

Chapter 2

Industrial Products and How They Are Developed

2.1 Man-Made Objects and the Laws of Nature

In our everyday life we are surrounded by and dependent on man-made systems and products. These are sometimes referred to as *artifacts*. We live in an artificial environment, we breathe air that comes heated and humidified out of the ventilation system, we use electronic equipment to communicate at distances out of hearing and we use transport systems to travel around the globe. When we venture a leisure trip into nature most of us bring special clothing and equipment to preserve our artificial environment as far as possible. The principles of natural and man-made objects are elaborated in depth by Nobel Laureate H Simon in his classical book *The Sciences of the Artificial*, (Simon 1996). Here we will discuss how the natural laws of physics will interact with the design rules of human origin in product development and what the implications are for the planning and functionality of a design automation system.

Natural laws are highly predictable and highly repetitive—if you throw a stone into the air, gravity will force it back to the ground with 100 % certainty. The same cannot be said for design rules created by man. The artificial laws that we impose on manmade things reflect intentions, and these intentions may well change over time. The intentional rules or constraints that we impose on a manufactured object usually reflect design knowledge based on intuition, experience or policy and are often termed *heuristic* rules. It is an important quality of a design automation system to be flexible in the sense that it allows easy change of the design rules, especially those having an artificial origin.

Of course all man-made products are subject to the inherent *natural laws* of e.g. physics or chemistry, but their properties and behaviour are also subject to rules defined by man. These *artificial laws*, represent the conscious *intentions* of the persons or organisations that designed and manufactured the products. Hence, the designer of a complex mechanical product is governed not only by constraints

originating from technical disciplines like structural mechanics, fluid dynamics, thermodynamics and so forth, but also by constraints originating from legislative or national standards, corporate policies, manufacturing system characteristics, customer specifications, or, maybe most importantly, the experiences and intuition of the individual designer. For the product to be successful neither category of constraints, natural or artificial, can be neglected.

To exemplify how these two categories of constraints interact and affect the design solution, let us consider a simple mechanical component like a threaded fastener in the form of a cylindrical steel screw. If the screw is subject to a tensional force F , has a radius r and the steel used has a maximum permissible stress of σ_y ; then the required dimension, the radius, can be calculated from:

$$r = \sqrt{\frac{F}{\pi \sigma_y}}$$

Assume $F = 10,000$ N and $\sigma_y = 250$ MPa, then a value for r of 3.57 mm will result. A screw with a diameter 7.14 mm would thus exactly be able to carry the required load. The result follows from elementary stress theory, i.e. a law of nature.

If the fastener had been part of a living creature and made of biological tissue the diameter would probably due to the evolution of the species converge towards this optimal value in the same way as a human thigh-bone is carefully sculptured to carry the loads that it is exposed to. For a part of a man-made object however, other considerations and design rules must be taken into account. Clearly nobody would want to use a screw with an odd diameter like 7.14 mm. Such a screw cannot be bought off the shelf, it would require special stock of spare parts and no service technician would possess a socket wrench that would fit the head. The designer would instead use a standard M8 screw, thereby slightly over dimensioning the component for the sake of standardisation. Behind this decision is an intent to follow metric fastener standards and company policy.

Does this distinction between natural laws and design rules of human origin have any specific bearing on issues related to design automation? Very much so, as we will see these differences have a profound effect on how to realise such a system.

First of all the two categories of design rules and methods usually require different methods of implementation. Computer tools for representation of objects under the influence of natural laws of physics are mainly directed towards mathematical modelling and simulation. These tools thus need to have powerful functions for analytical and numerical processing. The design rules and methods that are of human origin on the other hand, are frequently of a *heuristic* nature. By heuristic knowledge in the context of engineering is usually meant simple design rules discovered through experience or common sense, see Sect. 3.2 for a more exhaustive definition.

Typically the man-made rule sets are large compilations of rather simple relations and constraints, e.g. lists of allowable values or combinations, sometimes

with many levels of conditional statements and nested loops. Representation of the human design intent of a product design very often requires programming paradigms that are efficient for logical reasoning and combinatorial problems. The differences between the two categories of product design knowledge demand corresponding differences in the functionality of the computer programming paradigms used in a design automation application.

There is a second significant difference illustrated by the screw example. First assume that screw diameters in the range three to ten millimetres are allowed. The proposed diameter of 7.14 mm resulting from the calculation based on elementary stress theory is a real number. The solution space of real numbers is continuous with an infinite number of solutions in the permissible range. If we introduce a design rule of typical human origin saying that only metric standard screws are allowed, then only discrete diameter values may be chosen. The solution space is then reduced dramatically and in our example consists only of M3, M4, ... M10, i.e. the solution space consists of eight discrete numbers. For discrete solution variables within a limited solution space different types of *search* methods can be employed.

Limiting the solution space thus opens up for other programming paradigms than we are used to from traditional programming. For many design automation tasks computer methods based on some form of search strategy have proved very powerful. Let us return again to our example with the cylindrical screw. If we apply the search approach of *generate and test* here, we would start with the smallest screw, M3, calculate the arising stress and compare with the allowed value. If the resulting stress is below the maximum permissible value, this screw is selected, otherwise the next size is tried until a satisfactory size is found. This example is trivial and here the search based method has no real advantage over direct calculation and rounding, but for many ill structured design problems search based methods are very powerful, particularly from a computer programming perspective or may even be the only available method that works in practice. It should be emphasised again that these methods can only be used when the solution space is finite and consists of discrete solutions, which is often, but certainly not always, the case for problems governed by design rules of human origin.

2.2 A Generic Process for Product Development

The purpose of a design automation system is to simulate the product development process as it is traditionally carried out. Before we focus on design automation a brief orientation of this process is called for.

Product development is a process characterised by creativity, trial-and-error, subjective judgements, unexpected surprises and many difficult trade-off decisions. Should it be seen as a work of art or a process of science?

Management would much prefer to see product development as a scientific process. Experience however tell us that product development projects are notoriously

difficult to control. Time and cost budgets are frequently overdrawn and fulfilment of requirement specifications often becomes a matter of negotiations between the departments for engineering and marketing. In a survey of ten companies in the defence and supplier industries budgets were on average overdrawn by about 50 % and time gates by about a factor of two, (Svensson 1990). A large automotive project typically involves several thousand development engineers each with specialist knowledge of a particular part, system or property. Requirement specifications for such projects could be of the order of two thousand to three thousand measurable quantities (Sunnersjö 2003). All in all these large scale projects represent rather formidable efforts, which management cannot really control in a strict sense, but has to rely on the skill and judgement of others. Assuming that all project participants have sufficient skills for their respective tasks, the management challenge is to assure that the whole development team moves in the same direction and has a common understanding of what is to be achieved. This can be more easily achieved if there is a map of the process and this map is well known and accepted by everybody involved.

The map should take the form of a description, or model, of the product development process. In a survey (Nordström 1995) of how twelve Swedish manufacturing companies used ten popular product development methods it turned out that the only quality assurance method widely used and appreciated was when companies established a company specific product development model. The mere process of setting up such a model improves the process itself and the existence of a model clearly helps in making everybody aware of their own role in the process and how the different steps in the process depend on each other. It was generally accepted that the companies with the best development process will in the long run also produce the best new products.

The introduction of the concept of a product development model is usually attributed to Pahl and Beitz (1996). Their book, *Engineering design—a systematic approach*, first published in the German language in 1977, must surely be the most quoted publication in the engineering design literature. This book takes a significant step in moving product development towards a scientific process. It presents a detailed and logical step by step procedure for product development, parts of which was later used for the German standard, VDI-2221. Another significant contribution in the same direction was made by Hubka and Eder (1988), who introduced the concept of system theory into engineering design.

Both these works focus on engineering design tasks, which are central, but not stand-alone, activities in product development. Marketing, production planning, business activities, intellectual property rights and so forth are other aspects that must be considered. Also technical pre-studies, which are more general investigations intended to create a level of preparedness for future technical developments that may be required in new products, is part of the complete development process.

The concept of integrated product development (“total design”, “simultaneous product development”, “concurrent engineering”) became the generic panacea when planning product development processes in the early nineties with Pugh (1991) as one of the dominating text books. For most steps in the product

developing process there exist general design methods with the intention of providing stringency and effectiveness to the process. Two popular textbooks, both with a methodology focus, are Ulrich and Eppinger (2012) and Ullman (2003).

All product development models with appended methods present a simplified view of a very complex process. Nevertheless experience has proved them useful. Below an example of a simple model integrating design and manufacturing developments is presented, see Fig. 2.1. It has been used by the author for about 20 years and found acceptance where applied. It is a useful frame of reference when studying traditional development processes with the intent to set up a design automation system.

The process of Fig. 2.1 normally proceeds in a zig-zag fashion starting with the conceptual design phase generating a concept model, then moving to an inventory of production processes suitable for the planned product concept and resulting in a production process proposal. Then the process again moves back to the design leg with the preliminary design phase and so on.

In an ideal world the process will always move forward (double arrow, “synthesis”) passing through all the required stages. In real life however, product development inevitably becomes an iterative process with many loops (single arrow, “analysis”). Obviously we would want these feed-back loops, indicating that something must be corrected, to be few and short. It would be very unfortunate if a serious problem, that requires a change of the chosen concept, would be discovered first when the product has been launched on the market. Much time and money would then have been wasted and even worse, customer confidence might have been upset.

To minimise the risk of late changes in the product development process it is important to analyse the proposals for design and manufacture so that unsuitable solutions are discovered at an early stage. This analysis could be performed by e.g. computer simulations, scale model testing or prototype testing. Between all stages in Fig. 2.1 there are feed-forward synthesis arrows and feed-backward analysis arrows that constitute the two elements of the iterative process.

Before proceeding we need a somewhat more stringent definition of the two terms *synthesis* and *analysis*. In this book the terms are used according to the following definitions:

- **Synthesis:** The phase where an identified engineering task or problem definition is addressed in order to find a satisfying solution based on previous knowledge and expertise (Johannesson and Persson 2004).
- **Analysis:** The phase where the product or product part solution is evaluated based on its physical representation and characteristics. Importantly, the analysis is also an activity that precedes the synthesis phase as a problem definition is broken down into manageable parts (Dieter and Schmidt 2009)

To exemplify the concepts of synthesis and analysis, consider Fig. 2.2. The truss is defined by the rod dimensions, the rod lay-out, the boundary conditions and the material properties. Using these data the resulting stiffness, K , can be calculated by e.g. the finite element method. This is an example of *analysis*.

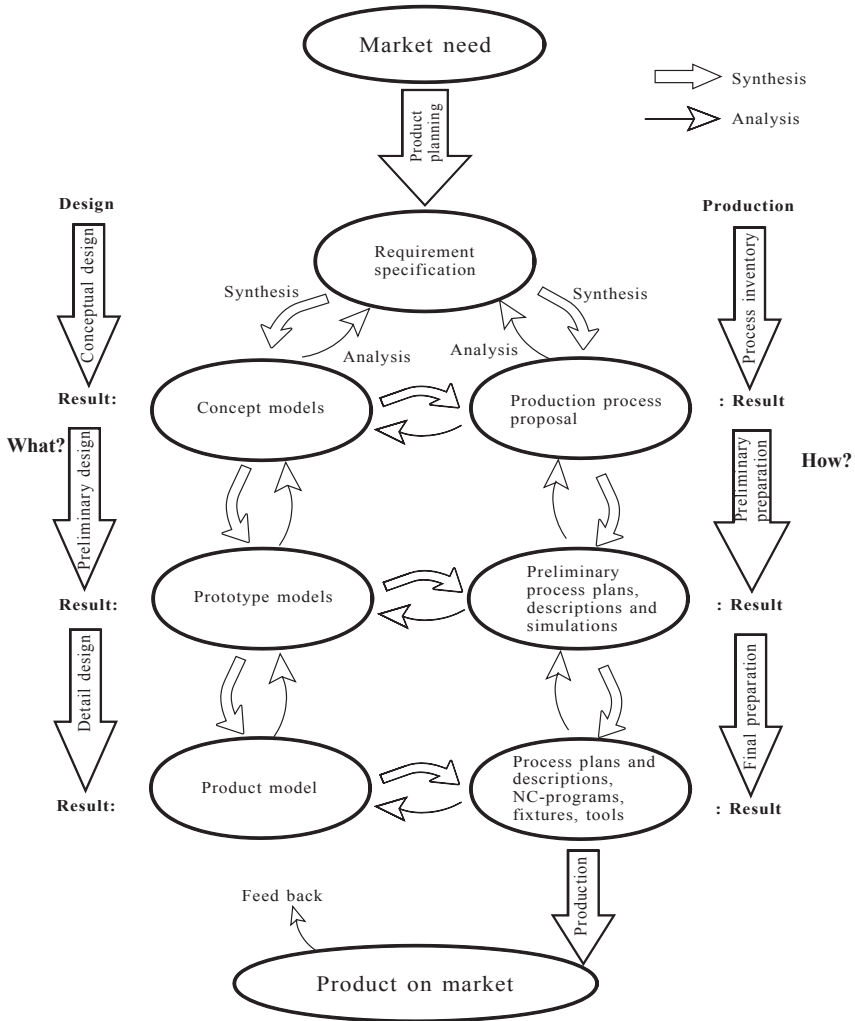


Fig. 2.1 Process model for product development that integrates product design (*left leg*) and production preparation (*right leg*). The process starts with a product plan from the marketing department and results in a complete definition of the new product and plans for its manufacture. Synthesis activities denoted by *double arrow*, analysis activities with *single arrow*. See also Rask and Sunnersjö (1998)

If we reverse the problem and instead start out with a required stiffness and length and want to define a truss structure that will provide this stiffness we have a *synthesis* problem. Clearly a large number of solutions exist, e.g. we could use thinner rods made of a stiffer material or arrange the lay-out differently. A synthesis problem is often governed by *constraints* that will reduce the number of possible solutions. For instance, it might be a requirement that all rods are of the

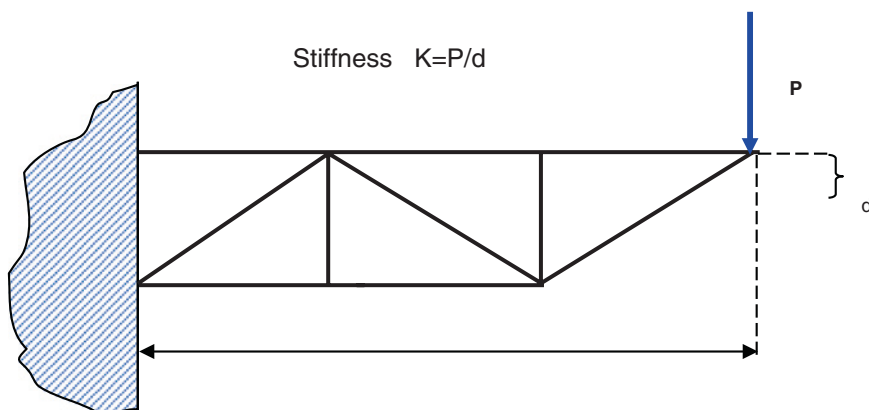


Fig. 2.2 Cantilever truss subjected to force P causing the displacement d

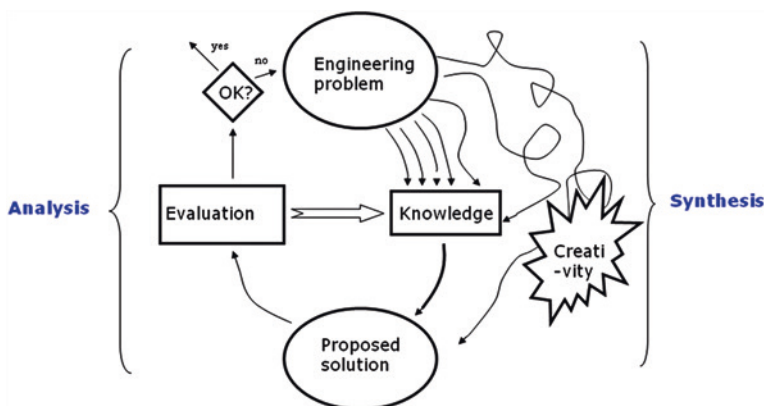


Fig. 2.3 Solving engineering problems through successive alternations between synthesis and analysis activities. See also Rask et al. (2000)

same diameter and material. If we are satisfied just by finding a solution that is consistent with the given constraints, we have a *constraint satisfaction problem*. If constraint satisfaction is not good enough, but instead the best possible solution is required, then we have defined an *optimisation problem*.

The alternation between creative synthesis actions and evaluative analysis actions is typical not only for the product development process but also for many other creative intellectual tasks. This iterative process is illustrated in Fig. 2.3. Starting from a problem definition, i.e. in our case a requirement specification for a new product, we apply our previous knowledge and experience from the specific technical field together with an element of creativity. Going from problem definition to solution proposal is a synthesis activity. As the figure illustrates the task at hand could be more or less straightforward depending on the necessity of creative, new

ideas. For design automation applications it is of course only possible to fully automate development processes where all necessary product knowledge already exists.

When the synthesis phase has resulted in a design proposal, this will be evaluated for functionality and compared with the original specification requirements. The evaluation implies analysis of the product and its functions which could be achieved by e.g. computations, expert review or prototyping. The analysis phase will result not only in a conclusion whether the specifications are fulfilled or not, but also in new knowledge about the product and its behaviour and properties. This is added to the already existing knowledge bank and improves the chance of succeeding in the next synthesis loop, should the previous one not having resulted in an acceptable solution.

2.3 Design Tasks with Potential for Automation

Over the last three to four decades amazing progress has been made on computational methods for analysis and simulation. Finite element methods, FEM, for structural and solids stress and strain analysis, Computational fluid dynamics, CFD, for flow problems, Boundary elements for potential field problems (e.g. acoustics) and rigid body dynamics, RBD, for mechanism analysis to name only a few. Corresponding progress in the other leg of Fig. 2.8, the synthesis tasks, has however been much slower. This is no surprise, synthesis problems are much more open, do not have a fixed solution path, have many possible solutions, are often ill structured and the knowledge base is made up by a mixture of rules of varying character and origin. Computational synthesis tasks seldom require heavy number-crunching. Instead the difficulty lies in the capture of human knowledge and human working methods within the narrow confines of the digital computer. Modern programming methods do however open up new possibilities to solve synthesis problems. The cases of applications must however be carefully chosen and this section will discuss what characterises problems that are potential candidates for automation.

2.3.1 Tasks Suitable for Automation

Selecting and defining the task(s) to automate is the first and foremost step when planning a design automation project. Figure 2.1 shows the complete product development process for original design from start to finish. It is rarely feasible to automate the process in its entirety. Instead typical design automation systems are intended for some kind of variant design and address only those parts of the process where the ratio cost/benefits is particularly favourable. It is often tempting to address too large tasks, but the key to success lies in finding appropriate limitations.

The engineering design process spans over a wide spectrum of tasks with different requirements of creativity starting with straightforward selection based on given criteria and leading to problems of original design requiring a high degree of innovation. A categorisation of design tasks according to degree of creativity was suggested by Ullman (2003) and is given below in an adapted form suited for the purpose of design automation. The list is given in ascending order of demands on creativity:

- **Selection:** Choice of individual component among predefined sets to satisfy specified constraints and objectives
- **Parametric Design:** Dimension driven geometry that adapts a predefined basic design (including topology variations) according to input specifications, formulas, methods, constraints or relations (template design)
- **Configuration:** Choice of individual components (from predefined sets) to be assembled into a system with specified properties (“catalogue design”). Choice governed by specified constraints and objectives.
- **Configuration of parametric components:** Combination of above
- **Redesign:** Includes work to adapt, modify, improve and optimise an existing design solution to fulfil new requirements
- **Original design:** The design task is defined by requirement specifications and given constraints but the principles as well as the details are left to the designer. Future variations of an original design concept belong to previous categories

These six categories of design tasks form a suitable starting point when discussing the applicability of design automation methods. Obviously we want to build such systems at low cost and at the same time reaping large benefits. How to achieve this will be elaborated more in detail in Chap. 8, but a few fundamental considerations will be discussed here. First of all the task to be automated must be chosen so that it is not mutually dependent on other tasks. If mutual dependencies with other tasks exist, these tasks will need to either be included in the same system or solved by human intervention.

One basic assumption is that all knowledge required to solve the design task is available directly or indirectly in a format that can be implemented by a computer program. A well-defined workflow for the design tasks that are to be implemented must exist or be possible to develop. These prerequisites constitute what is called a *computable design problem*. Another way of putting this, which is nearer to computer science terminology, is to say that the problem must be possible to solve subject to *the closed world assumption*. The implication of this assumption is that all facts are regarded as false unless they have a value that is either stored in the computer memory or possible to infer.

These criteria indicate that products based on mature, proven and well documented technology are good candidates for automation. The more stringent and explicit this product knowledge is, the easier it is to program it into executable code. Referring to the list above it could be concluded that the first four design categories, *selection*, *parametric design*, *configuration*, *configuration of parametric components*, are the primary candidates for automation. Original design is not

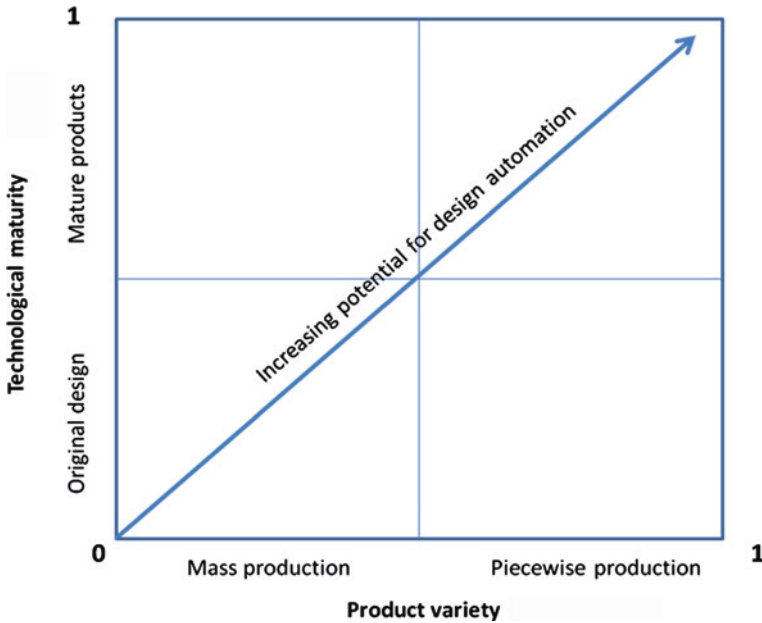


Fig. 2.4 The potential for design automation is high for products being technologically mature and being produced in many variants. The value 1 on the *abscissa axis* denotes that all products delivered are unique, while the value 1 on the ordinate axis denotes that all the required design rules are known and documented

a suitable application, while redesign offer some possibilities for more advanced forms of design automation.

The previous discussion concerned the feasibility or ease of building a design automation system for a given task. The other consideration when selecting tasks to be automated is what benefits that could be expected. Often customisation has a value in its own right and is the dominating driving force. Providing fast and accurate quotations for customer adapted products is a specific business factor where design automation might be crucial. Generally speaking, it is when a product will need to be redesigned in a large number of variants and when design cost or design lead time become a problem that it will be advantageous to use design automation methods. Figure 2.4 illustrates how the potential for design automation increases with product variety and technological maturity.

The previous discussion implies that variants are sold directly to the customer and that the ability to meet requirements of products adapted to customer requirements have a significant business value. There are however also useful applications of design automation where many variants are created for the internal development process but only the final version is actually sold to the customer. This kind of *iterative* design process often has the objective of optimising certain properties of the product and the design automation system becomes an instrument for this process.

2.3.2 Design for Variety, DFV

The process of product development will be different if detail design and manufacture is initiated by a customer order compared to when the product is designed, manufactured and stocked, waiting for a customer. The two alternatives are illustrated in Fig. 2.5. The first three steps are similar, but while the traditional, market based process continues until the product is launched on the market, the order based design makes a halt until a customer order with specifications is placed. Only then is detail design and manufacture carried out. The traditional market based process means that the marketing department takes the role of a customer

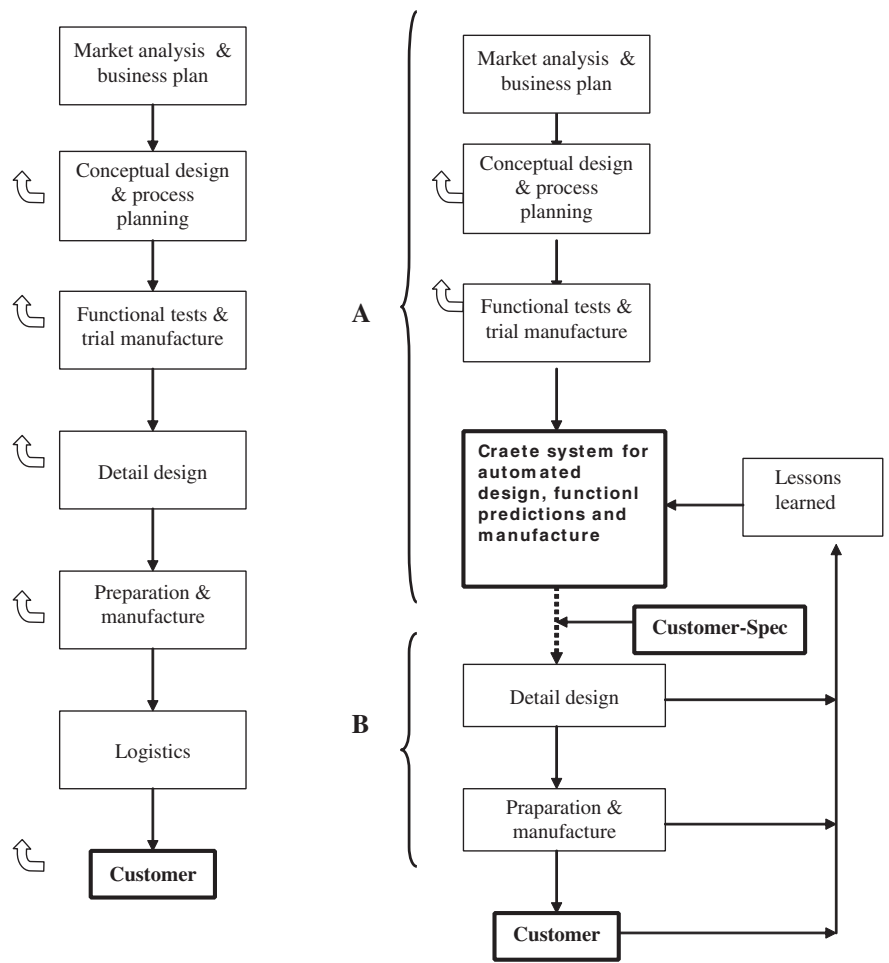


Fig. 2.5 Market based and order based design and manufacturing processes. Part A is carried out prior to order, part B when an order has been placed

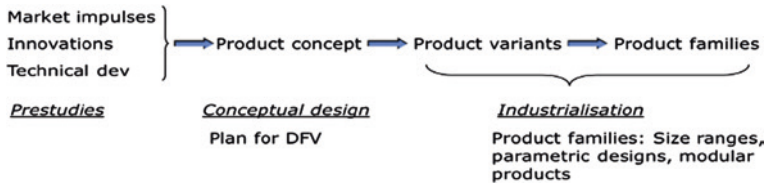


Fig. 2.6 Design for variety: from pre-studies to product families

representative and specifies the product trying to anticipate what the market preferences will be at the time the product is launched. Needless to say, considerable resources have been spent before the product meets the customer and if development time amounts to several years, there is a risk that the internal specification has not been on target.

The order based process avoids much of these problems but requires some sort of automated design system and a manufacturing facility with high flexibility. For products that are built in short series and high costs, e.g. ships, power plants, railway cars, production equipment etc., order based design is the self-evident way of working. But there are many degrees of variations available where the product platform is market based and the equipment is order based, e.g. in the motor car industry.

Standardisation is a core value in almost all efficient industrial processes. All experience indicate that industrial success is promoted by standardised products and procedures. However, as we have seen, with increased global business environment there is also a trend towards increased product variety. How can these contradictory trends be reconciled? How can flexibility and standardisation be combined? Design automation methods have an important role in this respect, but of course, the product itself must be designed in such a way that the required variations can be realised with relative ease. The methodology to achieve this is called *Design for Variety*, *DFV*, and the corresponding industrial process is illustrated in Fig. 2.6.

From a business point of view one can distinguish between four categories of design variations.

- **Select from stock:** All available variants are designed and manufactured prior to customer order
- **Configure to order:** Standardised modules and components are assembled in a variable product structure
- **Modify to order:** Use of a generic product structure with adaptable modules and components
- **Engineer to order:** Use variable product structure and adaptable modules and components

Customer influence and customer involvement in the product development process is low at the top of the list and increases as we move downwards. For *engineer*

to order contracts, e.g. in shipbuilding, power plants and production equipment, the customer usually plays a very active role and often defines requirements and choice of technology. At the top of the list, it is up to the manufacturer to predict what preferences the potential customers will have when the product is ready to launch. The latter situation, e.g. for production of motor cars or house-hold appliances, involves a higher risk because much has been invested in the new product before it is launched and the market response can be evaluated.

Irrespective of what business case is at hand, in order to be successful in DFV it is important to plan for *product families* and not for singular products from the beginning. By “product family” we mean a group of related products that to a high extent share the same features, components or modules. A viable product family concept will consist of products where a large proportion of the products have a common set of features, components or modules that remain the same within the whole product family. This common set is called the *product platform*. If a high proportion of the products are made up of this platform the product family is said to have a high degree of *commonality*. If standardised features, components or modules are used in different platforms these are said to *carry-over*. Of course, methods like commonality and carry-over save resources and should be exploited when possible without sacrificing the variety that the market demands.

To satisfy the requirements of a DFV process, the “Product planning” phase of Fig. 2.1 should thus result in a specification for both the product platform and the complete product family, i.e. the full range of planned products. It will be much more difficult to expand this range at a later stage. A good rule is to always aim for exploration of the principles of standardisation when planning the product range.

2.4 Investing in New Products

Development of complex products is a costly process with inherent risks. For a manufacturing company however, updating of its product range is a necessity if the company hopes to survive in the longer perspective. To give an indication of the economic aspects of product development some examples from the motor car industry will be given below.

A rule of thumb for the relative development costs attributes about 2/3 of the total budget to product development, i.e. the left leg of Fig. 2.1, and 1/3 to what is often referred to as the *industrialisation* phase, which is the right leg plus acquisition costs for tooling, production equipment, premises and so forth.

For the total project development costs some examples from the history of the automotive industry are given in Fig. 2.7. Note that the vertical axis is graded in a logarithmic scale, which means that the increase in development costs, which appears to be linear with time, in reality is exponential. The trend points towards twofold cost increase over 15 years and in 1995, when the graph was made, development costs accounted for between 5 and 10 % of the turnover for typical mass producing automotive manufacturers.

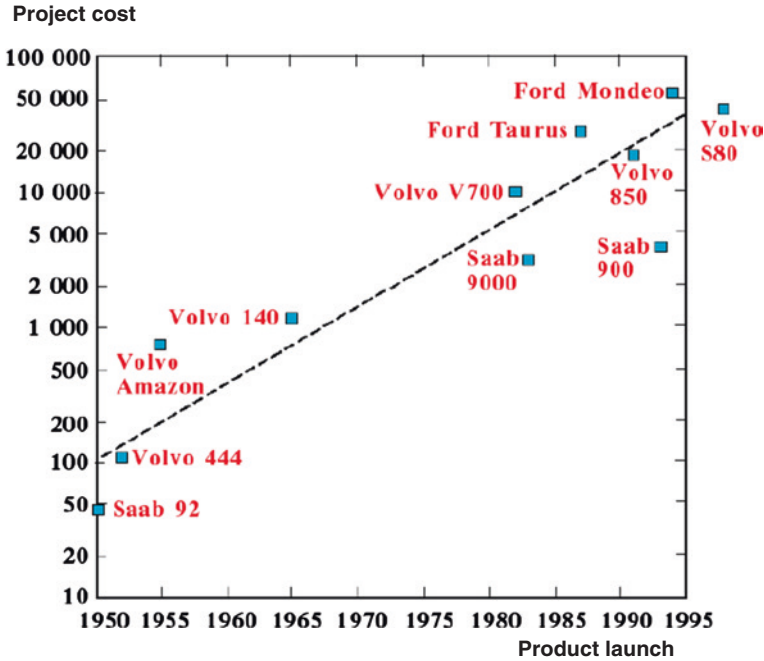


Fig. 2.7 Total project costs for car models launched over 50 years. Costs in million SEK (one SEK approx. 9 euros), compensated for inflation and normalised to 1995. Numbers are rough estimates based on information reported in news media and interviews with development personnel

Although the costs given are rough estimates based on indirect information and some rather crude assumptions, the general trend of costs escalating out of proportion is indisputable. This is of course a situation that cannot be sustained in the longer perspective and the industry has also applied various approaches to arrest the accelerating development costs, which at this time seemed to be out of control. Apart from a general strive to improve efficiency in all phases, most methods boil down to an effort to distribute the heavy development costs over as many units as possible.

Popular concepts like *outsourcing*, *commonality* and *carry overs* have proved their worth to control cost escalation during the last two decades. The methods all have in common that they, in different business scenarios, seek to reuse existing, proven designs, adapt them to new specifications and avoid costly tasks of original design starting from scratch.

Take outsourcing as an example. This approach implies that the car manufacturer abstains from the responsibility of developing new technology and new designs for those parts of the car that are not considered to contribute to the *core value* of the brand. Instead a supplier takes this role and specialises in certain components. This supplier will usually have several industrial customers who want much the same

component, but with various specified adaptations. The core technology is the same and many parts thus share the technical development costs. It is then up to the supplier to be able to adapt the design correctly, quickly and cheaply so that each customer specification is satisfied and, recognising the archetypes of variant design as described in Sect. 2.3.1, this is where DFV by design automation has an important role to play.

Commonality is the term used where many models in a *product family* to a large extent share the same components and *carry-over* is when parts and modules from previous designs are reused in new projects. In these three efforts to take control over accelerating development costs, the concept of design automation has the role of enabling technology. Adaption and reuse of proven design solutions saves engineering time, but this is not the main saving. If proven solutions can be used the heavy costs of prototyping, testing and regulatory approvals can often be avoided. Combining reuse of proven technology with state of the art design automation systems is a very competitive concept.

References

- Dieter, G., Schmidt, L.: Engineering Design. McGraw-Hill, New York (2009)
- Hubka, V., Eder, W.: Theory of Technical Systems. Springer, Berlin (1988)
- Johannesson, H., Persson, J.-G., Pettersson, D.: Produktutveckling (Product Development). Liber, Stockholm (2004)
- Nordström, L.: Kvalitetsmetoder i konstruktionsarbetet (Quality Methods in Engineering Design), vol. 1. Swedish Association of Manufacturing Industry, Stockholm, Sweden. Faktarapport No 1103-7067. (1995)
- Pahl, G., Beitz, W.: Engineering Design—A Systematic Approach. Springer, London (1996)
- Pugh, S.: Total Design. Addison-Wesley Publishers, UK (1991)
- Rask, I., Sunnersjö, S.M., Amen, R.: Kunskapshanterande IT-system för produktframtagning (Knowledge based IT systems for Product Realisation). IVF Publication No 00823, Gothenburgh, Sweden (2000)
- Rask, I., Sunnersjö, S.: Konceptkonstruktion—Val av material och tillverkningsmetod (Conceptual Design). IVF, Göteborg (1998). ISBN 91-89158-08-3
- Simon, H.: The Sciences of the Artificial. MIT Press, London (1996)
- Sunnersjö, S., Rask, I., Amen, R.: Requirement-driven design processes with integrated knowledge structures. In: Proceedings of DETC03, ASME 2003/CIE-48218, USA (2003)
- Svensson, P.: Management of product development. PhD thesis, Chalmers university, ISBN: 91-7032-479-4 (1990)
- Ullman, D.: The Mechanical Design Process. McGraw-Hill, New York (2003)
- Ulrich, K., Eppinger, S.: Product Design and Development. McGraw-Hill, New York (2012)

Intelligent Computer Systems in Engineering Design
Principles and Applications

Sunnersjö, S.

2016, XVII, 156 p. 92 illus., 52 illus. in color., Hardcover

ISBN: 978-3-319-28123-0