

Chapter 19

Challenges to Digital Product and Process Development Systems at BMW

Dietmar Trippner, Stefan Rude and Andreas Schreiber

Abstract Today, the methods of model based product development are well-recognized and wide spread, at least, in the automotive industry as well as in the aerospace industry and their suppliers. But, current challenges of these industries like light weight design, electro mobility, modern mobility concepts plus those caused by rising product complexity bring this concept to its limits. An overall approach is progressively requested, which is able to continuously integrate requirements, functions, logic and physical product descriptions (RFLP). This should be possible not only for mechanical aspects but also for electronics and software development. The approach of system engineering addresses the continuous availability and linkage of product information. This concept, which is well-known in the aerospace industry for a long time, is only recently used in automotive industry. An example is the use of integrated development environments. Nonetheless, the realization of this concept in an automotive company is definitely a challenge. Examples for these problems are differently coined. Examples are detailed requirements (client requirements versus requirements to a complete vehicle and to components properties), consideration of configuration, validity and maturity, complexity management (complete vehicle to component, vertical integration, plus integration of early concept phases over development, verification, clearance to the production start-up, horizontal integration) and multi-disciplinarity (mechanics with calculation, electronics and software). The realization of systems engineering does not only create high demands to the design of the process-IT (authoring systems, TDM and PDM), but also has to consider organizational aspects (process and structure organization, integration of development partners and suppliers). Frequent acquisitions under IT system vendors, especially, in the

D. Trippner (✉)
dreiconsult GbR, Feldstr. 8, 82290 Landsberied, Germany
e-mail: dr.trippner@dreiconsult.de

S. Rude · A. Schreiber
BMW AG, Munich, Germany
e-mail: stefan.rude@bmw.de

A. Schreiber
e-mail: andreas.as.schreiber@bmw.de

CAD/PLM/CAE market as well as the selection of the systems for functional and economical aspects lead to increased requirements concerning open interfaces. In the present document, findings and experiences from the introduction of systems engineering for automotive processes are described. Effects on the process IT architecture are outlined. “Lessons learned” and necessary changes in process-IT, in form of selected examples and solution alternatives, are discussed.

Keywords Systems engineering · PDM · TDM · CAE · Simulation · RFLP · Code of PLM openness (CPO)

19.1 Introduction

Digital product and process development systems have become indispensable for the permanent advancement and new development of products from the product program of BMW. Coming from the development of aircraft engines, motorcycles and racecars, the current product program of the brands BMW, Mini and Rolls Royce is presently enhanced massively in direction of electro mobility and mobility services (Fig. 19.1).

The premium strategy is consequently pursued in the different vehicle classes. By doing so, besides the classic car bodies also new variants are offered, for example like SACs, and all variants are offered again with different extra and country specific equipment (Fig. 19.2).

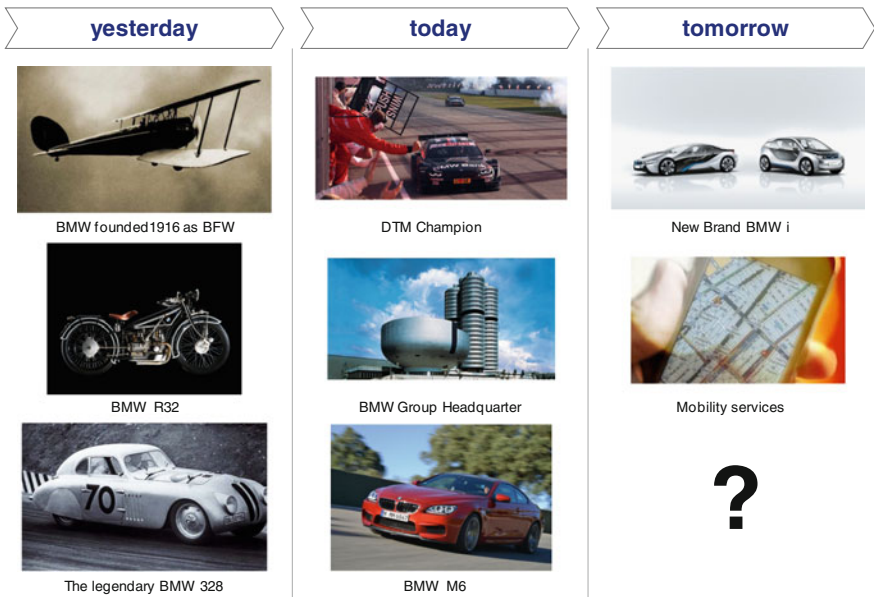


Fig. 19.1 BMW group—company portrait

Class	Limousine	Touring	Gran Turismo	Hatch	Coupé	Cabrio	Roadster	SAV	SAC	SAT
Super Luxury Class										
Upper Class										
Upper Middle Class										
Middle Class										
Compact Class										
Small Cars										

Fig. 19.2 The BMW group product portfolio

Beside the growth of the product program, the possibilities of electronics lead to exponential gains of the vehicle functions, which are more and more realized as linked, software based functionality. Secondary, to those for the client facing functions, is the multiplicity also caused by increasing safety requirements of the vehicles and the improved client service. Additionally, the realization of multi-faceted requirements, for example to meet legal regulations, is relevant for the increasing complexity (Fig. 19.3).

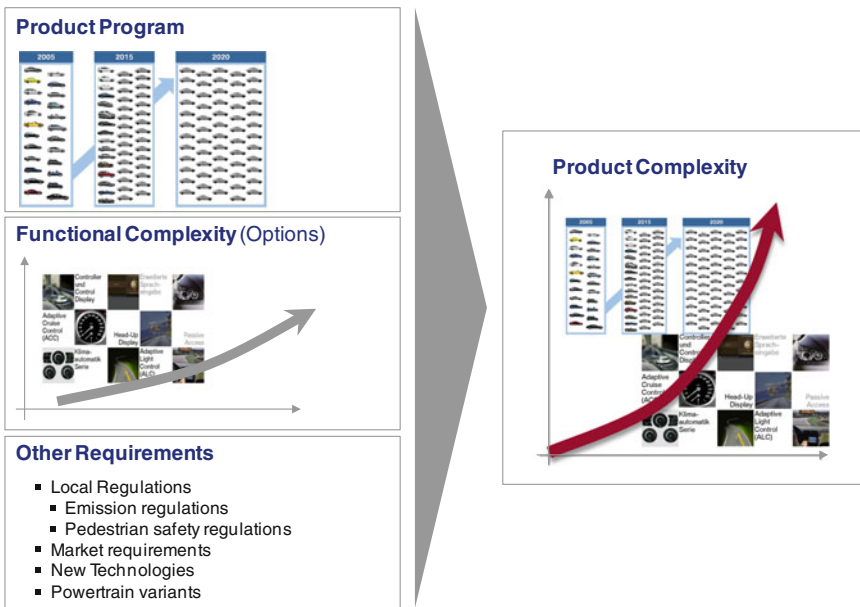


Fig. 19.3 Increased product complexity in the product development process

19.2 Requirements Caused by the Development Process

To control the multitude of requirements for the development of the complex product programs a highly optimized development methodology is necessary (Fig. 19.4).

Requirements for vehicles are described and modeled in the requirements management. The derivation of requirements from subsystems, which realize single functional groups, are going to be standard for different component or system suppliers. Still, BMW differs in geometrical design, functional design and system design. Functional design is reflecting the vehicle physics (properties like acoustics, oscillation or crash behavior), while system design is seen as the definition of the E/E systems inclusive their realization in vehicle software. Here, simulation methods already play a role, especially in the components design process.

Geometrical/functional integration as well as system integration serve for the gradual assembly of components, subsystems and complete vehicles. A gradual concept serves for the execution of necessary verification methods (test and validation) on a cost effective level. By virtual means as well as on the base of physical prototypes verification is understood as validation against specified requirements. The validation process checks against client expectations until the final approval of the complete vehicle.

As a base for the resulting design process the V-model, which is known from systems engineering, is suited (Fig. 19.5).

Development process requirements have to be defined as properties and functions. The gradual definition of subsystems and components is called architecture

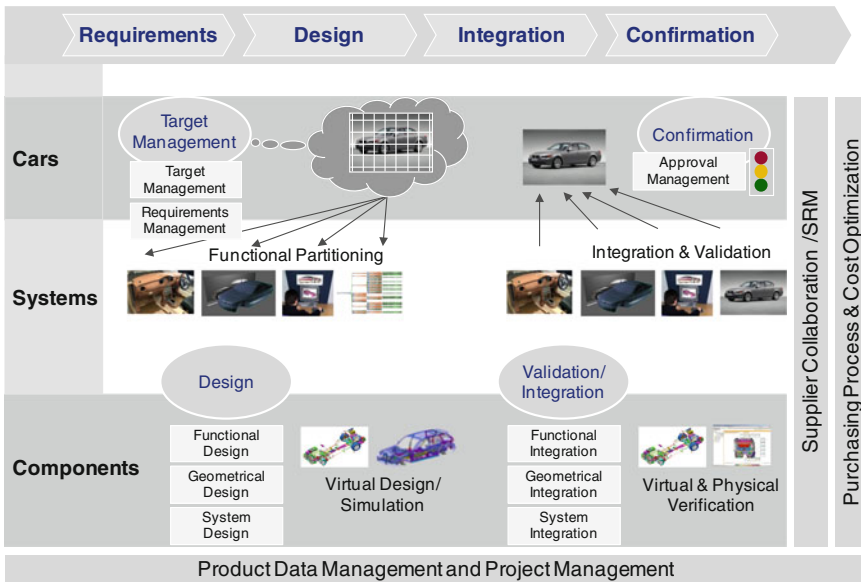
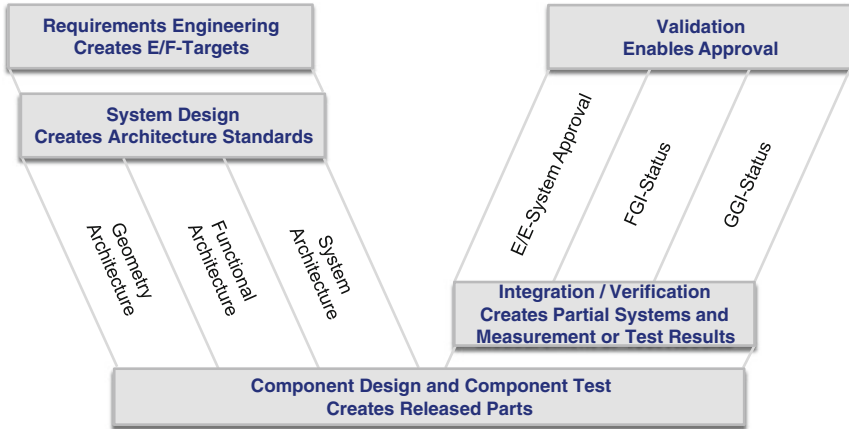


Fig. 19.4 The principles of the product development process



Legend: E / F – Properties/ Functions, E/E – Electric/ Electronic, FGI – Functional Design and Integration, GGI – Geometrical Design and Integration

Fig. 19.5 Development process in the V model

development, which has a different interpretation in different disciplines. Specifications for the component development are, normally, documented in the form of a specification sheet. The integration also happens on different levels. This counts for both the E/E system, the functional integration (validation of the physical properties) and the geometrical integration for the test of geometrical coherence. The sum of all single approvals creates the final vehicle release.

Currently the virtualization in the single disciplines has highly progressed (Fig. 19.6). The typical S-curve by GGI, FGI and SGI shows already a flattening course. However, the progressed virtualization was reached by the application of different IT-systems, at BMW for example more than 1,000 different applications. This leads to enhanced requirements for consistent integration.

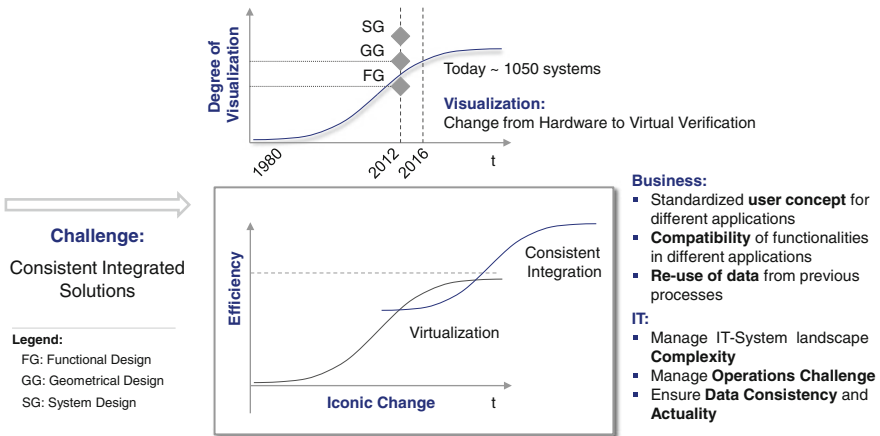


Fig. 19.6 Requirements in the product development

To realize a relevant efficiency increase in handling engineering tasks with additional tools in new application areas is going to be more and more complicated just by increasing the tool coverage. A new level of efficiency can be reached, if once created data are made available for downstream systems. Though, the downstream usage requires consistent integration among processes and used systems.

Basically, different levels of consistent integration can be distinguished (Fig. 19.7). These are the horizontal, the vertical, and the interdisciplinary integration. Additionally consistent integrated system GUIs (frontend integration) are required in the engineering workplace.

Horizontal integration is focusing the process from early requirements management up to the final product release. This can be seen in a time and a procedural sequence. Time: the amount of requirements which should be tested at the appropriate development milestone has to be known (maturity management). Procedural in the term of that the solution elements which belong to the requirements and the appropriate test cases have to be known (traceability). Vertical integration assures that requirements, solution elements and test and release procedures (divided in complete vehicle, subsystems and components) have to be modeled gradually up to a balanced structure. The interdisciplinary integration demands the unique identification of similar assemblies in the different disciplines. Finally the frontend integration limits the number of different user interfaces, the multiplicity of standard products in use and harmonizes the system interaction. The usage complexity has to be reduced, for example through role based user interfaces.

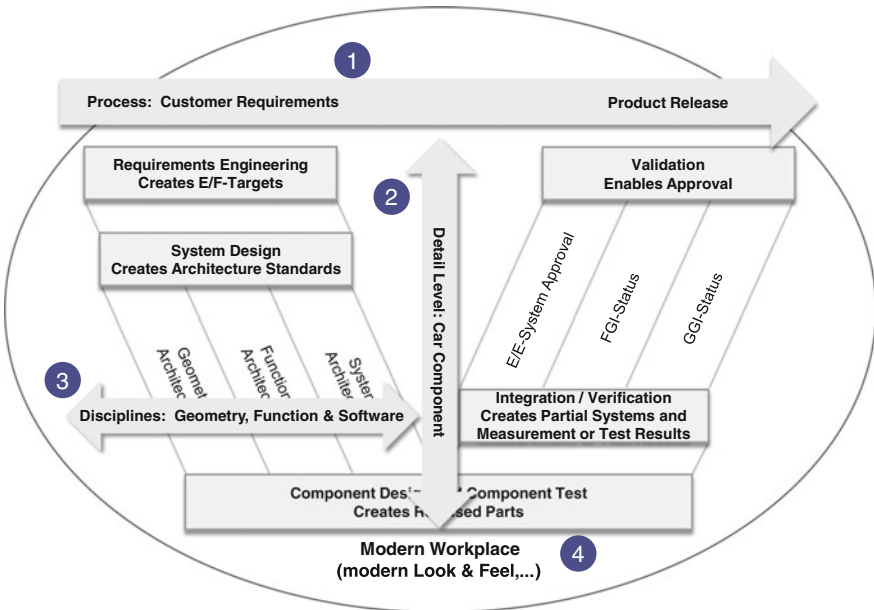


Fig. 19.7 Dimensions of integration in the V model

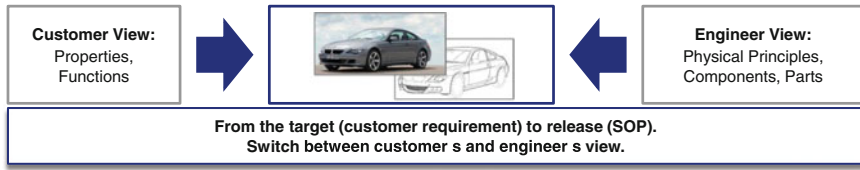


Fig. 19.8 Change between client and developer view

The horizontal integration has to interact especially with the matrix-like relations between the client view (properties and functions) and the developer view (components and parts) (Fig. 19.8). The combination of the components appropriate to a function or property is named “effect chain”. Because components or parts, normally, add something to multiple properties and functions, a matrix-like relation is generated.

Though, relation matrices arise in terms of vertical integration also on subsystem, component or on software module level each with a different level of detail. The requirements (properties and functions) change in the early phase of the development process, too. In addition, the management of a complex product program challenges the structure of the report level, which, normally, align themselves at the organizational structure of the enterprise. Because of this, the management of these always simple looking matrix relations is anything but trivial.

Modern IT-system solutions provide the RFLP-approach to present the necessary matrix structure between client and developer view. Properties and functions can visualized with requirements management systems and function modelers. Thereby, the client view can be modeled. For the developer view methods are required which show logical structures (behavior models, block diagrams for draft models) and physical structures (CAD-models, CAE-models and IT systems to manage the prototype parts) (Fig. 19.9).

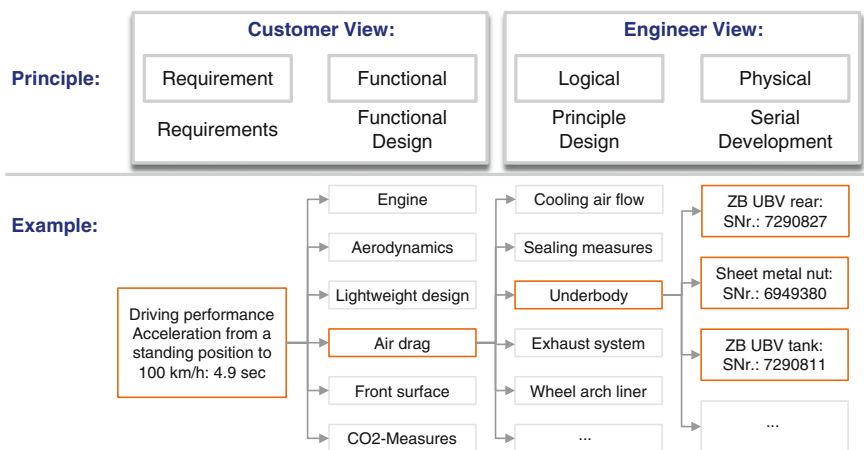


Fig. 19.9 Client and developer view—the RFLP-approach

19.3 System-Level Integration

To meet the introduced demands for integration requires an adjustment of the system scenery which is used in the product design process, especially, concerning functionality and interfaces. Figure 19.10 shows again the dimensions of integration. In this figure is also a classification of the requirements and the solution description or rather the coverage (test cases) in the V model done. Each of these as pyramids illustrated disciplines can be divided in the elements mechanic, mechatronic and software. It is important to know, that the discipline integration is not sequential and uni-directional. Instead a bi-directional and continuous integration scenario needs to be implemented. This is shown in Fig. 19.10 by a sinuous line. This interaction scenario happens on every level between all engaged disciplines.

Today, targets, requirements, functional aspects, logical connections and, finally, the geometry are managed in a heterogeneous system landscape. The demands of integrating this information is not new—but, is often met with proprietary solutions, single interfaces or by using excel spreadsheets. This approach isn't sufficient anymore for the requirements addressed in Fig. 19.11 on product and function-diversity.

General requirements have to be met before starting the system level realization, such as unified ordinal structures which are organizing all development data. These structures are ideally aligned to end customer functions. Furthermore, in the system design process the integration aspect needs to be considered. This can happen for instance through the use of standardized data models and the availability of interfaces [see code of PLM openness (CPO)]. For example it has to be possible to find

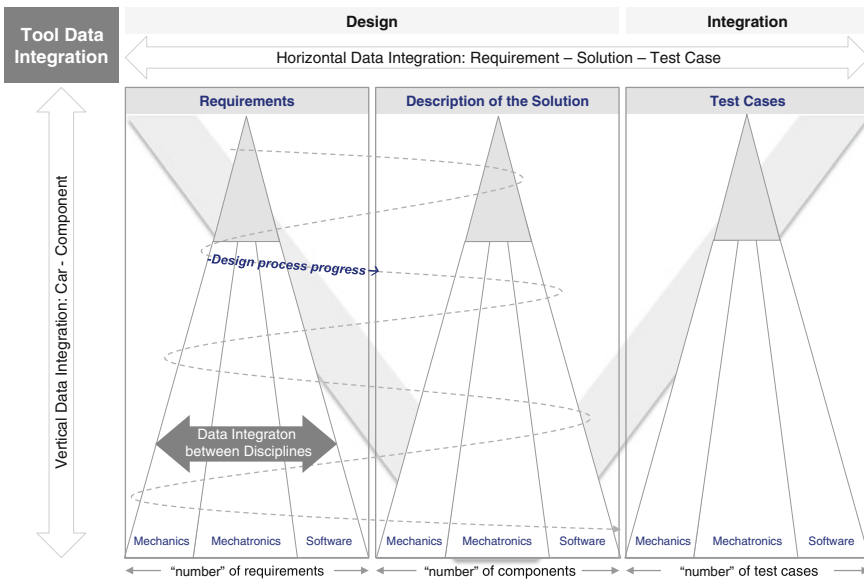


Fig. 19.10 The dimensions of consistent integration

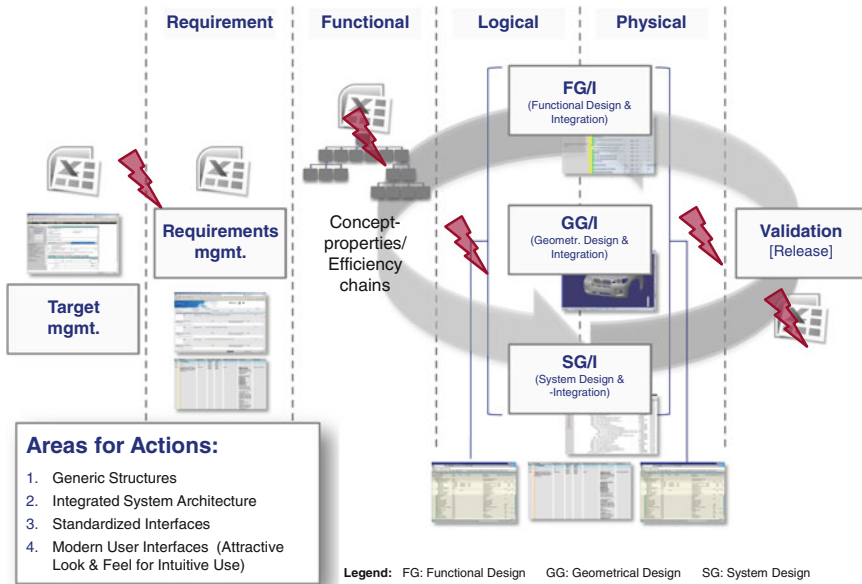


Fig. 19.11 Consistent integration: requirements

for a given validation case the relevant part versions. It has to be also possible to reach for every part or part version the requirements based on which the part has been developed.

19.4 Frontend Integration (Harmonized User Interfaces)

One possibility to realize these requirements are integrated/harmonized graphical user interfaces (see Fig. 19.12). Today, BMW uses a multiplicity of systems in the product development process. Each of these systems has its own graphical interface, often with complete different logical and graphical display options. The ordinary user, who works with up to 30 different systems, is exposed to an unacceptable complexity.

Because of that, different BMW projects address the design of an integrated work environment. The engineer does not need to be aware that the visualized data are originated from different systems, because he always works with an interface that consists mainly of the following components:

- “Google-like” search with a browser like interface to search over different data sources (for example PDM/TDM systems, requirements management, quality and maintenance systems). The browser also integrates the eBOM and product structure management as well as visualization tools.

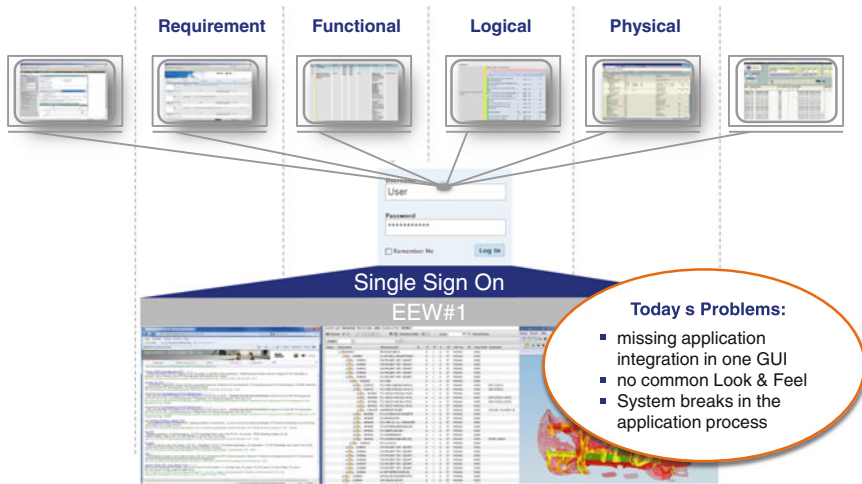


Fig. 19.12 Consistent user-interface

- eBOM and product structure navigator
- Visualization tool, to visualize the products and components in every stage of development. The visualization tool is going to be used also as a reporting tool (Visual Reporting) for the presentation of different parameters (weight, level of maturity, quality, guarantee incidents).

19.5 Consistent Integration in System Design

To understand the problem in the context of the whole product development process, it is necessary (Fig. 19.13):

- To align the process architecture map at the V model to harmonize cross discipline processes. Communication takes place over clearly defined working structures. The integration of results happens at synchronization points.
- To measure the target system architecture with the integration criteria: the relevant RFLP data have to be at least exchangeable over consistent interfaces or better already to be managed in a homogeneous infrastructure (TDM/PDM backbone). A consistent product structure has to be established.
- The concept of integration enables the consolidation of processes, systems and data with the objective of the availability of highly cross-linked development data over the product development process (Fig. 19.14). This integration has to happen both on the organizational level (clear communication rules by the use of V model aligned processes) as well as on the technical level (consistent interfaces and data models). To achieve the integration potential on system level

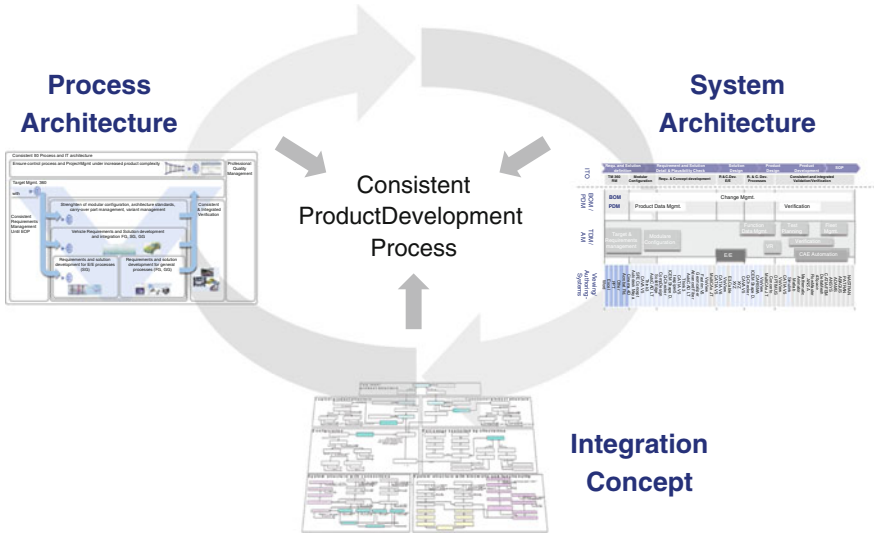


Fig. 19.13 Corporate strategy of the ITO architecture

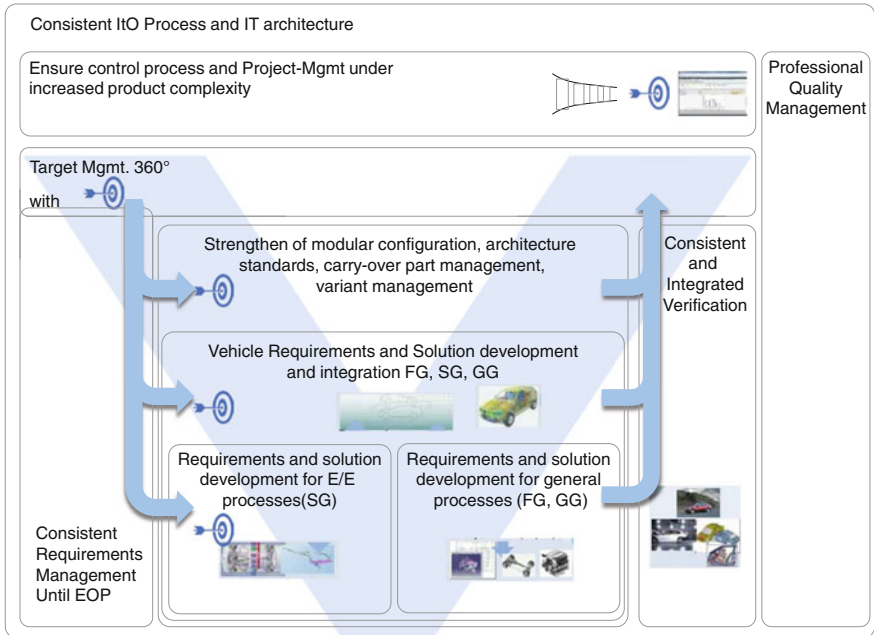


Fig. 19.14 Simplified example for the target process map

the “CPO” has taken a major role to improve the constructive collaboration between users and IT systems vendors since 2012.

Figure 19.15 shows BMW’s simplified present-day infrastructure in the ITO process (Idea to Offer, synonym for the product development process). The homogeneous architecture on the BOM/PDM level follows a very heterogeneous infrastructure on the team data management (TDM)—level. Approx. 1,000 different authoring systems are presently managed by approx. 40 TDM—Systems. The TDM architecture is very heterogeneous and incomplete, often only excel- or file-based implemented.

On one hand, the multiplicity of authoring systems is necessary to sustain the access to new technologies and developments. On the other hand, redundancy has to be avoided to reduce IT architecture complexity. The location of the applications respective to the implemented functionalities in the V model helps to identify and to fix these redundancies with adequate IT architecture measures.

The administration of authoring systems, especially the created product data, needs a re-organization with the objective of a significant reduction of the number of used systems, to reach the introduced consistent integration objective.

Figure 19.16 shows the vision of such a modification on the TDM level. Basic functionalities (geometry data-management, visualization, workflow) are realized with a TDM backbone. Based on the TDM backbone enhanced functionalities like

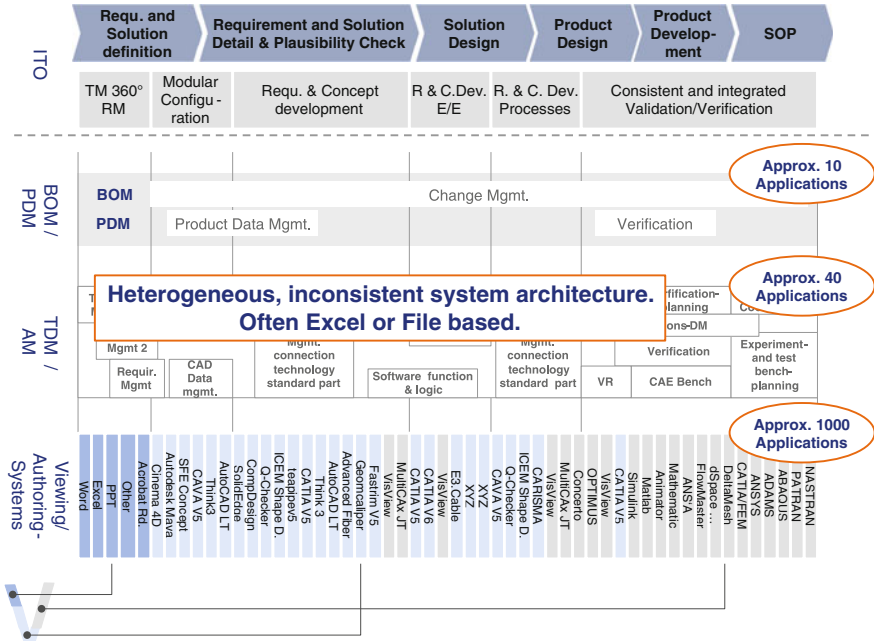


Fig. 19.15 Systems in the product development process (example BMW)

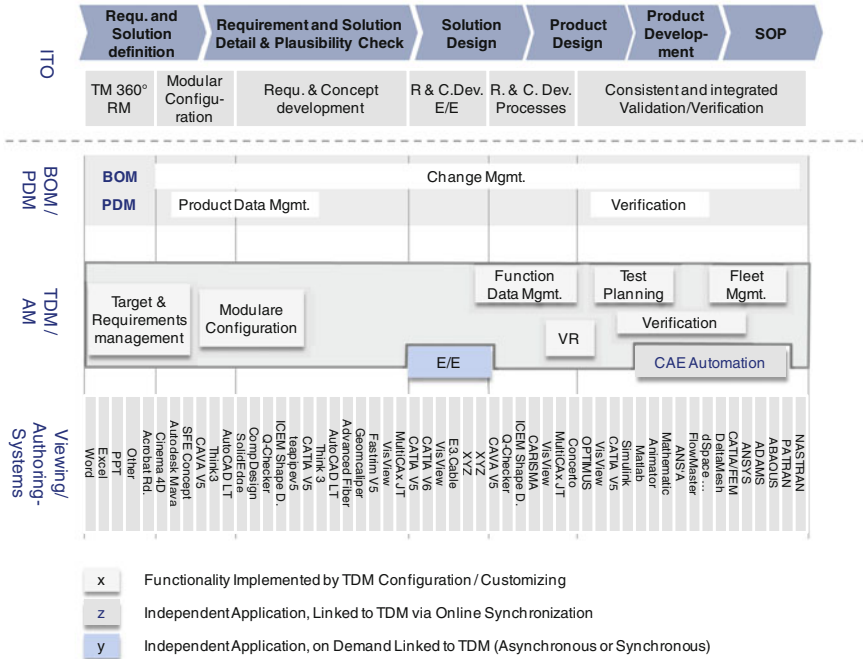


Fig. 19.16 PDP systems (example BMW)—vision

requirements management, modular product structure, function and test data management can be configured.

Because of the multiple requirements it does not make sense to implement all TDM functionalities with one single Backbone. Additional TDM applications might also exist. However, these have to be linked synchronously or asynchronously—appropriate to process requirements.

For the realization of such a consistent integration and the systems engineering approach in the company itself, the commitment of the IT system vendors with regards to openness and standardized interfaces is a necessary basic requirement.

19.6 Code of PLM Openness—Openness as a Requirement for the Implementation of Systems Engineering in a Company

These requirements can be met by using the CPO. The development of the CPO began in the end of 2011 in a close cooperation of different companies like BMW, Daimler, VW et al. (<http://www.prostep.org/de/cpo/unterzeichner.html>) plus a wide range of system vendors, managed and funded by the ProSTEP iViP Association,

Dilemma: Integration vs. Functionality

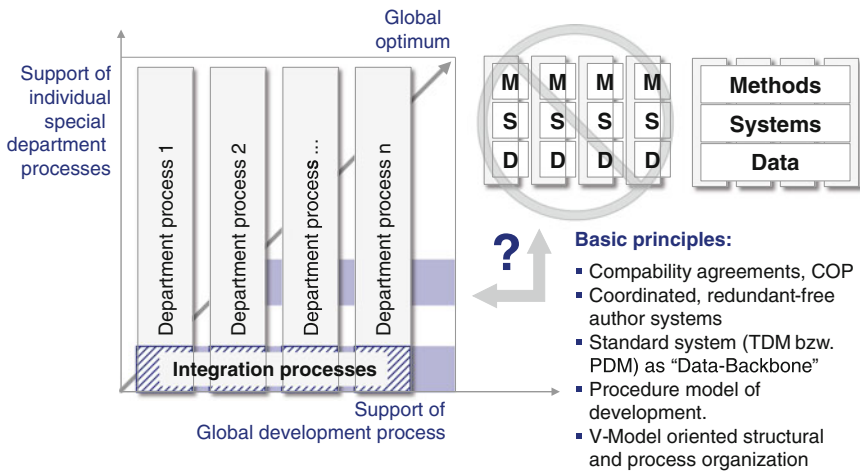


Fig. 19.17 Optimal engineering processes (functionality) versus optimal unified development (integration)

and was released in a first version in the beginning of 2012 (<http://www.prostep.org/de/cpo.html>). The code defines the term “openness” in the PLM context and defines a basic understanding in the following theme complexes:

- Interoperability and extensibility.
- System architecture and infrastructure.
- Interfaces and standards.
- Partnerships between IT system vendors and their clients

Meanwhile (as of 12/2014), over 80 companies signed the COP and, thereby, made a big step to support the necessary openness.

However, the code is only one element of the complete strategy. The code supports the creation of consistent integration in a complex and heterogeneous system landscape, but cannot be a replacement for an adequate process and system architecture strategy. The dilemma, to optimize on one hand discipline-related processes with highly specialized solutions and on the other hand to integrate all these applications to the perfect optimum, is not solvable with the COP alone. Today, the different methods, systems and data models don't follow a corporate schema. Thus, the incompatibility is quasi implied.

To what extent this applies can be affected with an adequate process-IT architecture strategy, so that in sum, still a good approximation to the targeted optimum can be reached (see Fig. 19.17).

Here, it's crucial to follow these basic rules:

- Harmonization of discipline-specific processes and integration processes with compatibility agreements concerning methods and procedures in the areas of interaction.
- Balanced selection of redundancy-free authoring systems, which are open as defined by CPO.
- Definition of a standard system (TDM or PDM) as Data-Backbone and Workflow-System. Here, the usage of CPO definitions is of significant importance to support also the integration of collaboration processes.
- Usage of a federal approach in the IT Architecture process under strict consideration of central defined rules for integration and compatibility.
- Establishment of a company organization structure that is oriented at the V-Model for both the engineering departments and the IT-department. Only by doing so, continuous solutions can arise and can be operated.