of the actual condition into the desired target condition depends on various influencing factors such as type of problem, situation, time, person, information, complexity and extent. This leads to different degrees of complexity as far as problems and their solutions are concerned. These problems are to be dealt with using the SPALTEN methodology in order to realise the phases of product development as well as the overall aim in the form of a marketable product.

1.1 Problem definition in Product Development

Being one of its key elements, the term “problem” is of special importance in the context of product development. In everyday language simple exercises and complicated tasks are both termed “problems”. Schlicksupp [1] uses the term to classify an operation as a “search problem”, “selection problem”, “analysis problem”, “constellation problem” or “consequence problem”. With so many different uses, a common description is necessary. However, no abstract description of the term “problem” is found in literature. Dette [2] introduces a broad spectrum of different problem definitions in a morphological chart (Figure 1)

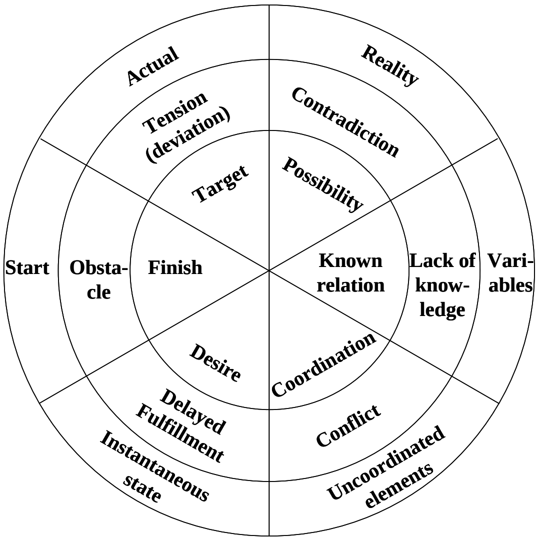


Figure 1: Problem definition (Morphological Chart) [2]

According to Dette, problems occur when forces meet resistances. A problem comprises a thesis and an antithesis, and the solution to it corresponds to the synthesis of the two.

In decision methodology, the term “problem” is used when acquiring information on how to achieve the target is somewhat difficult [3] and not obvious. One can interpret this as a gap between the Target- and the Actual State.

In mechanical engineering the term “problem” is mainly used with respect to the definition of different states [4]. Pahl / Beitz [5], Ehrlenspiel [6] and Lindemann [7] define a problem as an undesired initial state that has to be brought into a final state. To achieve this, an obstacle has to be surmounted which can occur in form of missing or unknown tools, existing or insufficiently concise goals and restrictions. Several other sources refer to these states as Target State and Actual State. Daenzer / Huber [8] thus define a problem as the difference between an existing and tangible Actual State on the one hand, and the Target State on the other hand. Similar definitions are found in Schweizer [9] and Johansson [10].

The analysis of different definitions shows that a problem can always be described by three components (Figure 2):

Initial State, Actual State

Final State, Target State

Obstacles and differences to overcome

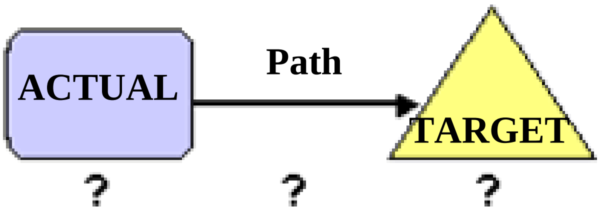
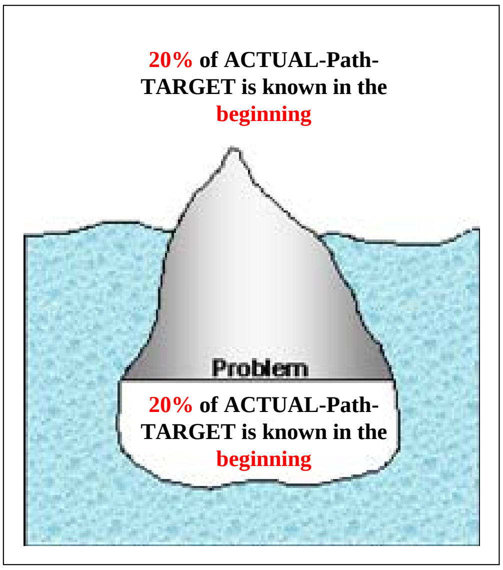


Figure 2: Components of a problem

“A problem is a deviation between the arbitrarily little known initial state (Actual State) and the desired arbitrarily vague final state (Target State), linked with the partially unknown path from the Actual to the Target State.”

The Actual State and the initial state are not considered undesired according to this definition. They represent the current, assumed, expected and real state and are to be seen rather positively as a chance to improve and optimize persistent states, or as a “kick-off” for planning. An Actual State is not necessarily random but can also be initiated, and its level of detail can exist and be described with varying vagueness. In practice, the level of detail and absolute knowledge of the Actual State depend on available resources and time.

The Target State describes the final state and the results that are to be achieved by processing the problem, with varying clarity. The final state can be formulated – similarly to the initial state – either in an exact and precise manner, or due to a lack of information and insight very vaguely. This situation occurs when the cause of a problem is to be eliminated without having a proven hypothesis of it.


20% of ACTUAL-Path-TARGET is known in the beginning
Problem
20% of ACTUAL-Path-TARGET is known in the beginning

The obstacle that has to be surmounted in order to get from the Actual State to the Target State lies on a partially unknown path. In the beginning of a problem solving process, according to the Iceberg formula only 20 % of the path from the Actual to the Target State are known (Figure 3). The main part of the solution approach is concealed like the bottom of an iceberg is, and has to be worked out or made available. A lack of skills may result from insufficient competence in situation analysis methodologies, solution finding, and social / professional areas.

Figure 3: Iceberg Rule for problem solving

1.2 Expanding the Problem Definition on Systems Engineering

The broader general definition of a problem can be transferred into the field of systems engineering. The Actual State represents the final system that is to be defined, while the Target State can be regarded as the object system that is to be achieved. The operating system refers to the path from the Actual State to the Target State and comprises structured activities such as humans, material and actions [11]. In systems engineering, therefore, a problem can be described as follows:

“A problem in systems engineering is a deviation between the arbitrarily little known system of objectives and a chosen arbitrarily vague object system, linked with the partially unknown operating system from objectives to object.”

1.3 Problem solving process (SPALTEN)

The SPALTEN-Methodology (*spalten = to split, to decompose*) provides an universal approach to handle problems of different boundary conditions and levels of complexity. With its use time and effort can be kept as low as possible and additionally to an optimal solution, maximization of safety can be achieved in problem solving. This methodology is based on research results and scientific studies in problem solving and systems engineering [12]. The general problem solving process – consisting of problem analysis, problem definition, system synthesis, system analysis, evaluation and decision – as described in VDI-Guideline 2221 [13] (Figure 4) is integrated in the SPALTEN-Methodology and extended by the work steps (modules) “Implementation” and “Recapitulate/Learn”.

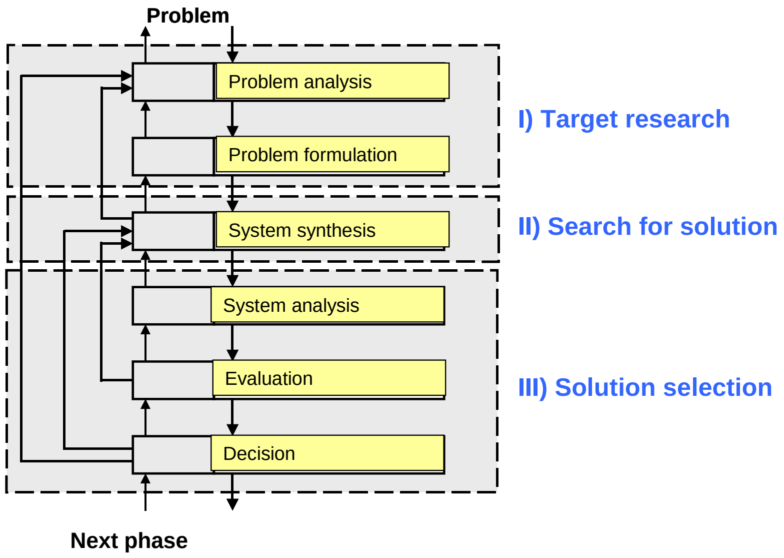


Figure 4: Systematic problem solving cycles following VDI-Guideline2221

Transferring the general problem solving process and the systematic problem solving cycle into the overall product genesis and/or a state in the life cycle of a system is done by a series connection of the cycles. This complete and universal applicability of the problem solving methodology in different phases of product genesis is a key factor of the SPALTEN-Methodology and hence to be regarded clearly.

The SPALTEN-Methodology is well suited for both future-oriented as well as spontaneously occurring problems. Depending on the task, SPALTEN has to be applied using different levels of concretion and abstraction, concretion in this case referring not only to the entire process, but also to its single phases respectively. An optimal benefit/effort ratio can be gained through the problem-oriented approach. Processing the problem using the SPALTEN-Methodology is facilitated by the integration of a handful of tools and methods of development processes. These tools and the approach are to be applied pragmatically in dependence of the boundary conditions rather than dogmatically.

1.4 The SPALTEN process

The basic structure of the SPALTEN-Methodology consists of seven modules, each of which can be individually worked on either sequentially or dynamically (event-oriented) in the course of the problem solving process (Figure 5) [14].

▪ Situation Analysis	(Situationsanalyse)
▪ Problem Containment	(Problemeingrenzung)
▪ Search for Alternative Solutions	(Alternative Lösungssuche)
▪ Selection of Solutions	(Lösungsauswahl)
▪ Analysis of the Level of Fulfillment	(Tragweitenanalyse)
▪ Make Decision/Implement	(Entscheiden/Umsetzen)
▪ Recapitulate/Learn	(Nacharbeiten/Lernen)

In the following, each step of the SPALTEN-Methodology is explained briefly:

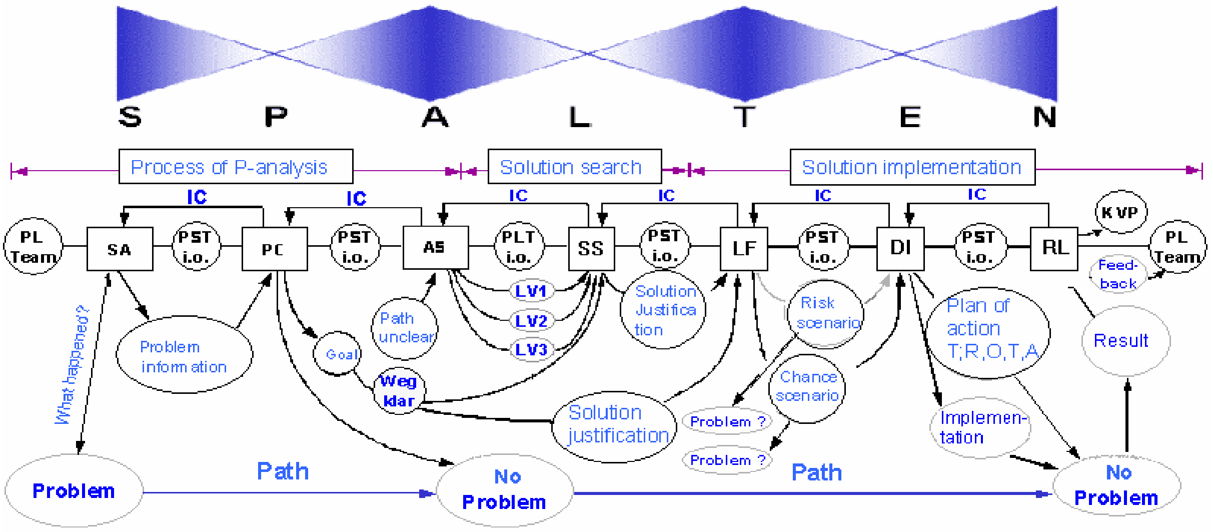


Figure 5: Honeycomb model and steps in systematic problem solving

SA – Situation Analysis

In the first module of the SPALTEN-Methodology the situation is grasped and its details are revealed. Data is collected, structured and documented, and the information then becomes basis of subsequent steps. In addition, the situation is assessed and a problem solving approach is chosen. An increase in the amount of information is characteristic for situation analysis and the change of plenitude comparable to a downward funnel.

PC – Problem Containment

This module is a means of focusing only on data that is relevant in problem solving. The amount of information is hence narrowed down and contained. An alternating increase and decrease in the amount of information on hand is referred to as the “Honeycomb model” of the SPALTEN-Methodology. Problem containment aims at identifying cause and effect of the deviation between the Target and the Actual State. This deviation is to be proven and the subsequent procedure such as further processing or stopping to process the problem is defined. The Actual-, (Not)Actual- and the Target State are concretized and precisely described. Possible causes of the deviation can be identified and hypotheses can be made. By the means of facts, simulation and experiments, or by focusing on the most relevant information, the hypothesis can be verified. As a result, this module determines the goals of problem solving and provides a more detailed description of the problem to be solved by means of a requirements list or other methods.

AS - Alternative Solutions

The aim of searching for alternative solutions is the development of solutions of the given task. In this module boundary conditions for solutions research, and an abstract wording of the aim – in order to avoid potential fixation on certain solutions – are established. Generated solutions are further analyzed, concretized, documented and recorded. This phase of problem solving is characterized by a necessarily high creativity potential of participants. The plenitude of information increases once again.

SS – Selection of Solutions

Solutions that have been developed up to this stage are analyzed, evaluated and the most promising ones are selected. Therefore evaluation criteria have to be defined and weighted. As an additional step the selected solutions are examined with respect to safety and their influence on the selection of solutions. The plenitude of information on hand is decreased by repeated focusing on the most promising solution.

LF – Level of Fulfillment

Predictable risks and opportunities of the selected solutions are investigated in this module. Therefore critical spots are revealed, which facilitate the determination of risks/opportunities, causes, and identifying causes as such. An insight into the aforementioned allows to take action in order to avoid risks and to realize opportunities. The aim of the module, hence, is the implementation of these actions in the problem solving process. The plenitude of information is again increased by the multitude of risks and opportunities and the implementation of these.

DI – Make Decision/Implement

In this module, results of the foregoing two modules SS and LF are gathered and processed. The decision to implement the solution to the problem is documented and the implementation itself can be divided in three phases:

- Planning
- Carrying out
- Finishing

The goal of implementation is a complete processing and realization of the selected solution(s) in association with the integration of the previously determined actions for minimizing risks and optimizing opportunities. This module narrows the amount of information down.

RL – Recapitulate/Learn

The last module is based on the continuous improvement process (CIP). The aim of the module is to extract fundamental knowledge and optimized approaches from the experience made while processing the problem, for future reference in problem solving processes. Documentation of this improvement potential once more increases the plenitude of information and moves toward the opening of the information funnel of the honeycomb model

Using the SPALTEN-Methodology – depending on the task – does not always require all modules, which may result in saving time. This way, through cause analysis directly following problem containment (PC) the origin of the problem may be detected ahead of time (for instance in the case of flawed components leading to an inability to deliver the product). The origin of the problem may of personal, company-related, supplier-related or associate-related character. If the origin of the problem lies with a supplier, the task can be assigned to them. In case of a personal character of the problem's origin, the solution does not necessitate a repeated process of solution research and selection, but a suitable way to monitor the supply of flawless parts. Beside this scenario two more cases can occur. The cause of the problem may be unclear, which has to be resolved by problem containment and subsequent search for alternative solutions. Alternatively, cause and path may be known and one or more solutions may be available, which does not call for searching for alternative solutions. In this case it is advisable to continue by selecting a solution. By skipping the "search for alternative solutions"-module a shorter processing time and an optimal utilization of resources can be achieved.

A distinctive feature of the SPALTEN-Methodology (cp. Figure 5) is the so-called information check (IC) and the step of determining the problem solving team (PST), which are both carried out between every two modules. Information check investigates if the available fundamental information is sufficient for processing the next module, and if information has been utilized well enough. An important aspect is weighting the benefit/effort ratio and the adequacy of the level of detail of the process step in question, with respect to the overall task. It does not make sense, for instance, to work on problem in many details and with high effort, if the achievable benefit and/or the occurring drawback are only a fraction of the effort put in.

An important feature that should be made use of within the SPALTEN-Methodology in order to achieve the Target State is the situation-dependent make-up of the problem solving team. When starting to work on a module, and moving on to the next one, the team is examined with respect to the skills that are needed to handle the task. The team may be reformed completely if necessary, before it processes the module. The module-specific team make-up is of special importance to solving the problem. By selecting the team members according to specific features the team can be fitted to the underlying task. So far existing problem solving systematics are restricted to problem processing and the approach that leads to the Target State, however, they do not have any special regard to team make-up. The SPALTEN-Methodology thus achieves an increase in efficiency.

Yet another element of the SPALTEN-Methodology is the so-called problem oriented continuous information and idea record (PCIR). This database is used to continuously grasp spontaneous ideas and information related to the overall approach and to solve the problem. Recording ideas and information takes place in form of tabular documentation or in written form. A data-base-supported filing system and systematic classification of ideas and information of the PCIR has not existed prior to this research.

The event flow is mainly different because of the arrangement of the first two modules. In case of an unforeseen event there is no time for preworking a problem that has just occurred, as there are no "warning signs". In a planning scenario, however, often times the problem can be defined roughly and stated clearly in advance. The person involved in solving the problem has enough time until work on the problem commences to analyze the situation and to carry out problem containment. By doing so, goals and boundary conditions that are to be met at the end of the problem solving process can be recorded and documented with the use of a requirements list. The actual process of problem solving starts with a kick-off to search for alternative solutions.

1.5 The Product Development Process and the Problem Solving Process SPALTEN

There are several analogies between product genesis and the problem solving process. It is rather difficult to differentiate precisely between the two, since product genesis corresponds to the problem solving process from an abstract view. Starting with a situation analysis (SA) and finding the product idea (as a part of the product development process), solutions are developed (AS), that is, products are planned and designed. Finally there are implemented and manufactured in analogy to the "Recapitulate/Learn"-step of the problem solving process. Literature also refers to the processes of problem solving and product generation as microscopic and macroscopic processes. The microscopic problem solving process differs from the macroscopic process in the possibility of its application at any stage of product generation. The problem solving process is based on Dewey's problem solving logic, which focuses on finding the target, the solution, and selecting the solution. In contrast, macro-logic (product genesis) relies on the phase-model. The reason for this is that the problem solving process on its own cannot handle the utter complexity of product genesis.

The SPALTEN-Methodology falls into the category of microscopic processes, since problems of any kind – any genre and any situation -- and from any stage of genesis can be handled. SPALTEN is thus a universal methodology to solve problems.

2 Software structure of the SPALTEN-Process

Coming from the field of psychology of thought, there is knowledge about no strictly linear event flow existing in problem solving. The procedure of solving these problems must be an iterative and flexible one, which suits the state of awareness and experience, the solution advancement and the personal capability of the problem solver, in order to achieve an appropriate, successful result [15]. In the scope of the SPALTEN-Methodology, such a flexible approach can be found in the dynamic sampling and event orientation.

In order to implement this flexibility, the SPALTEN-Methodology has been developed following the concept of a level-structure as in Figure 6. Based on the approach of changing from an abstract state to a concrete one, the SPALTEN-Methodology has been subdivided systematically. The first level represents the overall problem solving process (SPALTEN), which is divided into 7 modules ranging from Situation Analysis (SA) to "Recapitulate/Learn" (RL) on level 2. These modules are complete with respect to their content and constitute one thematic work step each. On this level of the problem solving process, the modules are connected according to the event flow of the entire process.

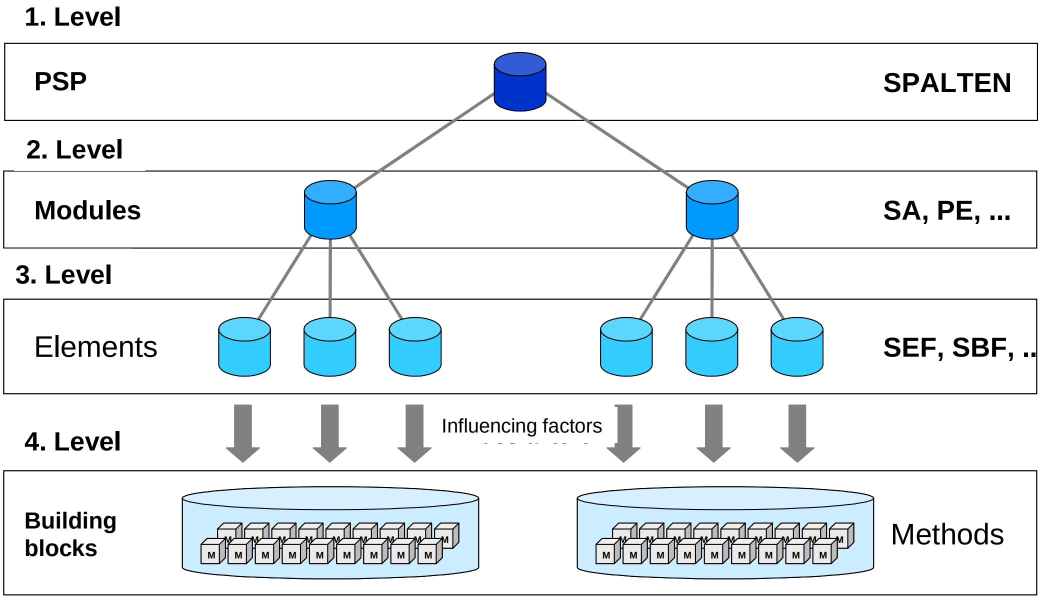


Figure 6: Level-Structure of the SPALTEN-methodology

The third level contains elements of problem solving. These elements are the actual work steps that need to be taken – in the form of instructions and approaches to solving the problem. Each element consists of text forms, which comprise methods and tools. Within the framework of the problem solving methodology there are 61 elements with a total of 604 pages of forms.

However, the elements are strictly assigned to modules. Even though with some modules of the methodology it is possible to directly access elements from other ones, these elements are still assigned to their original position. During implementation in Visual Basic 6.0 the elements were defined as stand-alone projects (*.vbp). These projects are embedded in a comprehensive Visual Basic Group Project. They are controlled by a central project (problem_solving_project.vbp) inside the project group, in order to prevent random navigation in the software environment. This structure allows to complete, interchange, modify or remove elements of the methodology, and to add new work steps. On the fourth level, additionally to the fixed methods within the forms further methods, tools and aids are provided for editing work steps in detail. These are generally referred to as building blocks. In the context of the problem solving concept large emphasis has been put on assigning these methods to corresponding elements in a flexible and dynamic way rather than in a fixed manner. An advantage – for the problem solver – compared to existing problem solving methods is that the methods construction kit is not rigid and fixed, due to the mentioned flexible way. Both the assignment and the volume of the methods construction kit are variable. Adding new methods, editing existing ones, assigning them according to influence factors and evaluating their applicability is entirely unproblematic.

This systematic structuring and modularizing provides fast and effective accessability and processability of relevant projects, despite a very complex programming structure.

On the basis of this level-structure the structure of the SPALTEN-methodology has been conceptualized and realized just like in Figure 6. Components and software tools of Microsoft Office 2000 are being used.

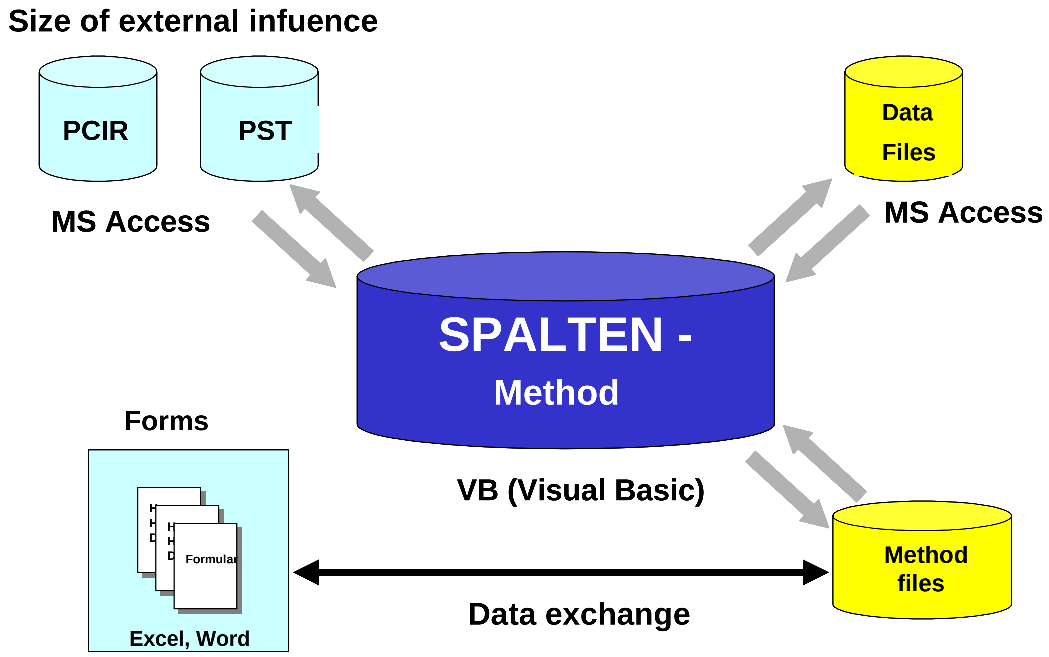


Figure 8: Structure of the SPALTEN-methodology

The central kernel of the computer-based problem solving process is the SPALTEN-methodology as can be seen in Figure 8. This kernel contains the first three levels of the level-structure and thus the working approach of problem solving. It was programmed – as mentioned before – using Visual Basic 6.0. Around the kernel three more functional areas can be found, which interact and exchange information with it.

One of the domains is the data file, which was programmed using Microsoft Access 2000. The function of this building block is to store the input of the problem solver with regard to the problem solving project. This input can be e.g. data about the storage of forms, indications of processed tasks or data that has been entered in the software by input or by activating buttons and fields. It ranges from modules or elements of the current problem solving project to problem solving processes and master processes/modules that have been worked on. Matching this database (MainDatabase.mdb) to the kernel is done continuously during work so as to minimize the loss of data that can occur due to abnormal program terminations. Further on, the methods-matrix – which contains methods and their assignments -- and the method-counter-matrix are saved and stored in the data file with the use of an additional database (Methods.mdb). This database (Methods.mdb) is activated according to the element and the use of methods from the 4th level.

Another building block is made up of the modules PCIR and PST. These modules were not integrated in the kernel, instead they were created as stand-alone programmed databases having corresponding interfaces using Microsoft Access 2000 (Access_new.mdb and CompetenceExaminor.mdb). The advantage of this concept lies in the applicability of these program parts outside the SPALTEN environment e.g. in different programs or application areas. The kernel makes use of these program parts on demand – by activation in the navigation bar (see chapter “interface”) or in the working window.

The third building block, which makes the SPALTEN-methodology complete, is the method file. Methods that are depicted on the 4th level are systematically placed here and stored. Each method makes up an independent building block just like the modules PCIR and PST.

The methods were created on the basis of Visual Basic 6.0 and can be accessed at any time, also outside the SPALTEN framework. To facilitate simple and continuous editing of the methods, and to provide high flexibility, they can access forms that were created in Excel, Word or other random formats. The method-building-blocks are variable to such an extent that forms from other software tools (such as a template in Mind Manager) can be integrated. Proper functioning of the module, however, can only be provided if the corresponding software (e.g. Mind Manager) is to be found on the computer. Interaction with the kernel takes place according to the activation of methods from off the kernel.

references

[1] Schlicksupp, Helmut: Kreative Ideenfindung in der Unternehmung: Methoden und Modelle. Berlin : Walter de Gruyter Verlag, 1977

[2] Dette, Klaus: Ideenentwicklungsmethoden: Untersuchung eines potentiellen Planungsinstruments. Berlin, Universität, Fachbereich Gesellschafts- und Planungswissenschaften, Diss., 1976

[3] Brauchlin, Emil: Problemlösungs- und entscheidungsmethodik: Eine Einführung. Bern : Verlag Paul Haupt; 1978

[4] Dörner, Dietrich: Problemlösung als Informationsverarbeitung. Stuttgart : Kohlhammer, 1976

[5] Pahl, Gerhard ; Beitz, Wolfgang: Konstruktionslehre: Methoden und Anwendung. Berlin : Springer-Verlag, 2003

[6] Ehrlenspiel,Klaus: Integrierte Produktentwicklung: Methoden für Prozessorganisation, Produkterstellung und Konstruktion. München : Hanser Verlag, 1995

[7] Lindemann, Udo: Skriptum zur Vorlesung Methoden der Produktentwicklung. München: 2002

[8] Daenzer, Walter F. ; Huber, Fritz: Systems Engineering: Methodik und Praxis. 10. Aufl. Zürich : Verlag Indstrielle Organisation, 1999

[9] Schweizer, Peter: Systematisch Lösungen finden. Zürich : Vdf Hochhschulverlag, 1999

[10] Johansson, Björn: Kreativität und Marketing. Die Anwendung von Kreativitätstechniken im Marketingbereich. St. Gallen, Hochschule für Wirtschafts- und Sozialwissenschaften, Diss., 1978

[11] Ehrlenspiel,Klaus: Integrierte Produktentwicklung: Methoden für Prozessorganisation, Produkterstellung und Konstruktion. München : Hanser Verlag, 1995

[12] Albers, Albert: Skriptum zur Vorlesung Integrierte Produktentwicklung. Karlsruhe : Institut für Produktentwicklung Karlsruhe, 2003/2004

[13] VDI-Richtlinie 2221 Mai 1993. Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte

[14] Albers, Albert ; Saak, Marcus ; Burkardt, Norbert: Gezielte Problemlösung bei der Produktentwicklung mit Hilfe der SPALTEN-Methode. In: Technische Universität Illmenau (Veranst.): 47. Internationales Wissenschaftliches Kolloquium: Maschinenbau und Nanotechnik – Hochtechnologie des 21. Jahunderts (Ilmenau 2002). Illmenau : Kretzschmar Verlag, 2002, S.83-84

[15] VDI-Richtlinie 2221 Mai 1993. Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte.

Prof. Dr.-Ing. Dr. h. c. Albert Albers
University of Karlsruhe (TH)
IPEK ▪ Institute of Product Development
Kaiserstr.12, 76131 Karlsruhe
Germany
+49 721 608-8062
+49 721 608-6051
albers@ipek.uni-karlsruhe.de