Chapter 10

The Dilemma of Managing Iterations in Time-to-market Development Processes

M. Meboldt, S. Matthiesen and Q. Lohmeyer

10.1 Introduction

This paper considers the authors' experience in industrial practice and reviews it from the point of view of scientific discussion. From scientific point of view there are three major Research questions: Why are iterations conceived differently? What makes an iteration valuable or harmful? What are appropriate strategies to deal with iterations under time pressure? The Papers give answers on this major research question by showing different aspects of time-to-market development processes and the challenge of effectively handling iterations within them. One of the authors was development head of a business unit responsible for various development and innovation projects. The other, in his position as global process manager for research and development, was responsible for the design and improvement of development processes in the same company. Thus both views represent the conflicting aspect of process modelling and iteration, which is a key topic in scientific discussion (Table 10.1).

10.1.1 Delimitation of Product and Process Area

Most publications and research in this area are concerned with huge development projects, *e.g.* in aircraft or automotive companies. These development projects are characterised by large development teams and stringent external safety regulations. In general there are only a few publications reflecting iterations in practice. For example Wynn *et al.* (2007) remark that there are only rare examples of the correlations between specific design reworks and project delays. One of the cited time-to-market projects is from the Airbus A380.

The authors' experience is based on design practice in small design and projects teams (3 to 15 development engineers, and project teams with up to 30 people). In addition to this, the external safety regulations for product approval are at a lower level. The major difference is that the decisions about product quality and maturity are made within the company itself. It has to be investigated in further research how these experiences can be compared to development projects with large teams and stringent external regulations.

Table 10.1. Different views of process modelling and iterations

	Process	**Iterations**
Development point of view	Every development process is unique and defined by the challenges resulting from the occurring problems, which are always different.	Iterations are day-to-day business. They are essential learning cycles in development that allow a continuous gain of knowledge and thus a mitigation of uncertainty and ambiguity.
Management point of view	The process model is a company standard and mandatory framework for risk mitigation. It is essential to effectively steer communication, resources and investments in the company.	Iterations are expensive exceptions. They cost time and money. Iterations have to be eliminated: The goal is a zero-iteration process as defined by "Do it right first time".

10.1.2 Iteration in the Context of this Paper

A time-to-market development process is about designing, producing and launching successful products in order to make money. During this process, a lot of decisions are made on vague assumptions and estimations. Later in the process these decisions are verified or found to have been false (falsification). A successful verification approves former assumptions, whereas a falsification always leads to minor or major iterations. On one hand these iterations are time and money consuming, on the other hand they are important to gain knowledge about the specific issues in a project. Thus, a falsification leads to learning cycles and in consequence decisions made are overruled and changes have to be implemented.

This paper focuses on development iterations in the product life cycle caused by the following areas: 1. technology and production issues, 2. market issues including competitor situation and patents, 3. changing company situation and strategy, 4. changing applications and external regulations made by customers or suppliers. Here, the following types of iterations can occur:

- worst case iterations: iterations triggered by issues occurring after the market launch - the consequences are product recall and loss of reputation;
- serious iterations: issues requiring the change of a decision from a previous development stage - for example a patent situation requires a conceptual change after the design freeze;

- targeted iterations: iterations within a development stage, which do not impact on previous gate decisions and do not jump back to a previous stage - these iterations lead to product maturity (Krehmer *et al.*, 2009).

Based on a research overview the paper shows how process modelling is adapted to a company process model. It describes the different aspects of iterations and escalations in regard to the project objectives. As a conclusion at the end of the paper, the authors give recommendations based on a good practice approach.

10.2 Models of Iterations

In the literature there are several definitions and classification proposals for iteration. According to Costa and Sobek (2003) a common approach is to consider iteration as repeating design activity. Most definitions in research state that iterations describe a cycle of gathering information, processing that information, identifying possible design revisions and executing those revisions in pursuit of a goal.

Wynn *et al.* (2007) differentiate between six non-orthogonal perspectives of iteration: exploration, convergence, refinement, rework, negotiation and repetition. These implicate differences in the understanding of iteration between the technical and management contexts. Le *et al.* (2010) summarise that authors focusing on the negative effects of iterations refer to unproductive rework, which can be caused by factors such as flawed design and inadequate quality assurance. Authors reporting the positive effects focus on iteration as being necessary to systematically explore and understand the complexity of design problems and their potential solutions, thus leading to a more efficient solution finding process.

10.2.1 Basic Models of Iteration

One of the basic models of iteration is the TOTE unit according to Miller *et al.* (1960). TOTE stands for Test-Operate-Test-Exit (see Figure 10.1). It is a model from cognitive science that represents an iteration as a continuous evaluation-action process that proceeds until a test sequence yields a positive result. The TOTE unit is applied in psychology, cybernetics and artificial intelligence to represent problem-solving processes.

Another basic model is the PDCA cycle according to Deming (1986). The model describes iteration as a cycle of the four generic activities ‘Plan’ (design the product), ‘Do’ (manufacture the product), ‘Check’ (test it in service, through market research) and ‘Act’ (put it on the market). Nowadays the model is applied to the improvement of processes, products, and services in several organisations, as well as to improve aspects of one's personal endeavours (Langley *et al.*, 2009).

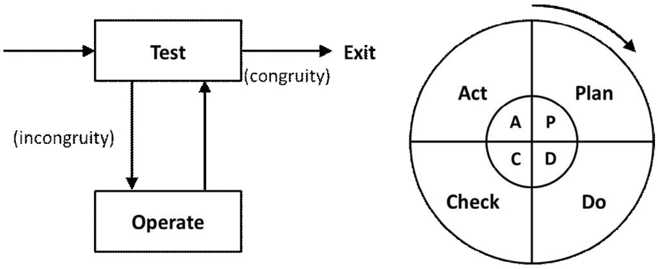

Figure 10.1. Basic models of iteration: TOTE unit (left) and PDCA cycle (right)

10.2.2 Advanced Models of Iteration

The Unified Innovation Process Model for engineering designers and managers according to Skogstad and Leifer (2011) depicts the design process and explains how its participants' actions affect it. The model considers three central activities: 'Plan', 'Execute', and 'Synthesise' that occur repetitively during every phase of the design process (see Figure 10.2). The model also includes three feedback paths: 1. Re-planning signifies the action taken by designers when the results gained during synthesis are so different from what was expected that they must return to planning and change their approach. 2. Revision occurs when the results of synthesis are not sufficient to qualify as a solution, but are not so far off that the overall approach has to be changed. 3. Reworking is the process of re-executing until the output is satisfactory enough to advance to synthesis.

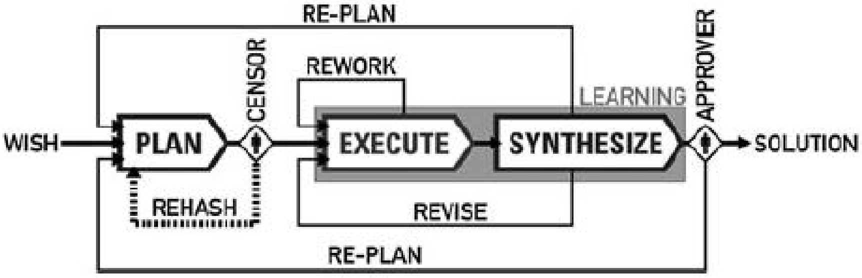

Figure 10.2. Unified Innovation Process Model according to Skogstad and Leifer (2011)

The Advanced System Triple according to Albers *et al.* (2011) describes product development as an iterative process of synthesis and analysis of both the system of objectives and the system of objects (see Figure 10.3). The model represents the designers as the operation system.

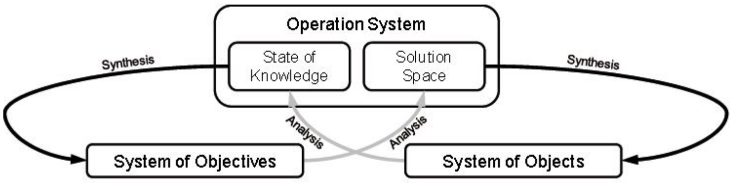

Figure 10.3. Advanced System Triple according to Albers *et al.* (2011)

Based on their knowledge and additional information gained from relevant objects, they make their first decisions about product-specific objectives. So they build up an initial system of objectives that frames a vague solution space. The solution space is a designer's mental model and therefore an element of the operating system. In the next step the designers have to find different solutions, where every solution embodies certain decisions. Solutions are described by virtual or physical objects that in turn lead to new information, which the designers use to refine the system of objectives. By repeating this refinement cycle, uncertainty is progressively mitigated.

10.3 Management of Time-to-market Processes

Development processes in companies are not static, they are subject to continuous improvement. Processes are adapted to changing internal and external needs and requirements. And the optimisation target is always to keep the non-value-adding activities to a minimum. The challenge is to reduce losses within development projects, by eliminating non-value-adding activities - inevitably this sometimes applies to iterations. A development model is always company-specific and adapted to the company's development philosophy and culture.

In industry, stage-gate models are the leading standard. In the nineties these "deliverables-oriented" management approaches replaced the "activity-oriented" methodological engineering design approaches as the leading standard in development processes. The key idea of these approaches is that between every stage of the process there is a gate which includes a management review. These validation points review the results and approve entry to the next stage (Cooper, 1990). The key purpose of these approaches is to steer investment and communications with other departments and partners. With the stage-gate approach the paradigm for process models changed from "the model tells the teams what they have to do" to "the model only describes, what the teams have to deliver at the gates".

Early stage-gate approaches included a large number of gates. These development process models had two major issues:

- many non-value-adding iterations: these detailed process models caused a lot of cycles. The key insight was that development activities cannot be separated into small units without increasing iterations which revise decisions from previous gates;

- decisions were shifted to management: The degree of freedom of the project team was limited and highly steered by the management. As a result teams worked to fulfil tasks not to set and achieve objectives.

After major reworks most processes were reduced to models with five or six stage-gates. Process design is always on a thin red line between loss of productivity due to non-value-adding bureaucratic process work, and loss of productivity caused by insufficient communication and rework.

10.3.1 Time-to-market Process Model

The time-to-market (TTM) process model used in the company the authors worked for describes a six stage-gate development process (see Figure 10.4). TTM processes are about bringing a product to market and ensuring return on investment. These processes start on the basis of a technologically mature product concept. By Gate 2 (G2), at the latest, the product prototype has to achieve 80% of the performance at 120% of the cost of the final product. Thus, at a very early stage in the process 95% of the detail design is already done, including tolerances, testing and manufacturing. From this point on iterations start to become critical.

At every stage of the process pre-defined key deliverables have to be fulfilled. For example, one of the key questions in Gate 3 (G3) is: Has the value proposition been confirmed by a customer acceptance test? Usually most iterations occur just before a gate, because all critical information and test results converge in gate decisions. This can be described as a “bow wave effect”. Critical topics and open questions are pushed towards the next gate decision. This effect masks the issues at the beginning of the next stage. In order to make robust decisions, it is highly important that the iterations occur early in the stage, right after the gate. If they occur early, there is still time to solve issues and optimise the design within the stage. In consequence it is important not to prevent iterations but to provoke them very early in a stage.

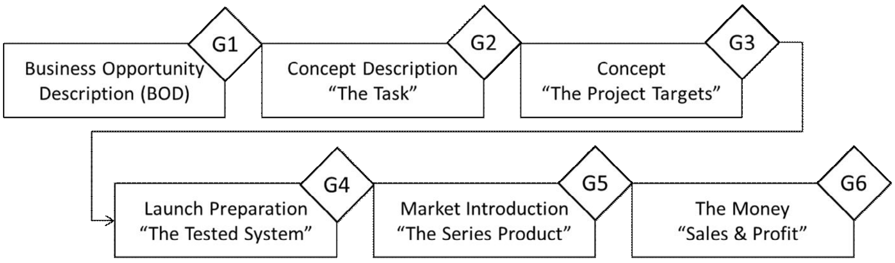
Business Opportunity Description (BOD)
G1
Concept Description “The Task”
G2
Concept “The Project Targets”
G3
Launch Preparation “The Tested System”
G4
Market Introduction “The Series Product”
G5
The Money “Sales & Profit”
G6

Figure 10.4. Time-to-market process model according to Esquius Serra (2010)

10.3.2 The Dilemma of Managing Iterations

The dilemma of iterations is whether an iteration can be predicted or not, and if an iteration should be included in the project plan or not. The central management assumptions are that, if an iteration is predictable, then it is preventable, and if an iteration is not predictable, then it is not projectable. From the management point of view planned iterations, without specific content are regarded as buffers. Buffers are “squeezed out” of the project plan for the very good reason that they inevitably extend the project duration.

The following dialogue gives an insight into this dilemma between a development engineer and the management steering board of a project. The development engineer, who is the technical project manager, is asked to present the project plan for the development of a new product. He planned three iterations in stage 2 until G2 (concept freeze) and in stage 3 until G3 (design freeze). Management asks him to report on the specific reasons and contents of the iterations and what he is proposing to avoid them in order to reduce development time: “The planning of these iterations is based on my experience. We need them to solve unexpected technical problems, we do not know yet, what they are about, but they will occur.” The management replies: “We need to shorten the development time, in order to be first on the market. An early market launch is directly linked to the return on investment that we can achieve with this product. You will get all the resources you need. Please specify these unknown problems and the additional competences you will need in your team so as to solve them before they occur.”

10.4 Two Fundamental Types of Iterations

In early development stages the issues of iterations are not critical, because there is still time to react flexibly. At that point, when the market entry communication (at G3) or the investments for series production tools (at G4) are made, the situation changes completely. After this point every iteration is critical. Based on our experience most of the iterations occur after this point (between G3 and G5). In this context we differentiate between two fundamental types of iteration. 1. In-stage iterations (these are iterations within a stage which do not impact on previous gate decisions); and 2. cross-gate iterations (iterations which affect decisions from previous gates and have an impact on investments and market launch).

10.4.1 In-stage and Cross-gate Iterations

In-stage iterations are learning cycles. Because of this it is important to provoke them very early in a process stage. Here, the most effective strategies are validation and system integration under realistic boundary conditions. Iterations must not be understood as doing the same twice, it rather is about having clear hypotheses of the iterations’ outcome in order to verify them or declare them false. Provocation of iterations is an objective-oriented step to improve the overall system. These iterations are essential to find the best solution.

Cross-gate iterations are learning cycles as well, but they are expensive in cost and time, so they have to be prevented. If functional issues occur early in the process the probability of an impact on other stages is small. It is important that the process and deliverables encourage the team to look at critical issues and test them as early as possible.

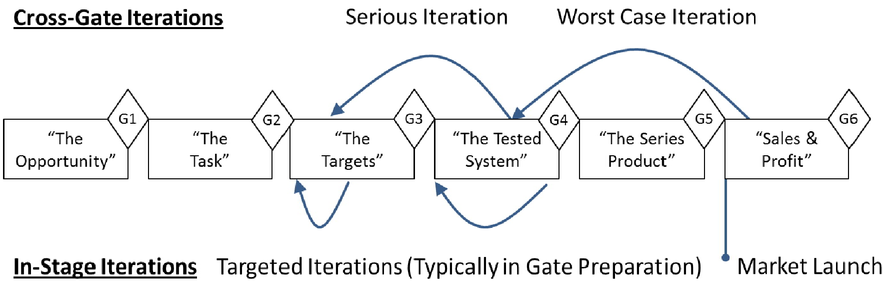

Figure 10.5. In-stage and cross-gate iterations

An experienced engineering designer is able to build up powerful mental models of the product, its application and boundary conditions. In several mental iterations he or she synthesises a design proposal, which can be validated later in the process. The designer is able to pre-think, how the design will behave for the customer in a real application under realistic boundary conditions (Matthiesen, 2011). Now it has to be validated to see if the mental model covers all relevant elements of the real application and environment. This is done, for example, by prototyping, testing and system integration under realistic boundary conditions.

One of the key deliverables at G2 is definition of the product's value proposition (VP). At G3 the value proposition has to be confirmed by customer experience with a physical prototype under application conditions (Customer Acceptance Test). Here, there are three possible outcomes:

- the VP and the design match the needs of customer (no iteration);
- the VP is verified, but the customer identifies weaknesses in the design of the product, *e.g.* the power tool is not well balanced. These are quantitative issues, which lead to a redesign (in-stage iteration);
- the VP is found to have been false and does not meet customer expectations. Thus, the VP, a key decision at G2, has to be redefined, *i.e.* the concept has to be changed. In consequence the development progress for stage 3 is obsolete and the process has to restart in stage 2 (cross-gate iteration).

10.4.2 Practical Examples of Iterations

The first example considers self-drilling screws that are used in steel construction to fasten metal sheets to metal structures. The performance of the drilling process is severely limited by the temperature of the screw's drilling tip, which increases quickly during the drilling process. A promising concept is to cool the cutting edge while drilling to increase performance. In the field of self-drilling screws this was a novel idea. Thus, there was knowledge about increasing performance by cooling cutting edges, but no specific knowledge about increasing performance by cooling the tips of self-drilling screws.

The idea was firstly realised in a power tool that pumps cooling liquid through a hole in the screw right to the drilling tip. The development team faced several problems and iterations while developing this power tool. Every iteration led to new knowledge about power tools that are able to cool self-drilling screw tips. When the power tool finally worked, it failed to achieve the predicted performance increase. Qualitatively, the function was not fulfilled. Thus, the result was a cross-gate iteration that threw development back from stage 3 to stage 1 of the TTM process model. Validation was concentrated on the power tool not on the general idea itself.

The second example concerns a fibre reinforced belt that was the most important part of a new power tool concept. This belt passed around some metal pulleys in order to transfer high dynamic forces. It was especially developed for exactly this application. The first prototype of the belt was produced and tested in a prototype power tool. After some hundred test cycles the power tool broke down. Due to the fact that the power tool was a functional prototype and thus not meant to meet lifetime requirements, the breakdown had been foreseen. The belt itself performed very well and all known requirements were fulfilled. The wear of the belt was low and projected to meet lifetime requirements. No further problems were expected.

Based on the test results gate G2 was successfully passed and management approval was given for investment in the production tool. In the following stage the team's knowledge was sufficient to design and produce the first power tool for lifetime tests. This time the belt broke after half of the estimated and required lifetime. The reason was found to be abrasion by the metal pulleys, which changed the stiffness of the belt and its pliability.

The development team did not foresee this problem, because it did not know about it and was not able to learn about it, before a lifetime test of the power tool had been carried out. Thus, the function was qualitatively fulfilled (*i.e.* the belt generally worked) but the quantitative requirements of the function were not achieved (*i.e.* the belt did not work as long as required). The failed lifetime test led to new knowledge and thus to the definition of new requirements and the redefinition of existing requirements.

In both examples knowledge was gained. The first example led to a cross-gate iteration, because qualitative fulfilment was not achieved, whereas the second example illustrates an in-stage iteration caused by failing to meet a quantitative target. Cross-gate iteration should generally be avoided. Here, one strategy is to set up validation tests very early in the development process to ensure qualitative fulfilment of a function. In the first example a substitution test might have worked in the following way: apply cooling to the metal base material and validate with regular self-drilling screws, to see if the requirements can be met.

10.5 Escalation Strategies in TTM Processes

The most critical iterations in TTM processes are iterations that affect the launch date. Several studies have shown that in the development of series products a delayed market launch is more expensive than an increase of the development costs (see Figure 10.6). Thus, in TTM processes the primary objective is to ensure the launch date. For a development team this means enormous pressure. If unexpected problems appear, they usually react by applying different escalation strategies. These strategies are not explicitly documented and they are not considered in a company's process models.

Escalation strategies contain different levels of escalation, which depend on the current problem situation and the company's prioritisation of objectives. Each level of escalation requires a "sacrifice" of cost, quality or functional objectives. Due to this the escalation strategies are also called "sacrificial strategies". In most cases the market launch will only be postponed when all levels of escalation have been exhausted. In escalation fixed project objectives are changed or even removed completely.

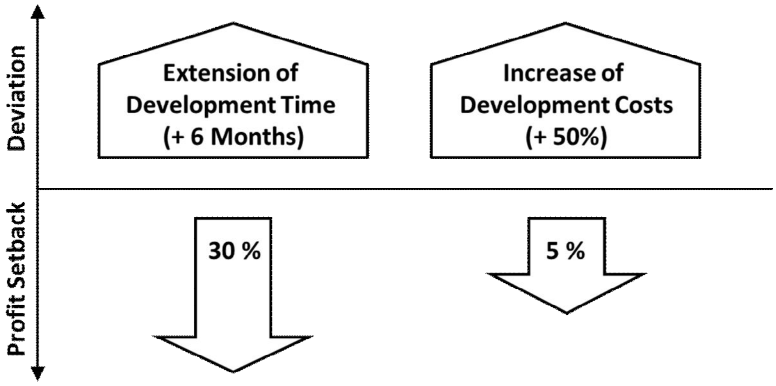
Deviation
Profit Setback
Extension of Development Time (+ 6 Months)
Increase of Development Costs (+ 50%)
30 %
5 %

Figure 10.6. Profit setback caused by an extension of development time and an increase of development cost (product life time: 5 years) according to Hall and Jackson (1992)

On the basis of the authors' experience, levels of escalation can be described as follows:

- Escalation Level 0: Initialisation of Escalation - a critical problem occurs that might lead to an unexpected iteration affecting the market launch date. Higher management's attention is attracted and the pressure on the project team is increased in order to prioritise the critical issues;

- Escalation Level 1: Sacrificing Project Resource Efforts - at this level the work load is increased by overtime work. Task forces are established that have a daily briefing at 7am and a wrap-up meeting at 8pm;
- Escalation Level 2: Sacrificing Cost of Development - at this level the project budget is increased and additional personnel resources are assigned to the project. These can be external personnel or people from other projects within the company, which are then stopped. In consequence the project plans of the other project become obsolete;
- Escalation Level 3: Sacrificing Cost of Production - here the product’s cost objectives are changed. In order to maintain the product’s quality as well as its launch date expensive improvements, *e.g.* coating, are approved;
- Escalation Level 4: Sacrificing Development Quality - in this case the functionality of the product is consciously reduced so as to maintain the market launch date. Here, quantitative fulfilment (*e.g.* ventilation slots in a previously closed housing) and qualitative fulfilment (*e.g.* entire functional features) are sacrificed;
- Escalation Level 5: Sacrificing the market launch date.

The application of escalation strategies is recognisable in different industries. The first generation of a novel product often has quality or functionality issues that are known. The first generation always has the target to bring the product to market, followed quickly by a successor with issues from the first generation fixed. For example in automotive industry, with a longer product life cycle, they handle these issues by recalls disguised as general service inspections.

10.6 Conclusions

The question of whether tight project plans and time are the most critical factors in the development process is a question for the management philosophy of the company. But it is clear that a company only earns money when the product is on the market. The strategy of ambitious project plans ensures focus and helps to set the right priorities. The competitor situation in the market is hard, and you need to take calculated risks to win the game. The guiding philosophy in a lot of companies is the quote from the Formula 1 driver Stirling Moss: “If everything is under control, you are just not driving fast enough”.

It is a fact that every product development project lasts exactly until the market launch date can no longer be postponed. The launch date is defined by the willingness to postpone the launch date - and if there is a buffer the time is always filled and at the launch date the product is ready although there are still a lot of aspects to improve. Development is not about bringing the best product to the market, it is about bringing a product to the market that is good enough. Good enough is the guiding principle. If the technical departments could decide, when a product was ready for market launch, the company would be bankrupt before the launch happened. This situation shows the daily conflict between technical development and market-oriented management. The technical view is that they want to bring the best product to the market; the management view is to bring a product to the market that is good enough.

In conclusion, the quintessence of the authors' experience in practice is that:

- iterations cannot be prevented; they are valuable and essential to gain insights and knowledge;
- cross-gate iterations should be avoided at almost any cost. One strategy to avoid them is to set up validation testing very early in the development process to validate the quantitative fulfilment of critical functions;
- in-stage iterations should be provoked to enable informed decisions by verifying qualitative function fulfilment at minimum cost;
- iterations should be provoked in the early stages of a project by validation with prototypes, system integration testing, and testing under realistic boundary conditions;
- iterations that affect previous gate decisions should be prevented by focusing on qualitative issues;
- the combination of mental pre-thinking and prototyping should be employed to minimise iterations.

In order to steer iterations effectively, a deep and detailed knowledge of the functions and dependencies of a system is an essential precondition. In further research it will be important to link process models and iterations to functional product models. Iterations have to be balanced; management's antidotes are the different escalation levels.

10.7 References

Albers A, Lohmeyer Q, Ebel B (2011) Dimensions of objectives in interdisciplinary product development projects. In: Proceedings of the 18th International Conference on Engineering Design (ICED 2011), Copenhagen, Denmark

Cooper RG (1990) Stage-gate systems: A new tool for managing new products. Business Horizons, 33(3): 44-54

Costa R, Sobek DK (2003) Iteration in engineering design: Inherent and unavoidable or product choices made? In: Proceedings of the ASME International Design Engineering Technical Conference (IDETC/CIE2003), Chicago, IL, US

Deming WE (1986) Out of the crisis. MIT Press, Cambridge, MA, US

Esquius Serra M (2010) Successful product launch in a direct sales environment from the headquarters point of view. Masters Thesis, EETAC, Castelldefels, Spain

Hall D, Jackson J (1992) Speeding up new product development. Management Accounting, 74(4): 32-36

Krehmer H, Meerkamm H, Wartzack S (2009) The product's degree of maturity as a measurement for the efficiency of design iterations. In: Proceedings of the 17th International Conference on Engineering Design (ICED'09), Stanford, CA, US

Langley G, Moen R, Nolan K, Nolan T, Norman C *et al.* (2009) The improvement guide, 2nd edn. Jossey-Bass, San Francisco, CA, US

Le HN, Wynn DC, Clarkson PJ (2010) Evaluating the positive and negative impact of iteration in engineering processes. In: Heisig P, Clarkson PJ, Vajna S (eds.) Modelling and management of engineering processes. Springer, London, UK

Matthiesen S (2011) Seven years of product development in industry - Experiences and requirements for supporting engineering design with ‘thinking tools’. In: Proceedings of the 18th International Conference on Engineering Design (ICED 2011), Copenhagen, Denmark

Miller GA, Galanter E, Pribram KA (1960) Plans and the structure of behavior. Holt, Rhinehart & Winston, New York, NY, US

Skogstad P, Leifer L (2011) A unified innovation process model for engineering designers and managers. In: Meinel C, Leifer L, Plattner H, Meinel C, Leifer L (eds.) Design thinking. Springer, Berlin, Germany

Wynn DC, Eckert CM, Clarkson PJ (2007) Modelling iteration in engineering design. In: Proceedings of the 16th International Conference on Engineering Design (ICED’07), Paris, France