



D2.3: Industrial service-oriented process methodology

Author, company:

Stefan van der Elst, KE-works
Erwin Moerland, DLR
Johnny van Lugtenburg, Fokker Elmo
Luc Hootsmans, Fokker Aerostructures
Sebastian Deinert, Airbus Defence and Space
Martin Motzer, DRÄXLMAIER
Cristina Fernandez, IDEC

Version:

1.0

Date:

October 9, 2017

Status:

Final / Released

Confidentiality:

Public

CHANGE LOG

Vers.	Date	Author	Description
0.1	01.11.2016	Johnny van Lugtenburg	Created table of contents and bulletwise contents of each section.
0.2	22.11.2016	Johnny van Lugtenburg	Updated bulletwise contents of each section within the document based upon workshop results.
0.3	25.11.2016	Martin Motzer	Created content for section 3.2, organizational enablers, more specifically sections 3.2.4, 3.2.5 and 3.2.6.
0.4	25.11.2016	Sebastian Deinert	Created content for section 3.2, organizational enablers, more specifically sections 3.2.1, 3.2.2 and 3.2.3.
0.5	28.11.2016	Johnny van Lugtenburg	Created content for section 2.1, current state Product Development Process. Integrated contributions from other partners.
0.55	01.12.2016	Stefan van der Elst	Input for section 3.1, technical enablers, and suggestions for integration with chapter 2.
0.6	09.12.2016	Johnny van Lugtenburg	Created section on Industrial challenges and updated section 2.1 Current state process.
0.7	05.01.2017	Erwin Moerland	Initial review of complete document Resorted the introduction Created content for section 3.1, in particular 3.1.1 on general architecture as blueprint for implementing the future PDP framework
0.8	23.01.2017	Johnny van Lugtenburg	Finalized the introduction according suggestions. Updated section 2.1 (current PDP) and created input for section 2.2 (future PDP), including figures.
0.85	01.02.2017	Johnny van Lugtenburg	Updated and extended section 3.2, organizational enablers. All enablers described.
0.9	09.02.2017	Luc Hootsmans	Remarks and contributions to chapter 2 and 3.
0.91	09.02.2017	Stefan van der Elst	Re-writing and extension of section 3.1 (technical enablers) and review of the document up to and including section 3.1.
0.92	14.07.2017	Johnny van Lugtenburg	Incorporating the remarks on chapter 2.
0.93	28.07.2017	Luc Hootsmans	Contribution to description of industrial example from Fokker Aerostructures in section 4.1.2.
0.94	09.08.2017	Erwin Moerland	Review and restructuring of complete deliverable, added contributions to section 2.2 and extended sections 3.1 and 3.2.
0.95	10.08.2017	Martin Motzer	Contribution to description of industrial examples from DRÄXLMAIER in section 4.1.4.
0.96	11.08.2017	Sebastian Deinert	Contribution to description of industrial examples from Airbus Defense and Space in section 4.1.1.
0.97	29.08.2017	Johnny van Lugtenburg	Final update on chapter 2, addition chapter 4.2 and integration of contributions in the remainder of the document. Released for formal review.
0.98	31.08.2017	Erwin Moerland	Performed extensive final review of complete document
0.99	04.09.2017	Cristina Fernandez	Performed extensive final review of complete document.
0.995	15.09.2017	Martin Motzer	Change in section 4.1.4 due to DRÄXLMAIER review.
1.0	09.10.2017	Johnny van Lugtenburg	Integration of final comments and document released.

Table of Contents

1	Introduction	5
1.1	Major goals of the future product development process	5
1.2	Industrial challenges covered within IDEaliSM	6
1.3	Focus and structure of this report	8
2	Current- and future PDP	9
2.1	Current state product development process	9
2.2	Future state product development processes	13
2.3	Future process opportunities with respect to current state bottlenecks	18
3	Enablers for future PDP implementation	20
3.1	Technical enablers	20
3.2	Organizational enablers	24
4	IDEaliSM example implementations and business impact	30
4.1	IDEaliSM example implementations	30
4.2	Business impact	33
5	References	37

List of abbreviations

AIF	Advanced Integration Framework
COTS	Commercial Off-The-Shelf
DoE	Design of Experiments
EL	Engineering Library
ELW	Engineering Language Workbench
IT	Information Technology
KBE	Knowledge Based Engineering
KM	Knowledge Management
MDO	Multidisciplinary Design Optimization
MT	Management Team
NRC	Non-Recurring Cost
OEM	Original Equipment Manufacturer
OML	Outer Mold Line
PDP	Product Development Process
PIDO	Process Integration and Design Optimization
R&T	Research and Technology
RC	Recurring Cost
RFP	Request For Proposal
ROI	Return On Investment
V&V	Validation and Verification

1 Introduction

This document provides a description of an industrial service-oriented process methodology capable of addressing the challenges within the development of automotive- and aerospace structures and systems observed nowadays. The main goal is to describe the methodology and enablers to facilitate highly automated and distributed development processes, as proposed in IDEaliSM. Hence, this document includes a definition of an innovative and generically applicable future Product Development Process (PDP), as well as a description of what an industrial company needs to put in practice when implementing these kinds of processes.

1.1 Major goals of the future product development process

IDEaliSM aims at drastically reducing the time-to-market and development cost of high-tech structures and systems, by delivering a novel product development framework being:

- *distributed, flexible and service-oriented*: engineering competences distributed across multiple competence centres are provided in the form of engineering services¹ being flexibly applicable to a multitude of products;
- *capable of utilizing multidisciplinary design and optimisation techniques*: the integrating capabilities of the framework utilize the principles of multidisciplinary design and optimisation to cope with the vast design spaces of the complex products considered;
- *capable of integrating people, process and technology*: allowing for the next generation of collaborative engineering, processes are formalized in which engineering routines as well as their owners (e.g.: domain experts) are seamlessly integrated.

The framework allows companies to apply and reuse their core knowledge effectively by offering their competences in the form of engineering services to a multitude of customers and programs and rapidly (re)configure processes and tools for new development projects. As depicted in Figure 1, this future state framework enables the continuous integration of information and changes throughout the complete value chain during the execution of product design projects.

¹ A definition of engineering services is provided in 2.2.1

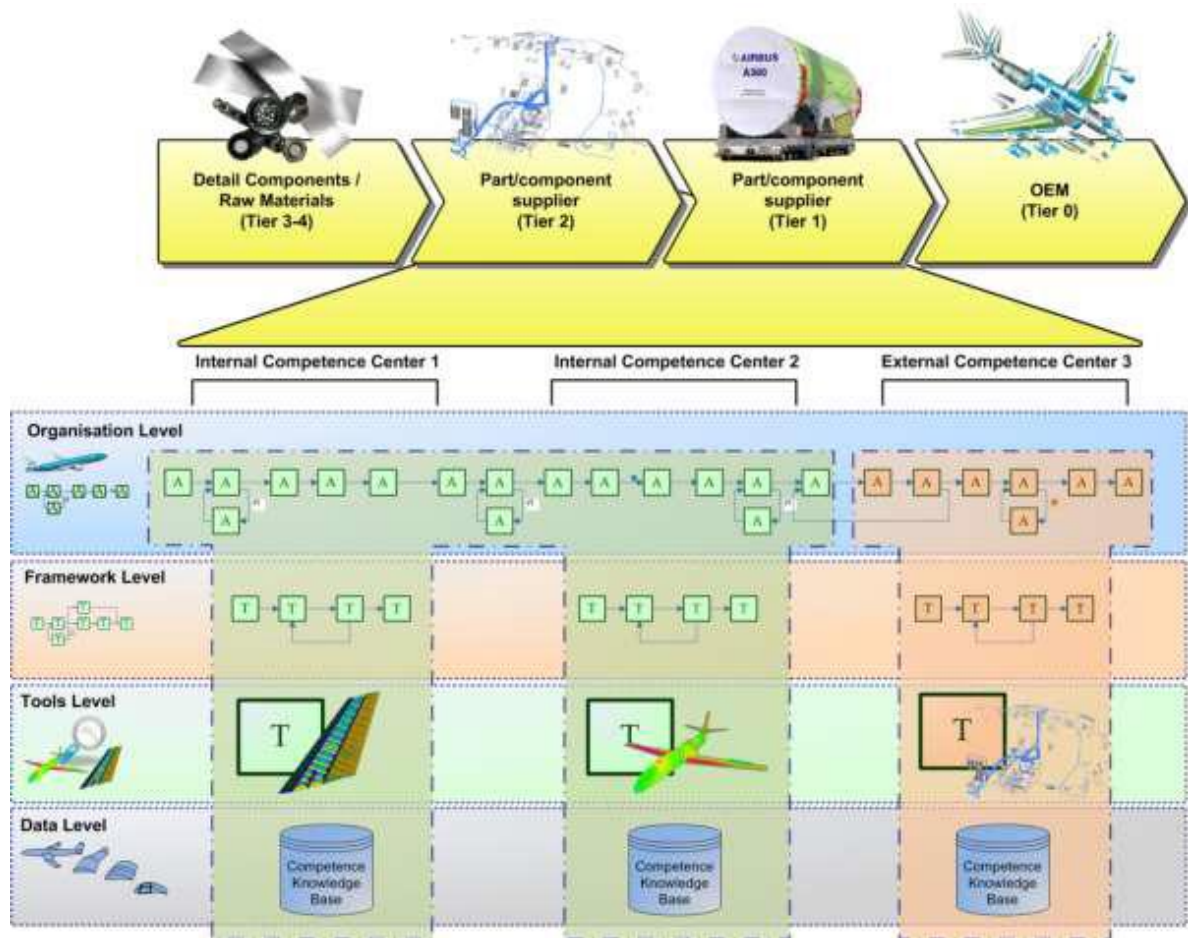


Figure 1: Integration of Competence Centres within and between company domains

1.2 Industrial challenges covered within IDEaliSM

The development of automotive- and aerospace structures and systems face a number of challenges that have not been sufficiently addressed and hamper further growth of the pan-European high-tech engineering industry. The following challenges are addressed by the IDEaliSM project and are laid out in the ITEA roadmap for Engineering-Process Support, part of the Engineering Technologies domain [1]:

21. *Integration and interoperation of engineering and tools*: The European high tech industry like automotive and aerospace faces the challenge of continuous integration of dedicated software tooling related to specialized engineering domains. This encompasses software development as well as the automated inclusion of Commercial Off-The-Shelf (COTS) software. To assure effective engineering process support, technical tool integration is required, enabling a seamless integrated engineering process;
22. *Distributed and collaborative engineering*: Besides the technical linking of software tools, the “linking of engineers”, i.e. the collaboration and integration of aspects of human interaction with the development process is of increasing importance. The aerospace and automotive design and manufacturing industry has rapidly “globalized” and there is a need for more flexible collaboration and integration between highly specialized teams spanning multiple sites and multiple companies;

- 23. Configurable methodologies and process standards: Most processes in the automotive and aerospace design- and manufacturing industry are a combination of several components from a large set of possible standard process components. Being able to rapidly configure and re-configure processes based on a library of standard process components allows for large flexibility in tailoring to the individual needs and requirements of a customer or partner within distributed design teams. This opens up the opportunity to support an increasing amount of specifically tailored design process at much lower cost and effort;
- 25. Knowledge Based Engineering (KBE): The complexity of the engineering processes within the automotive and aerospace design- and manufacturing industry has increased over the past decade, as the products themselves have become more complex. The engineering of such complex multi-disciplinary products is highly knowledge-intensive. When this knowledge remains an implicit asset of the domain experts (i.e. is not made explicit), it is difficult to access for other engineers within the integrated design process. KBE tools allow for capturing this essential engineering knowledge and re-using it in the form of decision automation tools and/or as enablers for Multidisciplinary Design Optimization (MDO);
- 26. Human Resources: The fact that on average 20% of the staff is lost per year, more senior engineers are retiring in the coming decade and fewer freshly educated engineers are expected to fill the gaps; leads to complications in the availability of sufficient experience and expertise. Ways need to be found to capture knowledge and stimulate knowledge re-use, which can include a single, central source of product- and process information and consistent tools and methods across engineering.

The aforementioned challenges are represented in Figure 2, which serves as structure for the remainder of this document. In this figure, the challenges are translated into more specific process bottlenecks as experienced today. Each bottleneck is connected to an IDEaliSM innovation, specifying the solution as brought forward by the project. In the third column, the expected business impact of this IDEaliSM innovation is depicted.

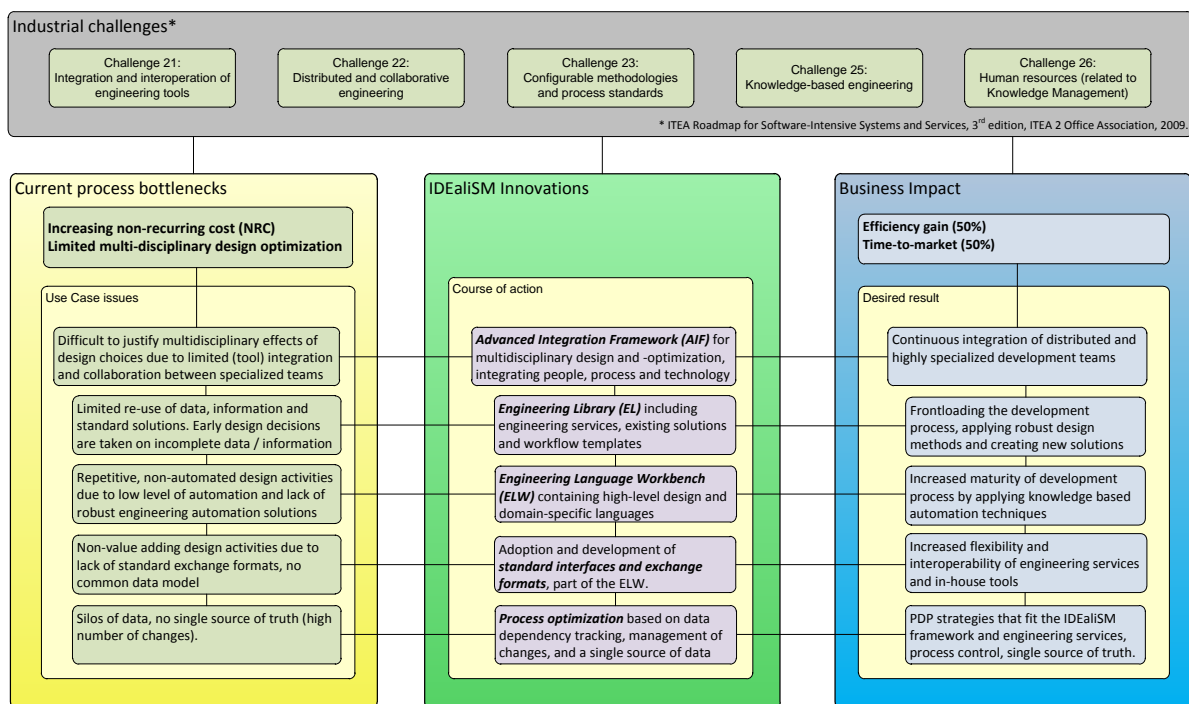


Figure 2: Process bottlenecks, IDEaliSM innovations and their expected business impact

1.3 Focus and structure of this report

The primary focus of this deliverable is on the framework- and organization level depicted in Figure 1. Thereby, it intends to provide a general overview of the idea and implications of the enhanced product development process as developed within IDEaliSM. Details on the required technical enablers as developed in IDEaliSM on the tools- and data level of the framework (Figure 1) are included in separate project deliverables. The interested reader is kindly referred to the documents listed in Table 1 for obtaining a more detailed insight in the technical implementation of the product development process. Each deliverable consists of a set of software titles contributing to the prototype demonstrators, as well as documentation describing the contents of the respective component.

Table 1: References to IDEaliSM technical documentation deliverables

ID	Title	Short description
D3.2	Advanced Integration Framework	The Advanced Integration Framework is the main deliverable integrating the workflow components, the engineering services and the distributed knowledge base.
D4.2	Engineering Library	The Engineering Library contains existing and known solutions and tools to enable rapid frontloading of engineering projects. It is a repository containing all available and formalized engineering knowledge.
D4.3	Engineering Language Workbench	The Engineering Language Workbench contains all the building blocks required to create engineering services. Typical components are domain-specific languages, programming languages and packages, data exchange formats, and development support systems.

The structure of the report is in accordance with the structure as presented in Figure 2, with each of the columns described in a separate chapter.

Chapter 2 discusses the main bottlenecks related to the current product development process (PDP). Thereafter, sequential implementation steps for implementing the envisioned future PDP are described, resembling a transition from sequential- and concurrent processes into front-loaded design processes. The front-loaded process is based on re-using corporate knowledge by effectively applying engineering services and facilitating a continuous integration of changes.

Chapter 3 provides the enablers for the implementation of the future front-loaded PDP, including an overview of the required high-level functional and/or technical aspects and especially oriented towards organizational aspects. Examples of organizational aspects include the impact on the roles and capabilities of engineering and IT teams and the need for organizational commitment and continuity as well as new financial models to implement and exploit such future PDP.

Chapter 4 includes example implementations of the future PDP by the industrial partners involved in IDEaliSM as well as a description of the business impact of the innovations as brought forward by IDEaliSM.

2 Current- and future PDP

This chapter provides a generalized description of the currently applied Product Development Process (PDP) and its bottlenecks in section 2.1. Subsequently, section 2.2 describes sequential implementation steps to achieve the novel PDP as developed within IDEaliSM, of which its opportunities are discussed in section 2.3 with respect to today’s industrial context as well.

2.1 Current state product development process

The nature of engineering processes within modern organizations has altered dramatically over the past decade. Today’s current practice for the development of high-tech solutions is oftentimes based on a concurrent product development process (Figure 3). Driven by increasing pressure on reducing time-to-market, different development phases run concurrently. This results in a shorter development lead time compared to those of traditional stage-gated and sequential processes [2]. However by having the different phases run concurrently, also inefficiencies are introduced. For example, assumptions need to be made because certain requirements from ancestor phases are not yet clear when a design is supposed to be made. Oftentimes those assumptions prove to be wrong, requiring non-wanted re-design activities. Likewise, concurrently working on different versions of data sets may introduce further changes and thus cost to the design.

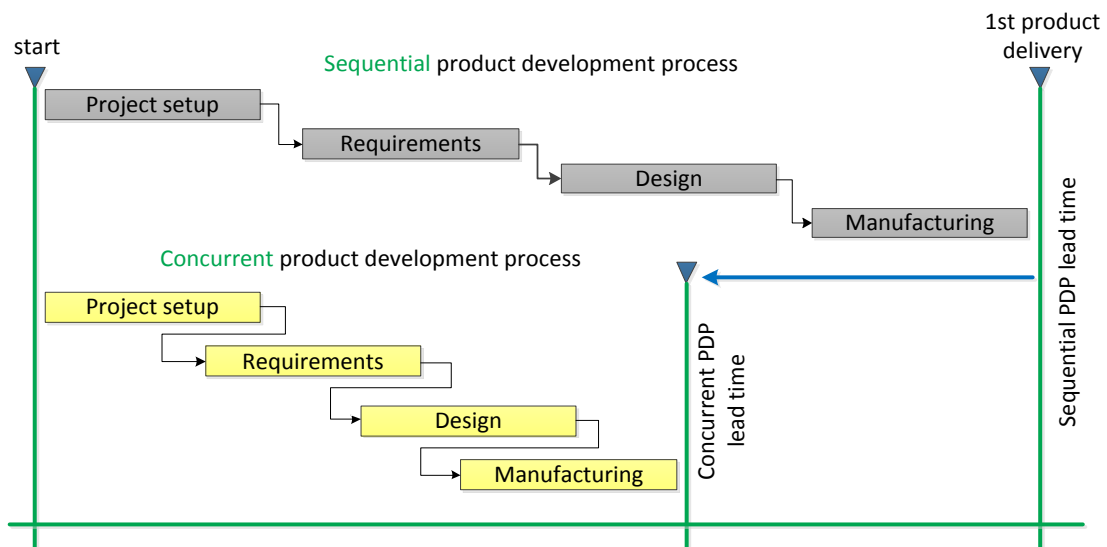


Figure 3: Schematic on sequential- and concurrent Product Development Processes (derived from [2])

2.1.1 Bottlenecks of the current state product development process

Within this section, main process bottlenecks related to the current PDP are made explicit and discussed into more detail.

Figure 4 and Table 2 include a supply chain product development process in which primary process bottlenecks for common engineering processes are listed. Note that the bottlenecks are highly interrelated and closely related with the industrial challenges. A more detailed description of these bottlenecks is substantiated in text below the table.

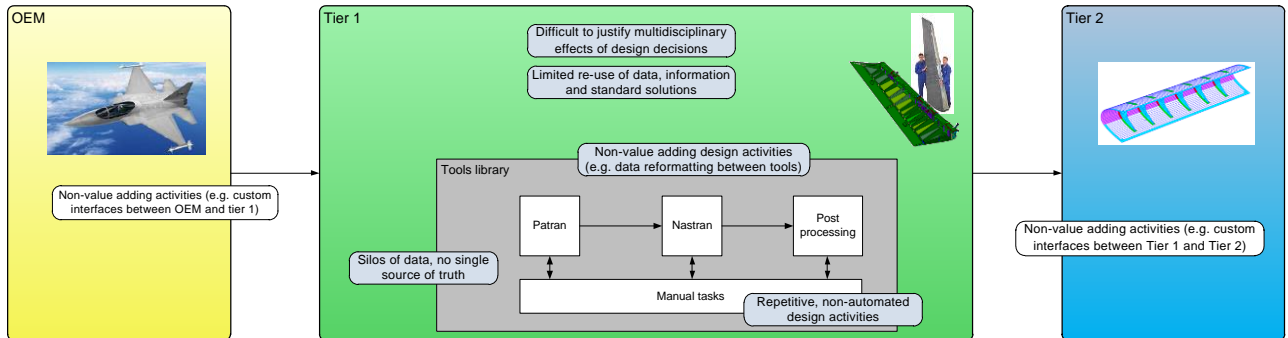


Figure 4: Example product development process and bottlenecks

Table 2: Main current state process bottlenecks

Current state process bottlenecks			Related to industrial challenge #
ID	Short description	Explanation	
1	Difficult to justify multidisciplinary effects of design decisions	Design decisions are primarily mono-disciplinary driven due to today's highly specialized teams, limited tool- and people integration as well as time-constraints and high number of changes within today's concurrent process.	21, 22
2	Limited re-use of data, information and standard solutions	Early design decisions are oftentimes based on incomplete information, requiring changes later in the process. This both impedes product optimization and results in costly incorporation of changes late in process.	23
3	Repetitive, non-automated design activities	Many repetitive and non-automated design activities are present within current engineering processes. This occurs during regular process tasks as well as when changes are to be incorporated. When these activities are not automated, this results in high costs of the process and can trigger errors as well.	25, 26
4	Non-value adding design activities	Many non-value adding design activities are present. Limited tool interoperability by lack of standard interfaces, data formats and common data models lead to frequent, non-value adding and manual data reformatting activities within the process.	21
5	Silos of data, no single source of truth	Often there is a lack of control over information and design status, since no single source of truth is available. Engineers concurrently working on different versions / sets of data inherently leads to a high amount of changes.	23

ID #1: Difficult to justify multidisciplinary effects of design decisions

In the current process, it is very difficult to justify the effect of design decisions whereas this crosses specialized teams, each involving many and detailed analysis efforts and dedicated software tools. This effect is present within a company between today's highly specialized teams but is even more profound in case of multiple companies working together, e.g. spanning OEM, Tier 1 and Tier 2 industrial companies. Further limited by time constraints within the concurrent process, in combination with constant changes to which the design subjected to, it is difficult to assess and make optimal design decisions. This bottleneck is strongly related to challenges 21 and 22 as described in section 1.2, referring to limited tool- and people integration respectively.

ID #2: Limited re-use of data, information and standard solutions

The current process typically includes growing data maturity and completeness over time. Combined with a limited re-use of data, information and standard solutions from legacy programs, this implies that the impact of design decisions early in the process cannot be fully evaluated, largely reducing the optimality of products. This effect is depicted in Figure 5: early in the process, decisions are made based on low maturity of available data. These decisions however have the highest effect on the resulting final products' Recurring Cost (RC) or other performance parameters (e.g. weight). This same bottleneck leads to the need for making assumptions, oftentimes triggering many changes in later process stages when the data is actually getting more complete and mature, substantially lowering the efficiency and increasing the cost of the process.

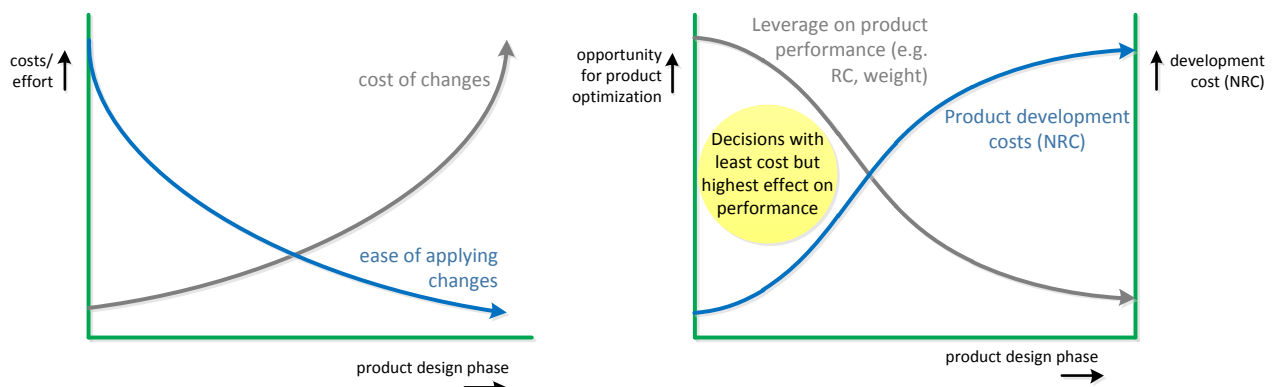


Figure 5: Process characteristic: early design decisions have least cost and highest effect on performance

ID #3: Repetitive, non-automated design activities

Today's design process is highly knowledge intensive and many design activities are non-automated, resulting in high non-recurring costs of the engineering process and inhibiting knowledge re-use (challenge 25). Especially in case of manual- and repetitive tasks (incl. changes as triggered by the other bottlenecks), this prevents the engineer to exploit his knowledge on the creative part of design and those activities that generate most added value. Furthermore, whereas much knowledge actually resides in the minds of experts, this triggers knowledge management challenges. This drives the need to capture engineering knowledge into engineering services that can automatically perform repetitive design tasks; also providing part of the solution for challenge 26 on difficulties in Human Resources (section 1.2).

ID #4: Non-value adding design activities

Many non-value adding activities are present into today's process, which are oftentimes also non-automated and thus related to unnecessary cost and effort.

Frequent data reformatting activities related to limited tool integration by the lack of standard interfaces and data formats and/or common data models (challenge 21) is an often observed concrete example of this problem. When these kind of activities can be eliminated or when design automation can be applied, the efficiency of the process can be largely increased.

ID #5: Silos of data, no single source of truth

Within today's industry context and concurrent engineering process, specialized teams are working globally and concurrently. To manage such a process efficiently, a single source of truth for all product- and change data, as well as for all models and tools both within a company and between companies within the supply chain is required. Often this is not the case: silos of data are observed which are not integrated across the supply chain, there is no single source with live up-to-date data and no homogeneously applied data format. Assuring data consistency can be problematic when no clear procedure for identifying master/slave roles within the development process exists, allowing data to be changed by multiple services at the same time. This leads to a lack of control over information, design status and product performance. Furthermore, traceability is limited, resulting in cumbersome Validation and Verification (V&V) of results and requirements as well as limited control and overview of project progress. This at its turn inflicts a high number of changes. For example in case engineers are working on different versions or sets of data, a cascade of changes over multidisciplinary teams and even throughout the complete supply chain (OEM, Tier 1 and Tier 2 companies) can be regularly triggered.

2.1.2 Performance of the current state product development process

The high level of specialization, widespread- and relatively low-cost availability of computing resources (enabling detailed analysis and high-level simulation opportunities) as well as principles of concurrent engineering allow for the development of very complex products with a high level of safety and within acceptable lead-times and product performance.

However, as Table 3 summarizes, due to the main bottlenecks as described in the previous section, the current PDP also results in:

- High Non-Recurring Cost (NRC) due to many repetitive- and non-value adding activities being manually executed, including the burden of costly changes primarily resulting from data incompleteness and immaturity as well as from the inherent inefficiencies of the concurrent engineering approach:
 - o While high non-recurring costs being problematic in itself, it can also lead to budget- and lead-time overruns of a commercial program due to non-anticipated major re-designs, which can incur large penalties to an industrial party within a supply chain.
- Limited product optimization due to limited application of multidisciplinary design optimization principles. Alternative design solutions and their effects cannot be evaluated early in the design process due to lacking data and design immaturity, while within the detailed design process time constraints (due to limited tool- and people integration) and continuously changing data are the primary impediments. This leads to products that are more expensive, less efficient or have a larger environmental footprint than necessary.

Table 3: Relating process bottlenecks with performance

Current state process bottlenecks		Primary impact on performance driver
ID	Short description	
1	Difficult to justify multidisciplinary effects of design decisions	Limited product optimization
2	Limited re-use of data, information and standard solutions	
3	Repetitive, non-automated design activities	High Non-Recurring Cost (NRC) (<i>& potential budget- and cost overruns of programs</i>)
4	Non-value adding design activities	
5	Silos of data, no single source of truth	

The bottlenecks in Table 3 are highly interrelated, making the total impact of the set of bottlenecks even larger than the sum of its individual impacts. Some examples:

- In the early stages of the process, the ability to accurately determine mono- and multidisciplinary effects of design decisions (bottleneck 1) is further impeded by data incompleteness and immaturity (refer to bottleneck 2). And calculating multidisciplinary effects is hampered by limited integration and/or interoperation of tools within detailed design phases (refer to bottleneck 4);
- Time constraints due to non-automated tasks (bottleneck 3), non-value adding design activities (bottleneck 4), silos of data (bottleneck 5) and continuous design changes further inhibits the capability to optimize the design, perform MDO and/or sensitivity studies (bottleneck 1) within the available development time;
- Incorporating changes is oftentimes a repetitive, non-automated design activity (bottleneck 3), in turn due to limited integration between specialized teams and -tools (bottleneck 1 and 4), data incompleteness (bottleneck 2) and the absence of single source of truth (bottleneck 5).

The interrelatedness of the bottlenecks argue for a solution that takes into account all these aspects simultaneously in order to provide the required business impact.

2.2 Future state product development processes

The technologies delivered in IDEaliSM enable a different way of working, opening the way towards radically new and improved PDPs. This section provides a description of the generic future state PDP and discusses its impact with respect to the process bottlenecks identified in section 2.1. The future process is applicable to different organizational entities within the supply chain (OEM / Tier1 / Tier 2) and to different industry domains (e.g. automotive/aerospace).

The future state PDP can at best be implemented sequentially in three major implementation steps:

1. Engineering services to automate repetitive manual design tasks;
2. Integration of a multitude of engineering services in business- and simulation workflows;
3. Transition to a front-loaded product development process.

These three implementations are ordered based on increasing solution coverage with respect to the process bottlenecks. Furthermore, the implementation steps are chosen such that the various phases incrementally improve the design process and such that the implementation of each of the individual steps already provides a significant benefit for the company.

This *staged implementation strategy* generally helps to gain confidence in the updated process and maximises the chance to proceed to subsequent phases. Obviously, to gain maximum benefits and profit from all efforts, the complete implementation of all steps is required.

The following sections discuss the three sequential implementation steps of the future state PDP.

2.2.1 Engineering services to automate repetitive, manual design tasks

The strategy in the first of three implementation steps entails the development of engineering services to automate parts of the design process that are currently of a repetitive, manual and non-value adding nature. In this scenario, engineering services form the basis of the future-state product development process and are defined as follows:

An *engineering service* is defined as:
a generically applicable software routine within the engineering domain, capable of automated handling input and output data in a standardized data format, which can be approached by other services via standard web or network technologies and ideally allows for batch execution without requiring any intervention of the user.

The foundation of creating engineering services lies in the structured capturing, formalization and automatic execution of company-specific engineering knowledge by using software technology. This principle is referred to as Knowledge Based Engineering (KBE): enabling the re-usage of knowledge accumulated over the years by automatically executing (mono-disciplinary) design tasks.

By adding interfaces to standard data formats for the exchange of process- and product information (i.e.: the capability to read input from- and write output to these data formats) and making sure the KBE application can run without user intervention, engineering services are created. The automated exchange of data through interfaces to standard data formats has multiple benefits. First, it further standardizes the execution of a KBE routine and allows saving input-output data according to standardized schemes interpretable by multiple engineers. Second, it enables the connection to other engineering services allowing for the standardization of complete parts of the PDP.

The principle of engineering services can either be created from scratch to increase the engineering capabilities of a company, or - as observed more commonly - can be used to automate legacy processes of a company. Engineering services can be developed for a diverse range of engineering processes, including conceptual- and design tasks, and applied to a broad range of domains, e.g. aerospace / automotive / shipbuilding, etc. It allows industrial companies to valorise on their core knowledge and speed up parts of the development process, as a source of competitive advantage.

When creating a set of engineering services which base on similar concepts (e.g.: similar geometric operations are required throughout the set of engineering services), creating a dedicated development environment is recommended. By setting-up such a development environment including libraries of high-level design languages and a set of standardized ontologies and domain-specific design languages, complete (sub-) processes of tasks can be automated efficiently and according to similar standards.

It is important to recognize that not everything can, or indeed should, be automated. There must be a careful trade-off between the development effort required for automation and the resulting business benefit. A typical implementation of engineering services within this implementation scenario is to provide economies of scale on repetitious tasks or tasks with a high degree of similarity. It can also be applied to complex engineering processes requiring simplification/standardization.

Within this implementation scenario, the major aim is to complete design tasks in reduced time, thereby achieving cost reduction. This scenario already yields considerable business value and will probably stimulate further development by gained confidence. The creation of multiple engineering services acts as an enabler for subsequent implementation steps, more focused on product optimization.

2.2.2 Integration of multiple engineering services in business- and simulation workflows

A typical design process encompasses multiple design disciplines (e.g. structures, aerodynamics), design phases (e.g. conceptual, preliminary, detailed), and levels of detail (e.g. macro, micro). A promising field of engineering that addresses the integration of multiple disciplines at multiple levels of detail and for multiple design phases is the field of Multidisciplinary Design and Optimization (MDO).

This second implementation step covers the development and integration of multiple engineering services within business- and simulation workflows. This largely widens the scope of automation applied to the design process and characteristically allows for the adoption of MDO techniques.

Within **business and simulation workflows**, multiple manual- and automated engineering competences are integrated into a single process, allowing for project performance monitoring, embedding requirements and change management, and optimization principles to be included.

The high-level architecture of the framework enabling the integration of the created engineering services comprises three layers: a Tools layer, a Simulation layer and a Business workflow layer. Figure 6 depicts a general example of an implementation at a Tier-1 supplier within the value chain. After development and testing of the required engineering services, these are made available through standard web or network technologies within the Tool layer. Oftentimes covering multi-disciplinary domains, these available services are connected in simulation workflows within the simulation layer. In this layer, multiple competences are integrated in a single MDO system, allowing for optimization principles to be included. Likely, this will involve integration of the specifically developed KBE applications as well as of COTS CAD/CAE tools in use within the already available design process. To complete the workflow-based process, workflows enabling the combination of automated design capabilities with remaining manual tasks are defined within the Business Workflow layer. Within this highest level of the architecture layer; also project performance monitoring, change monitoring, requirements management and configuration management are included.

Integrating engineering services and performing MDO enables the (semi) automated exploration of design solution spaces. Such design spaces are typically large and multidimensional and can oftentimes hardly be grasped manually by engineers. Pushing MDO and numerical post processing techniques into the covered design phase will support the decision-making process considerably through computerized generation of reliable physical information within the bounds of the explored design spaces.

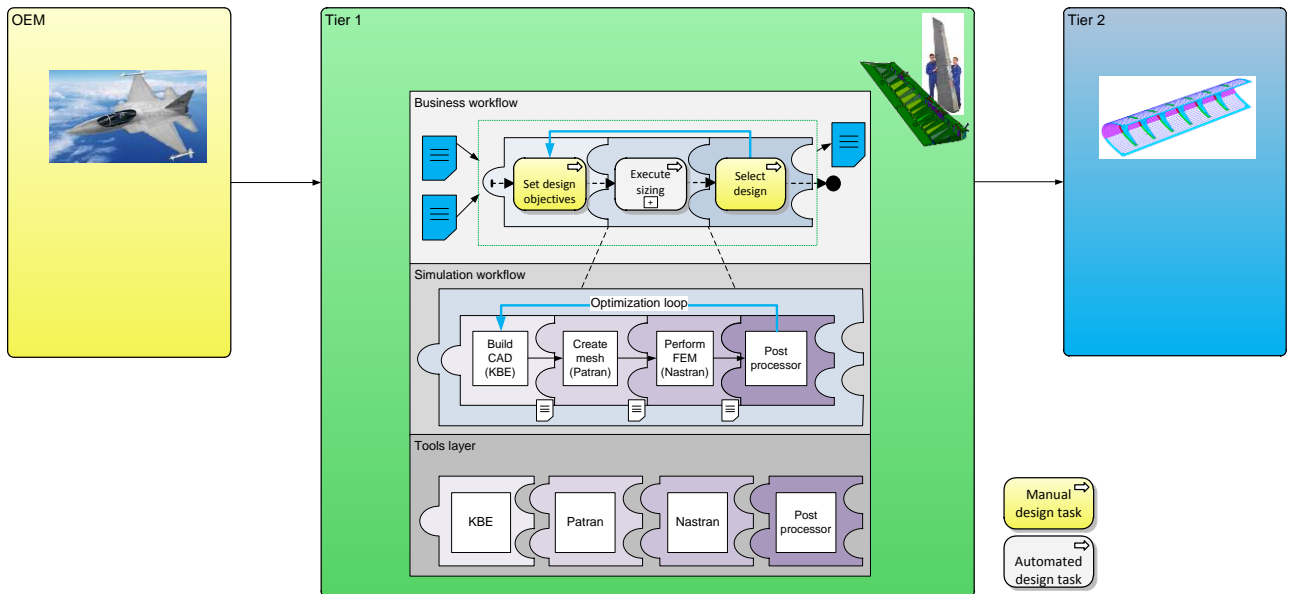


Figure 6: Multiple engineering services are connected through business and simulation workflows

Key characteristic of this implementation scenario is the ability to include product optimization and to substantiate the effect of design decisions on overall product design level, including its major properties (e.g. recurring costs, weight). When applied to the conceptual design phases of the product development process, this allows for more substantiated design decisions at moments when the effect of design decisions and design freedom is still relatively high and the cost of proposing changes relatively low (as was already depicted in Figure 5). Through this capability, the reduction of non-recurring costs due to the broader scope of design automation in early design phases represents a well-wanted secondary effect.

2.2.3 Transition to a front-loaded product development process

Within this last implementation scenario, building blocks of the previous scenario are used in conjunction with new enablers to radically change the product development process into a so-called front-loaded design process. Front loading is described by Thomke and Fujimoto [3] as “a strategy that seeks to increase development performance by shifting the identification and solving of design problems to earlier phases of a product development process”.

However, this strategy can be shifted even further forward, by developing engineering knowledge before the actual design process starts. Within IDEaliSM, therefore the following definition of front-loading is therefore introduced:

A *front-loaded product development process* is defined as:
 a strategy in which increased performance and reduced time-to-market is sought by shifting the identification and resolution of design problems to earlier phases, or even in front of the actual product development process.

The principle of front-loading is depicted in Figure 7. The difference of this front-loaded design process is depicted with respect to traditional sequential- and concurrent product development processes [2].

In the front-loaded scenario; product- and engineering knowledge from earlier projects is captured, re-used and standardized to enable rapid evaluation of many design variants whilst covering different requirements sets. During the subsequent design phases, the goal is to achieve full maturity of each of the promising design concepts evaluated and enabling continuous integration of changes. Multiple options can be kept open during the design definition phase, providing the flexibility to switch to alternative concepts when design requirements from the customer change or become more mature. This ensures a large reduction in time-consuming changes when a chosen design concept needs to be adjusted.

Since design concepts are completely analysed during the front-loading part of the development process, a high completeness and maturity of data is achieved early in the design process. This ensures that development setbacks can be identified when changes in the product are still allowed, and allows for a better response to changing requirements.

As second benefit, a better integration and collaboration through the supply chain, e.g. between OEM, Tier 1 and Tier 2 suppliers, can be achieved through quicker and better substantiation of the effect of changes.

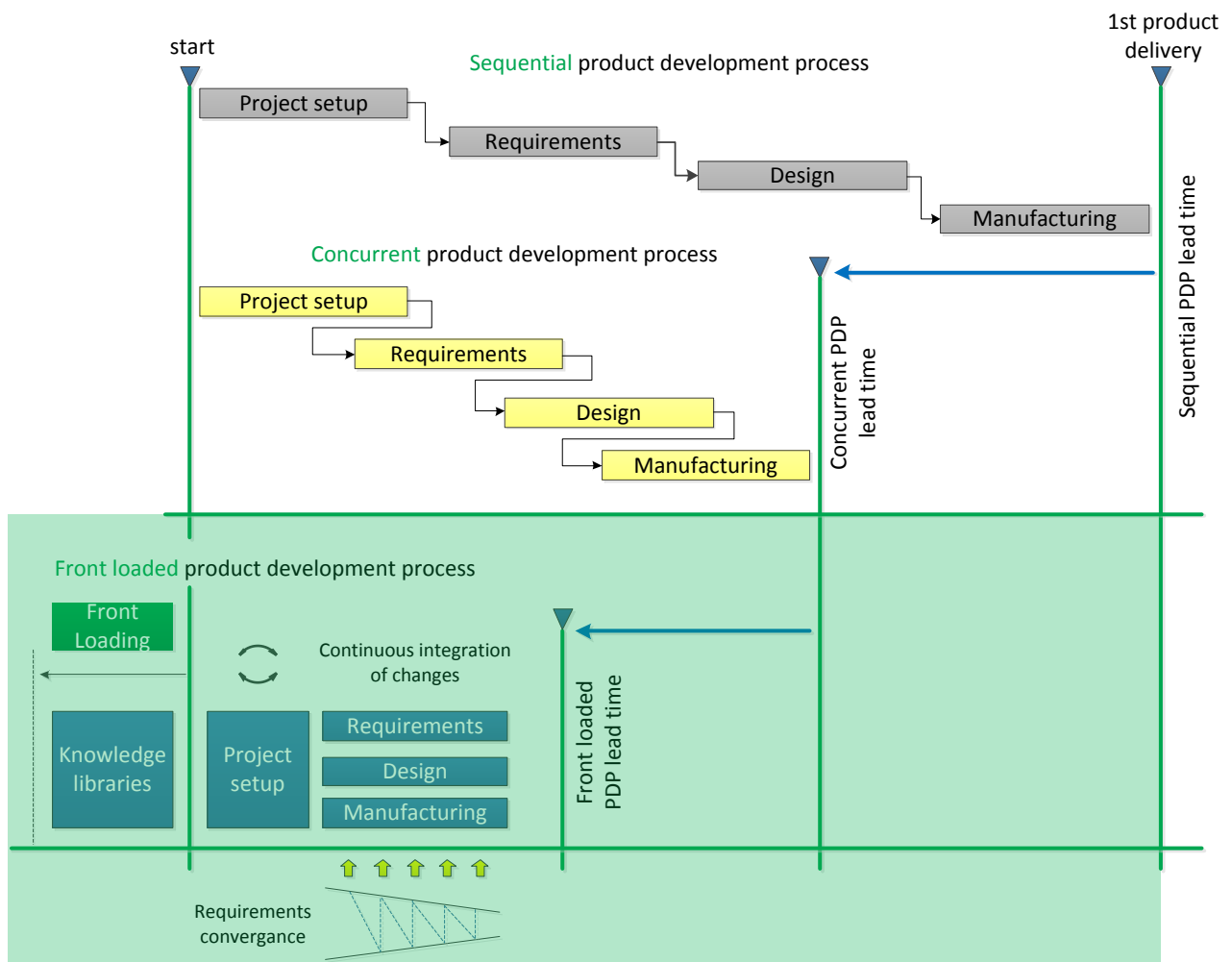


Figure 7: Innovative, front-loaded product development process

Once the front-loaded product development process is in place, the ability to challenge customer requirements by performing large amounts of “what-if” studies in an automated way occurs. Examples of such studies could be: what if the requirement on maximum runway length as provided by the customer is relaxed slightly, how would this affect overall aircraft performance? Or: what if the pre-provided cockpit geometry is adjusted slightly at “routing hotspots”, would this improve harness quality?

Especially related to Tier 1 and Tier 2 industrial companies, another major advantage is that the front-loading scenario allows to quickly respond to Request For Proposals (RFPs), which often need to be prepared in limited time, yet determining the project business case for many years to come. Within the frontloading scenario, preliminary design studies can already be performed before an actual requests for proposal is received, allowing better estimates of non-recurring-, as well as recurring costs and weight estimates of the created product. And as depicted by the “what-if” studies above, front-loading delivers the capability to challenge requirements within the RFP phase to provide customers possibly even better solutions to their problem.

2.3 Future process opportunities with respect to current state bottlenecks

This section describes the opportunities of the three future process implementations stages. Table 4 provides an overview of the current state bottlenecks on the left side and the opportunities of the different future process implementations on the right hand side.

Table 4: Opportunities of the three staged, future Service-Oriented Process in resolving current bottlenecks

Current state process bottlenecks		Service-Oriented Process opportunities		
ID	Short description	Stage 1	Stage 2	Stage 3
1	Difficult to justify multidisciplinary effects of design decisions	-	Multidisciplinary effects (e.g. cost/weight) of design decisions are substantiated, as enabled by the integration of people, processes and technology. <i>IDEaliSM Innovation: Advanced Integration Framework</i>	
2	Limited re-use of data, information and standard solutions (incurring changes)	-	-	Front-loaded design process using a knowledge library including the storage and re-use of validated standard solutions, enabling efficient product optimization and less changes ² . <i>IDEaliSM Innovation: Engineering Library</i>

² The number of changes can be reduced by:

- Tools to predict and complete missing data early in the design process;
- Identifying design concepts that would otherwise be missed within reduced design spaces of the current PDP;
- Sensitivity- and robustness analysis: choosing a design solution that is less sensitive to changes.

3	Repetitive, non-automated design activities	The cost of executing repetitive design tasks are reduced by the utilization of automated engineering services (created using KBE principles) and through the implementation of high-level design and domain-specific languages. <i>IDEaliSM Innovation: Engineering Language Workbench</i>	
		Design process tasks and changes can be quickly incorporated through automation on a small, mono-disciplinary scope.	Design process tasks can be quickly executed and changes can be quickly incorporated by interconnecting the automated components on a wider, multi-disciplinary design scope.
4	Non-value adding design activities	Interoperability of tools is enabled by standard interfaces and exchange formats through application of common data models. No time is lost through manual preparation and conversion of required data. <i>IDEaliSM Innovation: Engineering Language Workbench</i>	
5	Silos of data, no single source of truth	-	Process optimization based on data dependency tracking, management of changes, and a single source of data. <i>IDEaliSM Innovation: Advanced Integration Framework including process optimization</i>

Based on the overview provided, it can be concluded that each staged process implementation step provides an extended coverage to solve identified current process bottlenecks.

Obviously, a more radical future process also requires more resources for adequate implementation of both a technical- as well as organizational nature. The following chapter both describes the identified technical- as well as organizational enablers for advancing the product development process to the advanced front-loaded state as described above.

Actual industrial implementations are provided in chapter 4, including identification of the impact of the future process implementations.

3 Enablers for future PDP implementation

Both functional- and organizational capabilities need to support the future PDP scenario. Although functional (or technical) enablers are provided in the first section of this chapter, this document focuses more on organizational aspects in implementing a front-loaded PDP. Further details on technical content can be obtained from separate and more in-depth deliverables as indicated in the introduction of this document.

3.1 Technical enablers

Before detailing out the technical enablers allowing the integration framework for the future PDP to be set-up, the general architecture developed within the IDEaliSM project is introduced. This high-level architecture forms the blueprint for the technical implementation of the advanced integration framework.

Figure 8 provides a schematic overview of the main components and their role within the overall IDEaliSM framework architecture. The architecture consists of three major components:

1. The **Engineering Library (EL)** is at the core of the framework. It is a repository in which the knowledge, tools and services of all partners involved in a project are made available;
2. The **Engineering Language Workbench (ELW)** is the environment for creating and adjusting the capabilities within the Engineering Library;
3. The **Advanced Integration Framework (AIF)** allows for logically arranging the available engineering library contents and executing the analyses within the product development process.

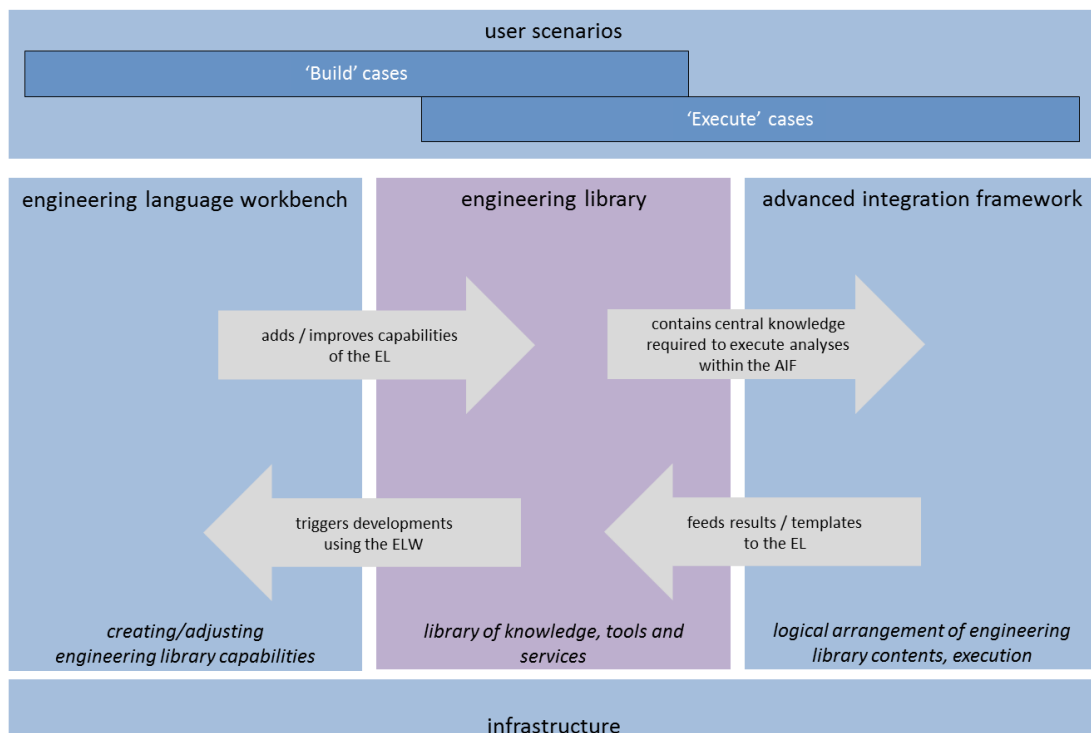


Figure 8: Basic view on IDEaliSM framework architecture, showing the main components and their interactions

Located within the Engineering Library, the engineering services as created during the first stage of the implementation of the envisioned PDP (see 2.2.1 for a definition) form the core of the framework. These generally applicable engineering services are created and adjusted using the capabilities of the Engineering Language Workbench. The Advanced Integration Framework on the other hand provides the means for interconnecting the individual engineering services from the library.

3.1.1 Technical enablers within the future PDP framework

This section highlights technical enablers being part of the IDEaliSM project, subdivided among the three major components of the introduced IDEaliSM framework.

Technical enablers within the Engineering Language Workbench

The following enablers allow for the structured creation and adjustment of engineering services:

- *A dedicated environment for the development of engineering services*: features a multitude of engineering service development toolkits and sets of libraries of standardized functions and languages. Each development toolkit is specifically dedicated to a certain problem domain;
- *A set of interfaces and standardized data exchange formats* to enable plug-and-play integration and interoperability of engineering tools, simulation workflows and business processes;
- *A set of standardized ontologies and graph-based design languages* to capture and re-use knowledge and compile this into engineering tools.

Technical enablers within the Advanced Integration Framework

The following list of capabilities allows for the integration of engineering services into simulation and business workflows:

- The formal definition of *hybrid workflows capable of integrating human-oriented business processes with automated tools and simulation workflows*. The business processes provide a web-based portal to end-users to share methods, tools and data irrespective of location. The simulation and design optimization workflows enable the batch execution of multiple engineering and analysis tools in a sequence using dedicated network technologies for data exchange;
- *A set of optimization algorithms and a supporting advisor* to generate feasible and efficient optimization architectures and processes for Multidisciplinary Design and Optimization;
- *A formal method for Product Data Management*, facilitated through a single source of data based on industry standards. This facilitates continuous integration of distributed development teams by live and up-to-date set of data throughout the complete development process;
- *A method for coping with Change Management and data dependency tracking* enabling indication of the impact of changes on the activities within the business processes.
- *A method for the cloudification of engineering services and workflows*, enabling unlimited scalability of the design spaces considered within predefined development durations.

Technical enablers within the Engineering Library

The engineering library mainly contains the following:

- *Reusable templates for multidisciplinary processes and simulation workflows* to improve the reuse and standardization and accelerate the configuration of new processes;
- *A library of engineering services* featuring plug- and play interfaces;
- *Reusable components, standard parts and materials and pre-existing solutions* to standardize designs and accelerate the design process.

The interested reader is referred to the specific deliverables on the technical features of the IDEaliSM framework as listed in Table 1.

3.1.2 Technical enablers during the staged implementation of the future state processes

The table below summarizes the identified technical enablers for implementing the service-oriented process methodology. It provides an overview of which technical enablers are required to implement each individual stage as described in the previous chapter.

Table 5: Relation between technical enablers and the staged implementation steps

Framework component	Technical enabler	Service-Oriented Process Methodology		
		Stage 1	Stage 2	Stage 3
Engineering Language Workbench	Dedicated environment for the development of engineering services	Forming the core of knowledge, engineering services need to be developed and maintained throughout all implementation stages of the service-oriented process methodology.		
	Set of interfaces and standardized data exchange formats		Although it does make sense to consider adhering to standardized data exchange formats already during the first stage, starting with the second stage this becomes of extra importance to ease interconnection of multiple engineering services.	
	Set of standardized ontologies and graph-based design languages	Especially when creating and adjusting a multitude of engineering services, standardized ontologies which can be re-used across the services provides a proper basis to quickly generate services and share general automation capabilities.		
Advanced Integration Framework	Hybrid workflows capable of integrating human-oriented business processes with automated tools and simulation workflows		In hybrid workflows, the tasks automated through engineering services are combined with the available manual services in logical process workflows. This allows for obtaining an ideal balance between automation and creative manual labour.	
	Optimization algorithms and a supporting advisor		Starting the second stage, multiple engineering domains get interconnected. An advisory system for setting-up workflows as well as dedicated optimization capabilities for obtaining optimal solutions on system level ensure the most effective usage of the multitude of available engineering services.	

	Product Data Management		During optimization cycles, the product data changes constantly. The product data management ensures the meta-information concerning the products components and performance retains available.	
	Change Management and data dependency tracking			Especially when utilizing the service-oriented process methodology to front-load the product development process, keeping track of changes largely increases the transparency of the choices made within the early phases of design.
	Cloudification of engineering services	Enabling unlimited scalability of the design spaces considered within predefined development durations.		
Engineering Library	Templates for multidisciplinary processes and simulation workflows.		In combination with the advisory system, workflow templates aid in setting up integration between the available engineering services.	Enables the adjustment and application of standard workflows to tailored product development processes.
	Library of engineering services.	To enhance collaboration, it helps when the created engineering services are made available in a company-wide library of services.	The availability of engineering services throughout the (company's) network represents the set of engineering capabilities for interconnection within workflows.	
	Reusable components, standard parts and materials and pre-existing solutions.			Especially when applying a front-loaded process methodology, the availability of reusable data is of large importance.

3.2 Organizational enablers

The implementation of the envisioned PDP by using IDEaliSM technical enablers as documented in the previous section also requires organizational adaptations. In the following sections, a number of organizational enablers as drawn from experience in industrial companies developing and already implementing IDEaliSM technologies are provided.

3.2.1 Commitment and continuity

The future state process is highly supported by engineering services, providing automation of tasks and ultimately enabling the ability to perform Multidisciplinary Design Optimization. These engineering services generally need to be developed for application to general analyses of general products and tailored to the specific problem during a specific design exercise. Furthermore, interoperability with existing (COTS) tools & methods within the engineering process is required. As such, developing and implementing engineering services requires investment (in the form of time of specialized employees, hardware and software for development, etc.). Since the intended business benefit is not directly observable, this requires commitment and continuous support and a clear definition of the intended business case by the Management Team (MT) of a company.

It is suggested to create a long-term roadmap for the development and implementation of such engineering services and to propagate this through all levels of hierarchy within a company. The creation and acceptance of such a roadmap needs to involve both technological, product as well as market (business development) representatives to link technological development to a product that is to be used for creating commercial advantage. The roadmap should focus on long-term financial benefits rather than limiting itself to the short-term increase in costs generated by building the engineering services making up the required knowledge in the engineering library of the product development framework.

Besides the long-term roadmap, it is strongly suggested to stick to a staged implementation plan (as presented in section 2.2), such that a first implementation already yields business benefits. This will significantly support management confidence, as well as raise the acceptance by users for which the daily routine will be changing in the updated PDP (see also section 3.2.6 on social change). The staged implementation approach also maximises chances to proceed to the full implementation of the front-loaded PDP. During the first implementation stage, part of the process is supported by engineering services whereas in subsequent steps more engineering services are developed and connected to each other, covering a broad engineering process and enabling optimization of the complete product involving all the required disciplines.

Finally, the creation of engineering services requires expert engineers to dedicate part of their working day for supporting the inclusion of product specific knowledge within the services. Until a productive state of the services is reached, this implies a lower operational availability of these engineers. It is not the intention to hire new co-workers doing 100% of the implementations required to achieve the envisioned PDP, since explicitly the expertise of the current- and experienced co-workers (where needed supported by knowledge engineers) is required in setting up the framework. It is the tacit knowledge that is most relevant to develop engineering services appropriately supporting the future development process; hence allocation of experienced co-workers is critical for success, supported by a continued MT strategic vision and support.

3.2.2 Change in company philosophy and financial models

Implementing one of the development stages of the advanced PDP will have direct financial impact on the short term. Depending on the process implementation step chosen however, this can even require a change of the overall financial model / philosophy as used within an organization on the long term. The impact is shortly described for each of the process implementation steps below.

In case of the first process implementation stage, engineering services are developed to automate repetitive, manual design tasks. From a financial point of view, the development of engineering services will involve financial investment into these future capabilities. These investments will impact the profit cycle in the short term by assigning engineering experts to development of engineering services without an immediate profitable outcome, see Figure 9. The financial- and productive benefits are reached in the long term when the engineering services are productively applied in the development process, lowering development times and effort as well as improving product quality through improved design processes. Focusing on the short-term profits in this case will impede innovation at its turn hampering the projected long-term improvement and growth.

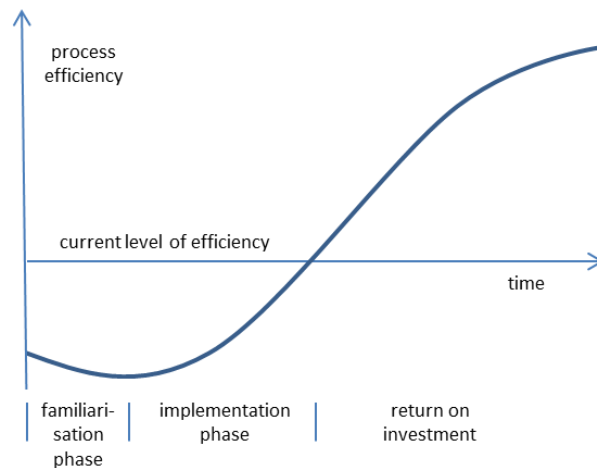


Figure 9: Typical investment / ROI for a development project

In case of the second process implementation stage, multiple engineering services are integrated into business- and simulation workflows. This requires an increased focus on investments in the early phases of the design process, to allow for more substantiated design decisions, design space exploration and MDO. Due to this however, a more flexible and efficient design environment can be achieved allowing faster design evaluations and a broader view of the design space; ultimately resulting in better performing products.

The final stage of process implementation, including the transition to the application of a front-loaded PDP in which (standard) solutions are being created even before program start, requires a change in the organizations' financial model and philosophy. This has been depicted in Figure 10, adopted from [2]. Based on information regarding new products, e.g. a new aircraft type, already some design space exploration can be executed in the frontloading phase of a program. For example, some key dimensions and a first guess for an Outer Mold Line (OML) can be input for automatic rudder design space exploration and multidisciplinary analysis. Standard solutions can already be compared and design parameter sensitivity studies can be conducted to identify the driving requirements or parameters early in the design.

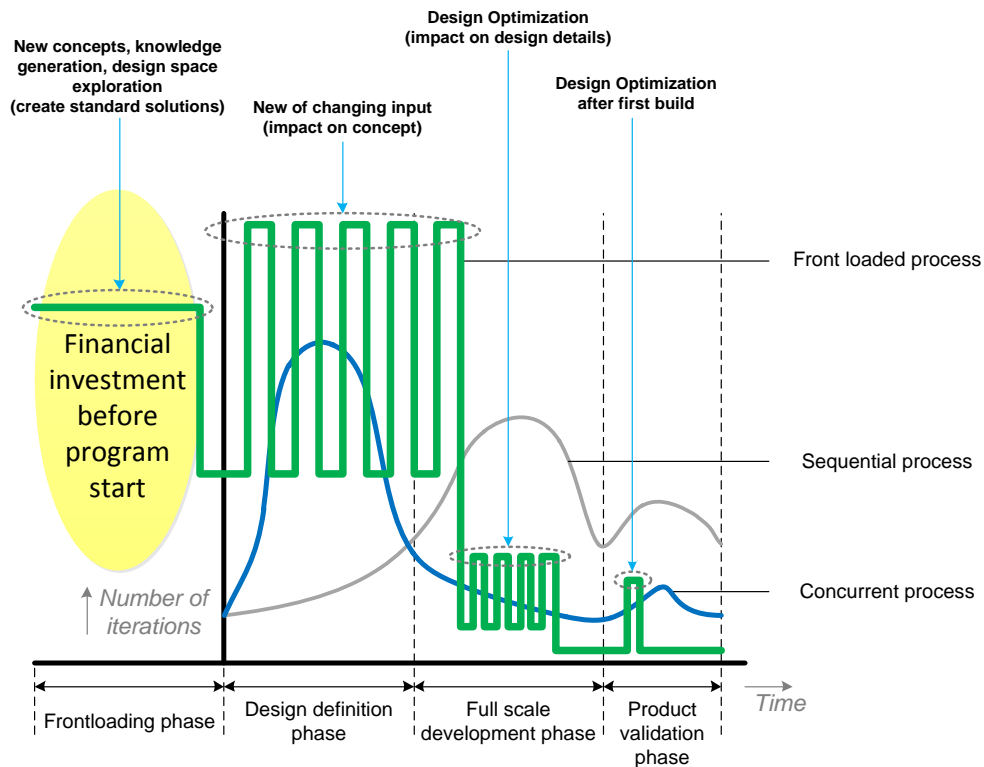


Figure 10: Investment before program contract / start (adopted from [2])

3.2.3 Effect on engineering capacity

Application of the innovations established within the IDEaliSM project in a productive work environment offers the opportunity to relieve engineers of non-creative and non-value adding activities. This leaves more time to focus on the optimization of the product, which should obviously in the end lead to better performing products or other related benefits.

One difficulty arises when observing that oftentimes the involvement of the same domain experts is needed for continuously improving the process by creating and adjusting engineering services and for operating and drawing conclusions using the established system. As actual design programs are generally prioritized over Research and Technology (R&T), this can represent a problem in achieving a full implementation of the new process. The way in which the engineering capacities are divided between technological research and operational application of the system should therefore be balanced.

Next to the involvement of domain experts for making their implicit knowledge available using engineering services, their experience is of key importance for Validation and Verification (V&V) as well as for the acceptance of the processes' results. Hence, it is vital that engineers are involved at all levels during the process development to ensure its relevance, ease of use, and applicability within its intended engineering domain.

As final effect on the engineering capacity within companies implementing the service-oriented process methodology, the resources required to cover maintenance and updates is mentioned. Since typically during the implementation phase the community of users increases the workload for maintenance and updates increases equally. Engineering involvement is relevant in case of required changes or extensions to the program, e.g. due to changed internal design rules or application for a different commercial program having specific requirements not covered before.

3.2.4 Effect on engineering capability

For enabling integrated product development using engineering services as described in the IDEaliSM service-oriented approach, the set of engineering capabilities within the company needs to be re-defined. Compared to the past and conventional PDPs, the roles and required capabilities of the company's engineers will change.

Next to continuing operational projects, part of the task portfolio of domain experts will be formalizing (implicit) knowledge as input for the development of generic engineering services. This requires a high level overview on ones knowledge as well as the capability to organize data, information and knowledge.

After development and integration of engineering services, the implementation and use of such services requires a shift in capabilities as well. Classically, engineers are operating within their specific engineering domains such as design, stress or manufacturing. Through experience, these engineers are specialized in the processes, tools and methods within their respective domain of expertise. A significant part of these engineers spent most of their time in manually operating IT tools and managing files. In the envisioned service-oriented PDP - especially in case when multiple integrated engineering services are connected within a front-loaded design process - engineers must learn to think in a multidisciplinary context by being able to judge results coming from various multi-disciplinary system analyses. Hence, engineers operating the engineering services need to understand systems of systems approaches and multidisciplinary correlations to be able to interpret engineering services outcomes as well as the results of system-level optimization from their perspective. The capability of interpreting the large amount of results produced by the system needs constant training and development within the company; even after significant parts of the process is being automated.

3.2.5 Effect on IT capability

For the acquisition and elicitation of knowledge and the following formalization into explicit code, well-skilled knowledge engineers experienced in both Knowledge Management (KM) and the Information Technology (IT) domain need to be consulted. These knowledge engineers have to team-up with available experts and have regular contact moments to elicit knowledge and verify the correct implementation of this knowledge into the engineering services.

Furthermore, the introduced approach requires a holistic view and IT landscape in support of the overall engineering process. At the basis of this, existing as well as newly developed tools need to be stored and made available using appropriate libraries for usage within readily accessible process integration and design optimization environments. Templates need to be developed and documented to provide the engineers with fast and easy to use plug-and-develop workflows. It is strongly advised to create separated infrastructures for application in the daily productive working environment and for setup and testing newly created or adjusted services and workflows. It is advised to create a corporate strategy for developing, operating and maintaining tools and services in an efficient manner. Such a high level strategy will ensure re-use of routines and sharing of lessons-learned throughout the company. In all, this brings along company-specific requirements to hardware, software maintenance as well as security. To get started with the IDEaliSM approach a certain starting investment in the IT infrastructure and architecture is thereby required.

3.2.6 Required change in company culture

Next to the organizational enablers mentioned in the previous sections, also a (slight) change in company culture is likely required to implement the described visionary PDP. Of all proposed organizational enablers, the required social change might be the most difficult to achieve.

First, large focus needs to be on showing that the integrated and automated approach at the basis of the envisioned PDP leads to opportunities instead of threats. Engineers need to be made aware that engineering services will be part of their daily routine and should be regarded as enablers and open up the possibility to focus more on creative tasks and product optimization; instead of spending valuable working time on performing the (manual) routine tasks themselves. The development of engineering services requires implicit knowledge to become explicit, which cannot be achieved when key engineering experts are not willing to share their knowledge and experience. In traditional organizations the notion of knowledge equals power might be relevant for part of the employees, making them reluctant to share what they know. To the authors' experience however, the role of the experienced employees changes, but definitely does not become less important. Interpreting results, constantly seeking for improvements in the processes and opening up unknown design spaces are among the many important tasks the engineers will have within the future processes.

Second, to achieve the required social change and to ensure trust in the developed design tools, the rules used in the engineering services and its application boundaries and possible results need to be transparent. The best resolution to this is making the rules itself as well as the traceability between the rules and the automatically generated product design accessible to the engineers. Furthermore, it is suggested to make intermediate results accessible at all times, to enable the engineer to assess the quality of these and allow for a comparison to the expected results based on experience [2]. In the end, trust is of vital importance for successful usage of engineering services in situations where engineers are still (personally) accountable for the quality of the final product.

When implementing the proposed improved process, engineers will work in a holistic design environment in which technical- and business parameters are integrated and multidisciplinary design effects are strongly present. A career will be a continuous learning experience and requires engineers to be open for new solutions and possibilities.

3.2.7 Organizational enablers during the staged implementation of the future state processes

Table 6 below summarizes the identified organizational enablers for implementing the service-oriented process methodology. The enablers are projected on the three stages of implementation as described in the previous chapter.

Table 6: Relation between organizational enablers and the staged implementation steps

Organizational enabler	Service-Oriented Process Methodology		
	Stage 1	Stage 2	Stage 3
Commitment and continuity	<p>Medium</p> <p>Existing company methods need to be updated and resources for automation to be allocated.</p>	<p>Medium</p> <p>Engineering services from multiple engineering domains should be interconnected and exchange information.</p>	<p>High</p> <p>Investments in creating a front-loaded design process required among all employees.</p>
Change in company philosophy and financial models	<p>Medium</p> <p>An initial financial investment is required for the development and integration of engineering services. With these services already a large ROI can be expected. These stages don't require a change in financial models of the company yet.</p>		<p>High</p> <p>Investments required up-front or early in the design process to ensure quicker and better solutions in later phases of the process.</p>
Effect on engineering capacity	<p>Low</p> <p>The opportunity for the creation of engineering services must be guaranteed next to the existing daily routines.</p>	<p>Medium</p> <p>More resources required for maintenance and updates of engineering services.</p>	<p>Medium</p> <p>More resources required for maintenance and updates of engineering services (since community of users increases).</p>
Effect on engineering capability	<p>Low</p> <p>Capabilities still of specialized, mono-disciplinary nature with focus on automation of repeated processes.</p>	<p>Medium</p> <p>Capabilities in performing multidisciplinary optimization and interpretation of results both within mono- and multidisciplinary engineering domains.</p>	<p>High</p> <p>Large focus on system engineers; focusing on design solution sensitivity, robustness and leveraging know-how from earlier projects.</p>
Effect on IT capability	<p>Medium</p> <p>Knowledge Engineers required to perform knowledge acquisition and engineering service development.</p>	<p>High</p> <p>Next to having knowledge engineers available, an IT landscape in support of the overall engineering process is required. Readily accessible PIDO environments need to be implemented.</p>	
Required company culture change	<p>Low</p> <p>More focus on creative design tasks through automation.</p>	<p>Medium</p> <p>Identification of product optimization opportunities in multidisciplinary engineering domains.</p>	<p>High</p> <p>Focus on validation and verification of the result and general trust the design solutions required.</p>

4 IDEaliSM example implementations and business impact

4.1 IDEaliSM example implementations

This section includes three implementation examples of the Service Oriented Process methodology based on the Use Cases provided by industrial partners and the IDEaliSM innovations as brought forward by the solution providers within the consortium. These Use Cases are linked to the Aerospace and Automotive engineering industry, although the Service Oriented Process is applicable to different high tech engineering industries as well.

The IDEaliSM Use Cases cover different process implementations with respect to the defined staged implementation approach. This highlights the flexibility to implement any of these stages including its dedicated technical and organizational enablers.

4.1.1 Use Case concerning “Aircraft predesign”

As part of the aircraft design challenge, this Use Case reflects the objective of Airbus Defence and Space to improve the established processes for early aircraft design by including multidisciplinary design optimization as well as increasing the degree of automation. Due to high uncertainties at this early phase coupled with limited knowledge about the aircraft concept’s properties, the process is based on a large number of small and lightweight tools, often based on statistics. This entails high manual effort in feeding the available information into the respective tools by repeatedly reformatting this data to comply with each tools’ input format requirements.

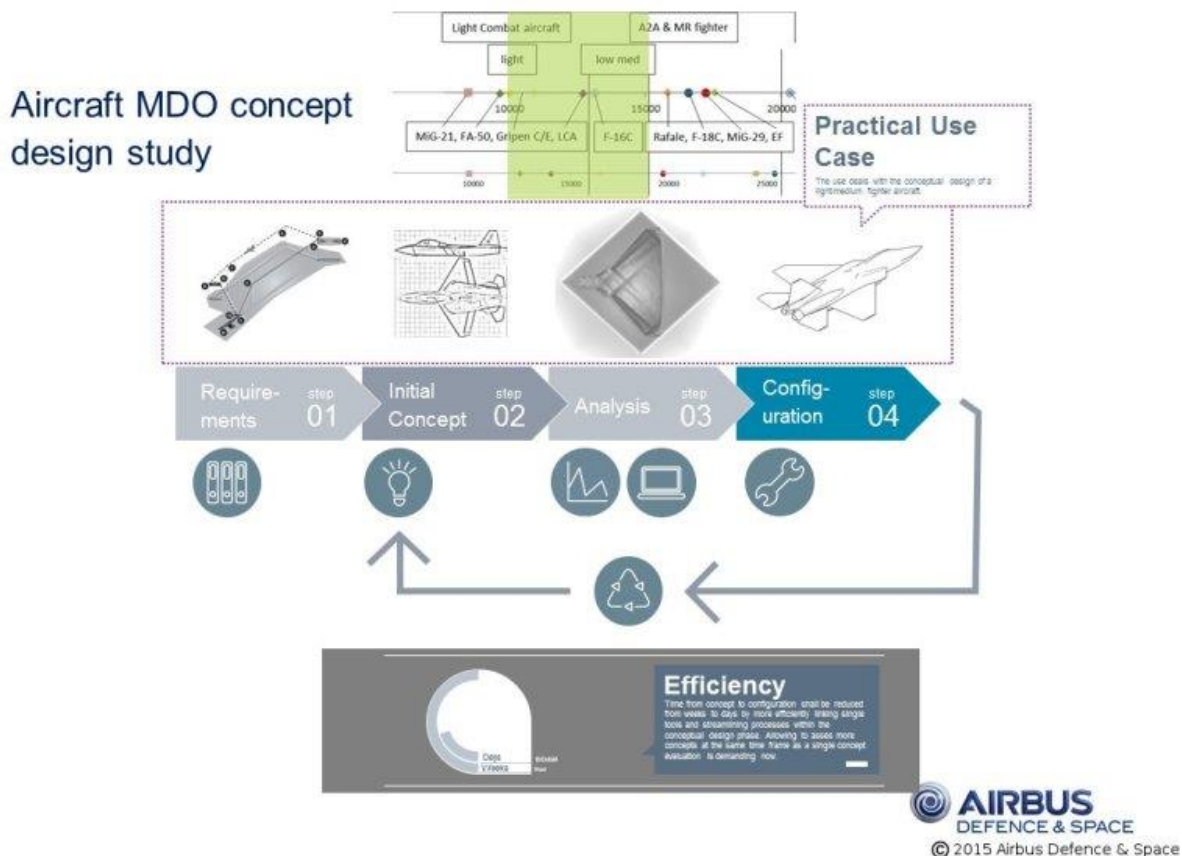


Figure 11: Conceptual design for a light / medium fighter aircraft supported by IDEaliSM

With respect to the staged implementation strategy, this Use Case covers mainly the first two implementation stages. In stage 1, a major step consisted of the introduction of the common parametric data exchange format CPACS. After extending all concerned analysis tools having data interfaces to read and write this data format, a large part of the formerly required manual workload of reformatting existing data to fit each analysis tools' needs could be eliminated. In fact, the respective tools were thereby converted in engineering services. The usage of the established engineering services made the process less prone to errors, since data does not have to be transformed several times into different formats along the process, reducing the amount of semantic interpretation errors. Furthermore, the process duration could be reduced considerably, as it enabled tools to directly work with the results originating from the previous engineering services.

Subsequently, based on the applied common data format, related engineering services could be combined into automated simulation workflows within a process integration and design optimization (PIDO) environment, allowing coupled execution of services. Next to allowing for further process speedup due to automated execution, application of further numeric solution finding methods such as parameters studies or design of experiments and even Multidisciplinary Design Optimization (MDO) was enabled. These innovations allow for better exploration of the available design space for new aircraft projects, due to which possibly better performing aircraft designs can be obtained than before. This corresponds with the benefits expected by implementing the first and second implementation stages.

4.1.2 Use Case "Rudder in a month"

As part of the aircraft design challenge, this Use Case reflects the vision of Fokker Aerostructures as Tier 1 to develop an aircraft rudder within the timespan of a single month to a level that corresponds to the normal results of the Full-Scale Development (FSD) phase up to the Critical Design Review (CDR).

While focussing on the design of an aircraft rudder, this Use Case extends up to and including implementation stage 3. First, different mono-disciplinary design tasks were formalized in knowledge rules and subsequently automated by the use of engineering software. The different tasks / services could be run independently, though input and output was to be linked manually and decisions on a next iteration step were still to be decided upon by the different disciplinary experts.

The next stage involved linking the different tasks / services with each other in a Process Integration and Design Optimization (PIDO) environment, input and output format is harmonized and linked with each other in such a way that the complete flow could be run as soon as the aircraft Original Equipment Manufacturer (OEM) provides its input. The flow can be run as a single design point but also Design Of Experiments (DOE) to identify the driving requirements as well as design parameters and finally design optimization runs can be performed.

The final stage of implementation concerned frontloading the overall design process. Due to application of the frontloading principle, the identification of driving design requirements concerning the product as well as finding promising candidates for given design spaces was performed even before the OEM has requested for a proposal. The goal is to explore and understand the different available concepts and their driving requirements and design parameters and to find optimized solution design spaces. This is done in such a way that by the time the OEM provides its input concept and design choices can be made much quicker without requiring extensive analyses and subsequently reducing the required response time.

The deliverables of each stage can be used independently and as such all have their added value.

4.1.3 Use Case “Vertical tail leading Edge design”

As part of the aircraft design challenge, this scenario entails the aircraft vertical tail leading edge design. In this Use Case, inputs are provided by Fokker Aerostructures as Tier 1 customer and passed on to IDEC as Tier 2 supplier.

With respect to vertical tail leading edge design, steps towards automation, optimization and integration have been developed, covering mainly the first two implementation stages.

In stage 1, the automation of existing (COTS) tools has been achieved. The leading edge design is typically required to be updated frequently during the design phase due to different geometrical modifications and changes, consuming much design process time and providing little added value. With the automation of the CAD and FEM processes using parametrization, sessions and templates, the implementation of changes and modifications has been interestingly accelerated.

These services have been linked in stage 2, in a Process Integration and Design Optimization (PIDO) environment. The workflow can be run as a single design and as Design Of Experiments (DOE) to finally achieve vertical tail leading edge design optimization. This new PDP enables IDEC to prepare quick commercial trade-offs and to be more competitive.

4.1.4 Use Case “Cockpit in three weeks”

The Use Case ‘cockpit in 3 weeks’ describes the overall innovation goal of DRÄXLMAIER: to develop an automotive cockpit wire harness within 3 weeks. The challenge in the automotive cockpit development is to integrate mechanical, electrical and electronic components inside the provided installation space including interconnecting the wire harnesses.

During the implementation of Use Case 3 the main focus was set on the development of engineering services to automate repetitive and manual, but also knowledge intensive design tasks, which refers to the 1st stage. Regarding the 2nd stage, which describes the integration of multiple engineering services, prerequisites have been researched and created. Regarding the 3rd stage, with the collection and storage of knowledge, which can easily be re-used to enable the re-execution of design tasks, first steps regarding front-loading the design process are also initiated.

In detail, the current process landscape has been examined and tasks suitable for automation have been identified, focusing on engineering service development. Furthermore, the integration of these automated tasks into a Process Integration and Design Optimization (PIDO) environment has been achieved. A first optimization scenario was implemented to run the automated tasks many times and execute designs of experiments (DOE’s) on a single domain. This proof of concept enables the implementation of more sophisticated scenarios in the future, ultimately regarding the application of multi-disciplinary design optimization.

In terms of the service-oriented process methodology, the overall process landscape was revisited and the automation through services was embedded in the future development process. By storing the knowledge and information in a design language, re-use of knowledge and re-execution of design tasks is enabled. Thereby initial prerequisites for enabling a completely front-loaded process for the product development have been achieved, enabling engineers to explore design spaces more intensively.

These successes in implementing engineering services and optimization scenarios can support to create awareness for the needed, more intensive future tasks to lift front loading to a next level, looking at single source of truth, data backbone, requirements engineering and configuration engineering, which will require high effort in restructuring and reorganizing processes and people.

4.2 Business impact

This section describes the impact of IDEaliSM innovations. By linking process bottlenecks, IDEaliSM innovations and business impact, the storyline as presented in Figure 2 is completed. Instead of Use Case specific impact, this section describes generic impact (based on evidence gained within the Use Cases) in a qualitative way. Quantitative and Use Case specific figures are included in separate project deliverables. Readers interested in Key Performance Indicators (KPIs) of each Use Case and its related quantitative impact as well as exploitable results are kindly referred to the documents listed in Table 7.

Table 7: References to IDEaliSM V&V documentation deliverables

ID	Title	Short description
D2.2	Process analysis and benchmark report	Describes current state process bottlenecks and benchmark measurement of KPIs, as performed for all three Use Cases.
D5.2	Integration framework analysis report	Describes the analysis and performance of the process as supported by the integration framework and will evaluate its performance against the identified KPI's from D2.2.
D6.5	Exploitation plan	Describes a general strategy for use of exploitable project results. Individual exploitable results of each project partner are included.

Section 2.3 already depicted the link between process bottlenecks and opportunities of the proposed future process, supported by IDEaliSM innovations. In the hereafter, the business impact will be appended in relation to these bottlenecks and innovations.

IDEaliSM innovations aim for a 50% efficiency gain as well as 50% time to market reduction (of the industrial processes in scope) for OEM, Tier 1 and Tier 2 companies in the high-tech engineering industry. These high-level KPIs are broken down into lower-level KPIs and related to current state process bottlenecks as well as IDEaliSM innovations. The results are summarized in Table 8.

The IDEaliSM innovations related to the first two current state process bottlenecks are primarily related to product optimization (e.g. weight / Recurring Costs (RC)) whereas IDEaliSM innovations related to the three remaining process bottlenecks enable Non-Recurring Costs (NRC) and lead-time reductions.

Each line item in Table 8, containing a process bottleneck with related IDEaliSM innovations and business impact, is discussed in the sections hereafter.

Table 8: Impact of IDEaliSM innovations w.r.t. staged implementation steps

Current state process bottlenecks		IDEaliSM Innovation			Business impact	
ID	Short description	Stage 1	Stage 2	Stage 3	Primary category	Key Performance Indicator (KPI)
1	Difficult to justify multidisciplinary effects of design decisions	-	<i>Advanced Integration Framework</i>		Product optimization	Increased product performance (e.g. weight) Reduced product Recurring Cost
2	Limited re-use of data, information and standard solutions (incurring changes)	-	-	<i>Engineering Library</i>		Number of design experiments within time Design space exploration Higher design maturity earlier Reduced effect of design changes
3	Repetitive, non-automated design activities	<i>Engineering Language Workbench</i>			Non- Recurring Costs & Lead-time	Reduced Non Recurring Cost (NRC)
4	Non-value adding design activities	<i>Engineering Language Workbench</i>				Reduced amount of human errors Reduced risk of budget- and time overruns
5	Silos of data, no single source of truth	-	<i>Advanced Integration Framework</i>			Reduced process / task lead-time Reduced design change lead-time

4.2.1 Impact of IDEaliSM innovations related to process bottlenecks

Starting with the first current process bottleneck limiting product optimization, the Advanced Integration Framework (AIF) covers the integration of people, processes and technology, which enables making substantiated design choices. Within the integration framework, multiple, multi-disciplinary engineering- and analysis tools are integrated into simulation workflows. Supported by optimization algorithms and an advisor for the generation of feasible and efficient optimization processes (see technical enablers in section 3.1), this enables Multidisciplinary Design and Optimization (MDO). With MDO, significantly increased product performance (e.g. weight, RC) can be obtained.

By including the business workflow layer within the integration framework, this provides the user the controls to inspect and/or change the design and objectively evaluate effects on product performance. The result of the optimization will be checked on being optimal (e.g. cost / weight) and robust within design requirements.

With respect to the second current process bottleneck related to product optimization, the Engineering Library (EL) includes the technical enablers to re-use data, information and knowledge by which, ultimately, the design process can be front-loaded. Using its library of engineering services, templates for multiple processes and simulation workflows and pre-existing solutions, many design iterations can be made very early - or even before the start of - a design program. This principle can extend to the Request for Proposal (RFP) phase, offering lower risk and higher fidelity in this phase, including the capability to even challenge the provided set of requirements. Different design concepts can be traded-off within a defined design space using techniques such as Design of Experiments (DoE), optimization and sensitivity analysis as well as to determine design space boundaries of a certain concept. Besides the ability to evaluate different design concepts on objectives such as product performance, this can extend to include objectives like 'robustness to changes'. In this case, this allows for the selection of design concepts that are least prone to changes, i.e. changes can be absorbed with minimum impact. This both reduces the effect of design changes in case these occur as well as reduces the required number of (iterative) design changes.

The third current process bottleneck is more related to Non-Recurring Cost (NRC) and lead-time of the design process, which is hampered by the existence of repetitive, non-automated design activities today. Here, the Engineering Language Workbench (ELW) provides the means to develop engineering services by its development toolkit including libraries and a set of standardized ontologies and graph-based design languages (refer to section 3.1). This allows complete (sub-) processes or specific tasks, suitable for automation such as repetitive-, frequently occurring and/or non-creative tasks, to be automated. This way, design process tasks can be quickly executed and design changes to be quickly incorporated. Hereby, the cost of executing these design tasks can be strongly reduced, just as its lead-time in case related and dependent design tasks are automated as well.

With respect to non-value adding design activities, one of the prominent examples today is the manual conversion of required data in the appropriate data formats as required by different tools. The ELW includes a set of standard interfaces and exchange formats through application of common data models by which the integration and interoperability of engineering tools, simulation workflows and business processes is enabled.

Finally, related to the fifth current process bottleneck, the advanced integration framework enables process optimization based on its technical enablers such as data dependency tracking, management of changes, and a single source of data. The latter facilitates the continuous integration of distributed development teams by providing a live and up-to-date set of data throughout the complete development process, both applicable to (product) data as well as tools, both within as well as between companies. This avoids errors and accelerates the development process.

Change management includes having intercompany indicators of (product definition) changes allowing users to keep track of changes, to (automatically) update related documents and models and to provide an indication of the impact of changes. Both primarily facilitate NRC and lead-time reductions.

4.2.2 Impact per staged implementation phase

This section includes the differentiation of business impact related to each of the staged implementation phases of the future state PDP.

In stage 1, engineering services target to automate labour-intensive and / or error-prone tasks. Thereby, main benefits related to the engineering services of this first stage are:

- Reduction in lead-time and NRC of the product development process through automation of repetitive tasks;
- Results of higher-quality and less rework due to the reduction of error-prone manual activities;
- Formalization of the rules and knowledge forming the core part of a company's experience (solving Knowledge Management challenges)

Within stage 2, the development and integration of multiple engineering services within a framework can obtain further lead-time and NRC reductions, as well as the opportunity to optimize the product (e.g. cost/weight).

The following main benefits are related to this second stage:

- Large scale reduction in lead-time and NRC in the product development process through:
 - o automation of a broader coverage of multidisciplinary (engineering) processes;
 - o increased interoperability (by standardized interfaces and data exchange formats) and flexibility of individual engineering services;
 - o cloudification of engineering services and workflows enabling unlimited scalability;
- Increased produced performance (weight, RC) based on the ability to perform MDO.

Finally, in stage 3, the framework is utilized to explore the design space and evaluate many design variants for different sets of requirements very early – or even before – a design program. This is used to identify driving requirements, explore promising concepts and especially to create a database of feasible solutions covering a large design space. The benefits of creating and utilizing such a 'front-loaded system' are:

- Higher maturity of data already during early design phases, offering lower risk and higher fidelity on Request For Proposals (RFP) and design concept decisions;
- Improved response to changing requirements possible, since fitting solutions can be directly retrieved from the established solution database;
- Drastic reductions in lead-time can be achieved, since available concepts only need to be detailed out in response to a RFP instead of being created from scratch;
- Design space scans are less prone to limitation due to time constraints. The possibility to find even more optimal solutions thereby arises.

5 References

- [1] ITEA Roadmap for Software-Intensive Systems and Services, 3rd edition, ITEA 2 Office Association, 2009
- [2] Berg, T. van den; Laan, T. van der; Rij, M. van; "Aerospace solution optimization through front loading techniques", Aerospace Structures and Materials, 2015, p1-12
- [3] Thomke, S.; Fujimoto, T., "The effect of "Front-Loading" Problem solving on product development performance", J PROD INNOV MANAG 2000; 17:128-142