



## INFORMATION FEEDBACK IN PRODUCT DEVELOPMENT: ANALYSING PRACTICAL CASES

S. Wellsandt, K.-D. Thoben and P. Klein

### Abstract

Information feedback is an important driver of iteration in development projects. Feedback emerges for various reasons and from different information sources and channels. This paper seeks to support the systematic exploration of information feedback in development. It identifies characteristics of iterative situations through literature analysis and proposes a classification schema. The identified characteristics relate to projects, processes, products and product life cycle management. Future research topics are the detailed analysis of information sources, channels and information quality.

*Keywords: information management, process modelling, product lifecycle management (PLM), quality management*

### 1. Introduction

Information feedback is an important driver of iteration in product development projects (Wynn and Eckert, 2017). In this context, *feedback* means any response that the producer or its employees receive when they present their work to a stakeholder, such as a customer, user, supplier or co-worker. Employee types that receive feedback are, for instance, engineers, sales persons, service staff, quality managers, and product managers. Responses emerge for various reasons and from different information sources. They spread via a multitude of communication channels including email, telephone hotlines, online discussion forums, social networking websites and weblogs. The employees can assess information feedback and use it to support decisions, for instance, about product design changes. Decisions may lead to rework, which is one of several iterative situations in product development.

An *iterative situation* is a set of circumstances that results in the repetition of a development activity. Iteration is a common phenomenon in Concurrent Engineering (CE). It emerges, amongst others, from the parallelization of tasks and the frequent information exchange between stakeholders. Especially the development of complex systems, such as machinery, inevitably requires that employees perform activities several times (Yassine and Braha, 2003). Iteration is necessary to make progress in a CE project, but at the same time it is also a cost factor and it extends the development time. For this reason, iterative situations should be managed efficiently. This paper focuses on feedback which includes the feedback process (information feedback) and the information item (feedback information). Their differentiation is important, because the process is a management concept, while the information item is a concrete asset used in decision-making.

This paper assumes that feedback information from heterogeneous sources affects iterative situations. An information's author (e.g. user) or its storage medium (e.g. database) are information sources from developer's perspective. Feedback information types range from informal face-to-face stakeholder responses to formalized product monitoring data stored in databases (Wellsandt and Thoben, 2016).

The following paragraphs outline two examples of iterative situations in product development. They illustrate the range of situation characteristics that relate to feedback. Knowledge about this range is important for producers, because the characteristics determine the feedback process and its information. A situation from a small-scale student project is the first example and the second one is a complex situation from the automotive industry.

The *first example* is about a one-of-a-kind development project that adapted a product's design to meet requirements of a specific production process. The developers elicited requirements, designed the product and performed experiments in the factory to evaluate the product's design. Regular meetings with the production workers accompanied the experiments. Several recurring planning, realization and improvement activities represented a major share of work in this project. Each iteration occurred on an increased level of information and the matching between the product's design and the customer requirements improved. In this example, feedback information concerns the customer's response during the project. It concerns the development results, such as, the requirements list, the concept drawings and the experiment setups. Frequent and in-situ interaction between the developers and customers are key characteristics of the information feedback in this example. The process is an active feedback process where the developers took the initiative to receive feedback information.

The *second example* is about an evasive manoeuvre for cars known as the "moose test" (Breuer, 1998). In 1997, the test drivers of a Swedish automotive magazine tested the first generation of the Mercedes A-Class. A rapid evasive manoeuvre to avoid collision with an obstacle was part of the Swedish testing procedures for years. In this particular test case, the vehicle overturned (Wüst, 1997). Product testers of other motor magazines reproduced the result leading to an extensive negative media response about Mercedes. The feedback about the A-Class caused Mercedes to stop the production and integrate an electronic stabilization control (ESC) in the car's design. The "moose test" facilitated the adoption of ESC in the automotive mass market. In this example, the feedback information was the media's response after failing the moose test. The producer's awareness for the problem increased and the management decided to address it by changing the product design of the already marketed A-Class. This change produced costs of around 100 Million Deutschmark (~50 Million Euro) per year (Wüst, 1997). The media response is an example for a passive feedback process, because the developers received the feedback information without taking the initiative.

The introduced situations represent a fraction of feedback-related iterative situations in development. Managing them in an efficient way oftentimes relies on information and communication technologies (ICT) and related support tools. The latter include email, telephone, online discussion forums, chats, weblogs, social networking services, product monitoring services, and specialized information systems (IS) used by maintenance and repair facilities. We identified two challenges that relate to the amount and heterogeneity of ICT-related tools:

- Business IS contain various measured data and unstructured text data that could be useful for developers in iterative situations. The design of IS that support the exploitation of these data requires knowledge about the data and the development tasks related to iterative situations. It is typically not readily available in companies and the engineering literature barely covers it.
- The application of feedback information in development requires that decision-makers understand its possibilities and limitations. Demonstrators from research projects typically focus on the possibilities, but they provide sparse insight about the limits. Lacking knowledge about these limits could result in costly development decisions.

The design and application of feedback-related IS could benefit from the systematic analysis of iterative situations. It is important that producers have a precise understanding of them for this reason. This way, they could communicate precisely their information need and reduce/avoid the costs of inappropriate solutions. In literature, the characterization of iterative situations is largely unaddressed. Only recently, researchers developed a first characterization for intra-project iterative situations (Wynn and Eckert, 2017). Feedback is typically mentioned in the engineering literature but not covered systematically.

This paper's *objective* is to outline the characteristics of iterative situations with a focus on feedback. It will help researchers to position their investigations and, thus, facilitate the systematic exploration and exploitation of information feedback in product development. Section 2 covers related work about iterative situations. It includes a brief overview about Wynn and Eckert's study and concepts from

feedback-related topics. Section 3 identifies the situation's characteristics. Furthermore, it develops an initial classification schema for iterative situations. Section 4 discusses this schema and the influences between the identified characteristics and feedback. Section 5 concludes this paper. The following list contains a brief glossary of important concepts and related references used in this paper.

- *Producer*. Organization that creates products through development activities.
- *Development*. Problem-solving process creating a physical product (Pahl et al., 2007).
- *Developer*. Person involved in a development process (e.g. engineer or manager).
- *Design*. One result of development (e.g. visual models or product documentation).
- *Iteration*. Repetition of a process due to access to new information (Yassine and Braha, 2003).
- *Process*. Set of actions intended to reach a goal (Westfechtel, 1999).

## 2. Related work

This section provides an overview about the research concerned with iterative situations in development. It bases on previous work of Wynn and Eckert (2017) who investigated this topic recently. They identified, amongst others, a difference between iteration within a project and iteration amongst multiple projects. The latter concerns, for instance, product changes between product generations. Wynn and Eckert did not further explain the relation between projects and iterative situations in development. For this reason, the following paragraphs motivate why this differentiation should be pursued.

*Projects* are important in development, because producers use them to bundle activities and resources to simplify their management. A project has goals, a budget, a start and end date, and project members that perform tasks. Project tasks perform in sequence, but oftentimes with considerable overlapping. In this context, project members share information with members working in the same and/or other tasks. The first example mentioned in the introduction section of this paper is a situation of this kind.

The long-term success of a product oftentimes requires that several projects work towards strategic goals. Strategic means that the goal's achievement takes several years. Some projects might need to address the shortcomings of already completed projects during this timeframe. Managers handle this situation systematically to keep the customers satisfied and to maintain the producer's competitive advantage. The second example mentioned in this paper's introduction is a situation of this kind.

In summary, iterative situations occur within a development project and between two or more projects. This paper defines the former situation type as "intra-project iteration" and the latter type as "inter-project iteration". Intra-project iteration focuses on development task performance, and inter-project iteration focuses on development-related management activities that aim for strategic goals.

### 2.1. Intra-project iteration

Wynn and Eckert performed a study about intra-project iterative situations in development (Wynn and Eckert, 2017). One of their findings was that the meaning of iteration depends on the development situation. The authors identified three functions of iteration through a literature analysis:

- **Progressive iteration.** The completion of a development project requires progress towards the planned objectives. One reason for iteration is the emergence of new information during the project. An example is the creation of prototype test results that developers use to change previously agreed design parameter values. A second reason for iteration relates to the bounded rationality of developers. For humans, it is difficult to reason about complex systems with many elements and relations. Therefore, they decompose the system into less complex parts they can work on. The developers apply the same processes iteratively to manage these parts.
- **Corrective iteration.** The emergence of new information can reveal design problems. Some of the technical problems must be corrected during the development project to avoid costly changes later, for instance during production and market entry. An example for this situation is a customer raising concerns about a product concept during a milestone meeting. The developers could address them by reworking the unsatisfying concept. In this situation, the customer's feedback information could motivate a design change.
- **Coordinative iteration.** Complex development processes should be coordinated to shorten development times and to meet budget constraints. One coordination activity is the design review

– a check between two development stages meant to confirm the adequacy of results (Unger and Eppinger, 2011). Developers perform reviews with customers to check if the design meets the identified market requirements. The information exchange between customers and developers is just one example for information feedback.

It is not our main goal to discuss Wynn and Eckert's iteration functions in this paper. Instead, we take them as an evidence that information feedback and iteration are topics with multiple facets to consider.

## 2.2. Inter-project iteration

The second iterative situation type focuses on the producer's management systems and approaches. They influence business functions, processes and in some cases development projects in particular. The following paragraphs motivate the selection of the management areas covered in this paper. Section 2.2.1 and 2.2.2 describe them in detail.

*Quality management* (QM) plans, realizes, evaluates and improves business processes to support the organization to cultivate the customer satisfaction (ISO, 2015b). It seeks to avoid the reoccurrence of product quality problems by facilitating information sharing between projects. QM is an important factor that influences iterative situations between projects.

*Product Lifecycle Management* (PLM) covers product-related activities along the entire product life cycle. One of its instruments is product information management (similar Saaksvuori and Immonen, 2008). Feedback information is a type of product information. PLM facilitates the exploitation feedback about marketed products. Subsequent product design adaption projects can use it to improve the product design in the long-run (Abramovici and Lindner, 2011). For this reason, PLM has a strong influence on inter-project iteration. A group of researchers defined the integrated view on PLM, feedback and product development as Closed-Loop PLM. This group mainly consists of the former consortium members of the large-scale research project PROMISE (PROMISE consortium, 2015). Their scope matches this paper's scope. For this reason, we adopt the notion of Closed-Loop PLM in this paper.

### 2.2.1. Quality management

Quality is the "degree to which a set of inherent characteristics of an object fulfils requirements" (ISO, 2015a). For profit-oriented organizations, the customer is one of the most significant stakeholders influencing requirements. An increase in quality improves, amongst others, the customer's satisfaction. In this context, *customer satisfaction* means the gap between the customer's expectations and the perception of the product (ISO, 2015a). Satisfied customers are loyal to the producer and contribute to higher profitability (Anderson et al., 1994). Therefore, it is important for a producer to consider the organization's view, as well as the customer's view. The former determines, for instance, the available resources and competencies to develop products, and the latter governs the customer's satisfaction (ISO, 2010). Providing information about the customer's satisfaction is one function of a quality management system.

A *quality management system* (QMS) is a management tool to improve product quality continuously. The ISO 9001:2015 standard suggests a structure for a QMS (ISO, 2015b). It is internationally acknowledged and, therefore, focused in this paper. The QMS has five elements: planning, operation and support, performance evaluation, improvement, and leadership. The following list outlines these elements with a focus on information feedback.

- *Planning* requires, amongst others, the definition of objectives for the quality management. It further concerns the preparation of actions to address risks and opportunities for the organization. Objectives should be monitored which is an argument for information feedback.
- *Operation and support* manage the quality-related aspects of the development process and the customer communication. The development process should involve customers and users. Its input information should be adequate for development purposes, complete and unambiguous. Inputs are, for instance, the functional and performance requirements, information from previous development activities, and potential consequences of failure. According to the ISO 9001, the customer communication process should obtain customer feedback. It includes "opinions, comments, and expressions of interest in a product [...]" (ISO, 2015a).

- *Performance evaluation* concerns monitoring, measurement, analysis, and evaluation of the QMS. In addition, the organization shall monitor the customers' satisfaction. It should determine the methods for obtaining, monitoring and reviewing customer-related information.
- *Improvement* refers to the actions that capacitate products to meet requirements, and address future needs and expectations. In addition, it shall correct, prevent and reduce undesired effects.
- *Leadership* promotes, for instance, improvement and ensures that the QMS remains effective.

The producer's QMS can obtain, monitor and review feedback information as described in ISO 10002 and ISO 10004. These complementary standards of ISO 9000 cover complaint handling, as well as monitoring and measuring of customer satisfaction. We do not describe them further in this paper to keep this section focused.

Performance evaluation is one of many approaches to share information between the producer and the customer. Wellsandt and Thoben, for instance, identified "Design for X", Open Innovation, product-related services and the exploitation of returned products as complementary approaches (Wellsandt and Thoben, 2016). The essence of this section is that the concept of feedback is central to an organization's quality management. It focuses on information sharing between the customer's domain (market) and the organization's processes.

### 2.2.2. Closed-Loop Product Lifecycle Management

The basis for a life cycle perspective is a *life cycle model* that specifies the focused activities. This paper uses a model consisting of development, production, usage, service, and disposal. Closed-loop PLM focuses on processes beyond the point of delivery (PoD). This point is the moment a product instance enters the customer's domain. A product instance is a specific, identifiable manufactured product. Its abstract counterpart is the product's design that serves as a blueprint for product instances. Literature refers to the customer's domain with different names, such as "field" (Magniez et al., 2009) and "middle of life" (Kiritsis, 2011; Wellsandt et al., 2016).

Many producers lose track of their product instances beyond the PoD (similar Kiritsis, 2011). This means that the producer cannot retrieve product information at will. Reasons for this are, for instance, that the customer is the information's owner and has no interest in sharing it or the product is not connected to an accessible communication network. Intermittently, the producer might receive customer complaints or other product information. Various communication channels, such as telephone hotlines, service centres and websites, convey this information. In some cases, service providers are contractually bound to deliver reports for activities, such as inspection, maintenance and repair. We identified two information sources involved in feedback processes (adapted from Wellsandt et al., 2015b):

- **Humans.** One practice to create feedback information is reporting. After-sales service staff, for example, reports on activities and the serviced product instance. In addition, customers report and converse with others about products in the Internet (Wellsandt et al., 2015a). Humans use communication channels, such as maintenance reporting software, online discussion forums and product review websites.
- **Machines.** Product embedded information devices (PEIDs) can create measured data through sensors (Kiritsis, 2011). After-sales processes, such as asset management and predictive maintenance, use PEIDs to monitor the product's performance. Embedded control systems can create log files that indicate the product's failure states in addition.

Closed-loop PLM is a concept that supports the producer to establish feedback processes for product improvement. For this reason, it is an influential management approach affecting inter-project iteration.

## 3. Findings

The following findings describe and justify characteristics of iterative situations in product development. We derived them from (1) the ISO 9001 QMS and the Closed-Loop PLM, (2) the results of Wynn and Eckert's literature survey, and (3) our experience from research projects in this domain.

Each of the following sub-sections focuses on one group of iterative situation characteristics. Grouping provides a first structure to the findings. The groups are complementary and the description of some

characteristics requires the introduction of others. This section starts with project-related characteristics, because they describe the starting conditions of development activities. Each of the following groups builds upon the characteristics of the previous ones.

### 3.1. Project-related characteristics

Wynn and Eckert ground their literature survey on the assumption that a producer schedules projects to instantiate the development process. A project has numerous characteristics, for instance:

- **Levels of information.** Understanding of problems and solutions.
- **Budget.** Defined financial and human resources.
- **Time schedule.** Start date, end date, and milestones.
- **Competencies.** Strengths and weaknesses of involved people and organizations.
- **Responsibilities.** Relations between superiors and subordinates.
- **Objectives.** Number of targets and relations between them.

This paper focuses on the information levels in development projects. Wynn et al. (2011) clarify this concept by an investigation of the influence of uncertainty during product development. They define uncertainty as everything that contributes to a lack of definition, lack of information, or lack of trust in information. It concerns aspects, such as imprecision, inconsistency, inaccuracy, indecision and instability. A project operates on higher levels of information by reducing its uncertainties.

Feedback information can reduce uncertainty and consequently increase the information level in a project. A design review with a customer, for instance, could reveal that the developers assumed a design parameter value too large. The correction of this value increases the product specification's accuracy. Activities that require this improved specification perform on a higher level of information.

### 3.2. Process-related characteristics

The product development process includes several sub-processes. It starts, for instance, with product planning and ends with the product's design and the preparation of production and operating instructions (similar Pahl et al., 2007). A project instantiates sub-processes and executes them in an organized way to achieve its objectives. Complex projects contain thousands of related sub-process instances. This complexity creates challenges that easily eclipse the influence of feedback between projects. One of these topics is change management.

The following paragraphs take a highly abstracted view on the processes in development projects. Our motivation for this is to establish a foothold for the systematic analysis and discussion of feedback in product development – it is covered further in the outlook of this paper.

This paper differentiates two *types of process organization* (derived from Wynn and Clarkson, 2017):

- **Logical organization.** Some sub-processes must or should be executed before others. Task analysis, for instance, should happen before the solution design.
- **Chronological organization.** Processes can be organized one after another or in parallel. The latter includes partial and complete concurrency over the process's execution time. This organization type depends on project-related characteristics. A tight project schedule, for instance, may require concurrent process execution.

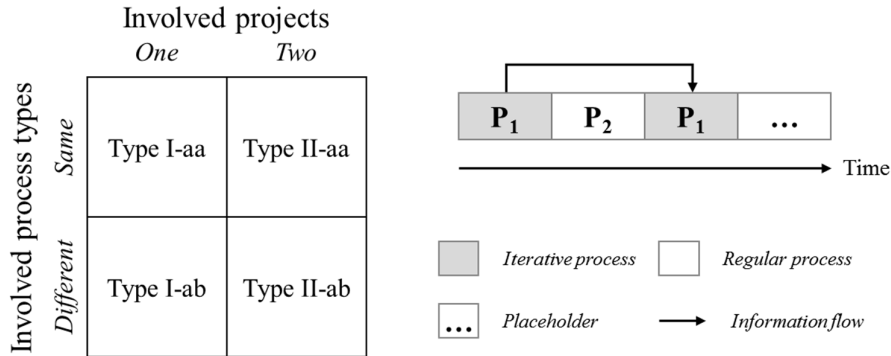
This paper focuses on the chronological organization. We used a *graphical notion* with boxes and arrows to describe process instances and information flows in iterative situations. This notion avoids the use of backwards-oriented flows, because they can be confusing and disguise the fact that even the simplest iterative situation involves two process instances. The downside of the selected graphical notion is that the description of multiple process instances becomes difficult to visualize and to comprehend.

We differentiate four iterative situation types that emerge from project count and process type variables.

- **Project count.** In its simplest form, an inter-project situation involves two projects. The project count of a situation can be one or two in the following paragraphs for this reason. We do not cover situations with more than two projects in this paper.

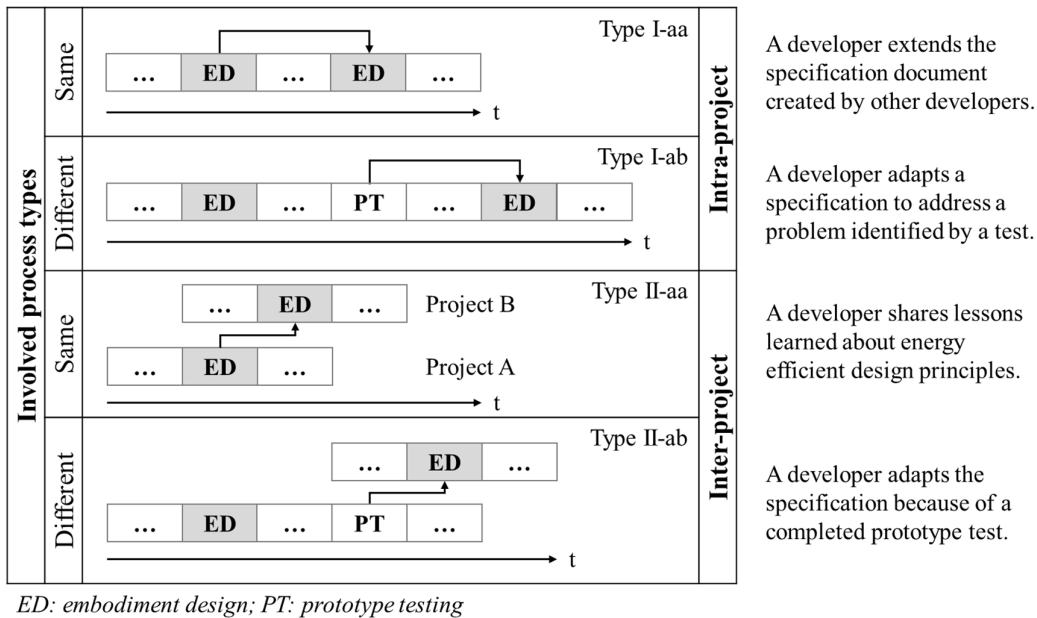
- Process types.** The process instances involved in an iterative situation can be of the same type or of different types. An iterative situation has one information flow in this paper. It can involve two process types at most for this reason.

Figure 1 summarizes the four situation types and the graphical notion used in this article. The following schema names the situation types: a Roman number represents the involved projects and two lowercase letters represent the involved process types.



**Figure 1. Iterative situation types (left) and graphical notion (right)**

The illustrated notion uses a set of elements to describe a situation. Grey boxes highlight the iterative process instance, white boxes indicate a regular process instance, and a box with three dots acts as a placeholder for other processes instances not covered in this paper. An arrow indicates the information flow between two instances. Figure 2 summarizes the four iterative situation types mentioned earlier.



**Figure 2. Types of intra- and inter-project iterative situations**

The selected process types are examples meant to make the illustration more comprehensible. In many situations, several business units and processes contribute to iterative situations. Fixing product failures, for instance, may require the collaboration of QM (perform root-cause analysis), sales (compensate customers), marketing (identify impact on customer satisfaction) and development (create technical solution to root cause).

The situation "type I-aa" contains the embodiment design (ED) as an example process. It performs two times on different levels of information. An example is the creation of a preliminary design document

that developers finalize near the end of the project. The break in ED could be necessary, because the developers lack essential information to specify design parameters earlier. This example represents a progressive iteration. Situation "type I-ab" includes ED and prototype testing (PT). In this case, the PT creates information that flows to the ED. An example is a product failure observed during a field test. Developers can use the failure information in a second ED to resolve the failure's cause. The second ED performs on different levels of information. This example represents a corrective iteration. Situation "type II-aa" contains an ED process instantiated in two different projects. Project A's ED creates information used by project B's ED. An example situation is the reuse of a best practice from a completed ED in subsequent projects. We did not classify this example according to Wynn and Eckert's approach, because it focuses on intra-project iteration. Finally, situation "type II-ab" contains one project with ED and PT, and a second project with ED. The information flows from the project A's PT to project B's ED. An example is a prototype test revealing design problems near the end of project A. The developers may be unable to address the problems identified in project A, for instance due to budget constraints. They could assess, if a subsequent project B addresses the problems by revising project A's ED results. We did not classify this example for the same reason mentioned in the previous one.

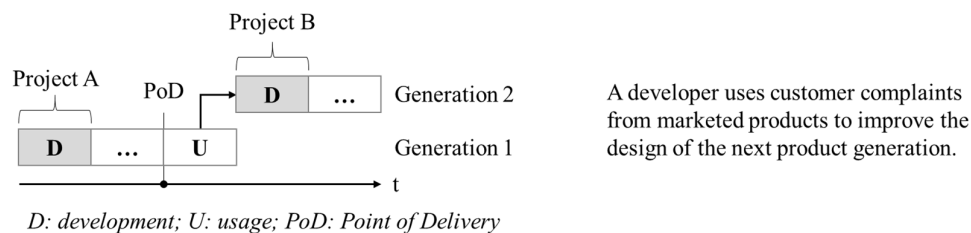
### 3.3. Quality-related characteristics

We assume that each development project focuses on one product at a time in this paper. From the QM perspective, the project's objective is to create a product design that satisfies customer expectations. The producer should evaluate its performance on the market continuously as explained in Section 2.2. This evaluation drives the scheduling of improvement projects where necessary. Performance evaluation and improvement process extend "type II" situations by the following characteristics:

- **Producer and customer perspective.** The PoD marks the transition from producer-focused processes to customer-focused processes. For the ease of simplicity, we summarize all customer-focused processes as "usage" process in the following sections. Wellsandt and Thoben (2016) provide further details about potential sub-processes of usage.
- **Product generations.** Product quality improvement implies that different product design versions exist. This paper refers to them as "product generations". Albers et al. label the transition from one generation to the next a "product generation development" (Albers et al., 2014). The product structure remains mostly the same during this process.

In this article, we assume that producers use projects to realize product generation development. This means that the transition from one generation to the next involves two projects. Eppinger describes the transfer of lessons learned between subsequent projects as "generational learning" (Eppinger, 2001). Despite this simplified notion for this complex topic, we adopt this definition as a starting point for this paper. The following sections aim to clarify its meaning.

Figure 3 illustrates a generational learning (GL) situation. It refers to a producer that performs product generation development on the basis of a marketed product. The PoD differentiates the producer-focused processes from the customer-focused ones. Usage includes the performance evaluation conducted by the producer's QMS. The illustration is even more abstract than Figure 2, because one box represents the entire development process rather than a specific sub-process. Producers typically have several products of different generations marketed at the same time. Lessons learnt can emerge from any of them. This aspect is not illustrated in Figure 3 to keep the illustration comprehensible.



**Figure 3. Generational learning situation**

A developer uses customer complaints from marketed products to improve the design of the next product generation.



The illustrated GL situation addresses the QMS objective to acquire information from the customer's domain. Developers can use this information to improve future product generations. The situation is similar to the situation "*type II-ab*", because it covers two different process types. Its scope is wider, though, because it covers the customer's domain. A situation of "*type II-aa*" can be a valid GL situation if, for instance, project A's development provides information to project B's development. We did not analyse this case further, because it does not focus the customer's perspective.

### 3.4. Product-related characteristics

We assume that each product belongs to a specific *product type*. This could be an assembly robot or a passenger car, for instance. The product type is an important characteristic of iterative situations. It describes how similar the context of projects is. Projects that share a context may share feedback information, such as lessons learnt, easier. The introduction of a product type extends "*type II*" situations by a third variable. Two capital letters reflect this variable:

- Projects focus on the **same** product type ("*type II-aa-AA*" and "*type II-ab-AA*")
- Projects focus on **different** product types ("*type II-aa-AB*" and "*type II-ab-AB*")

This paragraph introduces two example situations related to product types. They are simple but we use them to clarify the meaning of the third variable. The examples focus on the GL situation illustrated in Figure 3. An example for the situation "*type II-ab-AA*" includes developers that use performance information to improve a product of the same type. They make changes to the design specification of the previous generation for this purpose. The situation "*type II-ab-AB*" involves different product types. Developers could improve a product with performance information from a similar product type. An example is an electrified truck development that benefits from performance information of electrified passenger cars. Both product types are similar, because they use the same powertrain technology.

### 3.5. Life-cycle-related characteristics

A plain feature of the product life cycle is that it covers processes from the product's ideation to the disposal. Less intuitive are the related views of PLM on information management. In this context, we identified three aspects related to iterative situations:

- **Generalization.** A life cycle model may include all stakeholders of product-related processes. Effectively, it serves as a placeholder for "any relevant party or process". This means, that relevant iterative situations may occur in any life cycle process.
- **Information sources and channels.** Closed-loop PLM puts emphasis on information flows. We identified machines and humans as sources for product-related information. This information flows through communication channels, such as telephone or websites.
- **Tracking of product instances.** The PoD marks were producers typically lose track of their product instances. One reason for this is that the customer is responsible for the product after its delivery. Consequently, the producer has limited access to product instances.

GL is one iterative situation covered by the closed-loop PLM perspective. In contrast to the customer-focused QM perspective, it has a generalized usage process. This means that the usage process becomes a placeholder "... " process. Consequently, feedback information to product development can originate from any life cycle process beyond the PoD. An example is the recycling process that could provide information about product disassembly issues. PLM's information management aspect introduces information sources and information channels as situation characteristics. They allow us to describe iterative situations where a producer receives maintenance reports via email, or customer complaints via telephone or the Internet. These situation characteristics are difficult to specify, because there is no generally agreed list of sources or channels to our knowledge.

### 3.6. Summary

Table 1 summarizes the findings of this paper. We identified three basic characteristics and three complex ones from the previous sections. Basic characteristics describe an iterative situation according

to the number of involved projects, the process types and the product types. Each has two values providing a rough classification schema for iterative situations in product development.

**Table 1. Characteristics of iterative situations in product development**

Characteristic	Type	Relates to	Values
Project count	Basic	Project	One or two
Process types	Basic	Process	Same process or different processes
Product types	Basic	Product	Same product or different products
Information sources	Complex	PLM	No generally agreed list
Information channels	Complex	PLM	No generally agreed list
Levels of information	Complex	Project	Context-dependent

Complex characteristics are information sources, information channels and the levels of information. Their values are not generally agreed or context-dependant. We did not include the number of product generations in Table 1, because these situations result from the combination of the basic characteristics.

#### 4. Discussion

The *first part* of this section discusses the applicability of the previously proposed classification schema. We apply it to the two use cases from the introduction of this paper. Table 2 summarizes the results.

**Table 2. Characterization of iterative situation examples**

Characteristics	Example 1	Example 2
Project count	One	Two (assumed in this paper)
Process types	Same (development)	Different (usage, development)
Product types	Same (camera mounting)	Same (Mercedes A-Class)
Information sources	Human (business customers)	Human (product reviewer)
Information channels	Face to face interviews	Automotive magazine
Levels of information	Adapt existing design. Learned about task-specific requirements and constraints of the application context.	Develop new product class. Learned about the "moose test" steering manoeuvre and its consequences for the vehicle's stability.

We identified most of the basic characteristics easily. The project count for the second example was difficult to identify, because the literature did not clearly state that the producer scheduled a follow-up project. The main feedback information sources were humans. Information channels concerned direct interaction between humans via interviews and publishing in an automotive magazine. The identified level of information refers to the novelty of the development tasks and the lessons learnt.

The *second part* of this section discusses the influence between the identified situation characteristics and feedback. Its focus is on inter-project situations, such as GL. Project-related characteristics affect these situations in at least two areas:

- **Levels of information.** A project's information level influences how much and which information the developers' need to specify a satisficing product design. This level changes as the developers put news in the context of existing information. The change of the information level happens continuously, especially when feedback information is abundant and readily accessible by developers. A challenge of the dynamic level of information is that developers must assess new information regularly. They might reveal contradictions with existing information or identify errors they must take into account. Developers may also receive too much information to process. In this specific case, feedback causes an information overload which could increase uncertainty and reduce the level of information.
- **Competencies.** The developer's competencies regarding feedback could affect which information sources and channels they consider. A product manager with a background in information technology may know that websites, service reports and product-embedded sensors are valuable

information channels. The plethora of potentially applicable sources and channels could lead to new challenges for development. User-generated Internet data, for instance, is suitable to assess the customers' preferences (Tuarob and Tucker, 2016). The developer assesses if each source is credible enough to be trusted at the same time. Computer programs may support this assessment, but they might need to operate on credible algorithms to be trusted.

A limitation of the classification schema is that we assumed projects as the main instrument to realize development. In practice, several departments may collaborate in projects and/or on a continuous basis to realize satisfying products. Likewise, the process may involve different employee roles ranging from operative/technical roles to the strategic management. We did not investigate these aspects in this paper, because the resulting iterative situations might be much more complex and more difficult to analyse.

## 5. Conclusion

This paper proposed a classification schema for feedback-related iterative situations in product development. It is applicable to situations as demonstrated in the discussion. A major shortcoming is that situations are highly abstracted and many challenges of iterative situations were not covered in this paper. We identified the following aspects related to the classification that qualify for future research:

- **Sources and channels.** An iterative development situation can involve different information sources and channels. Their characteristics influence the information they create and convey. An understanding of the sources and channels may help PLM experts to identify the most relevant ones for specific development situations. It may also help information management to improve IS that provide feedback information to developers.
- **Information quality.** Developers typically expect information to be accurate and of the right age and/or precision for their work task's context. In practice, feedback information may not be perfectly fitting. Developers may still work with it, because they lack better alternatives. It is important to understand the limitations of the information in these cases. Future research could identify factors that limit the use of feedback information in development.

The latter aspect has extensive influence on the application of feedback information in development. Domains, such as information science and quality management, should be involved to improve the understanding about the quality of feedback information.

## Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements no. 636951 (Manutelligence) and no. 727982 (Lincoln).

## References

- Abramovici, M. and Lindner, A. (2011), "Providing product use knowledge for the design of improved product generations", *CIRP Annals - Manufacturing Technology*, Vol. 60 No. 1, pp. 211–214. <https://doi.org/10.1016/j.cirp.2011.03.103>
- Albers, A., Bursac, N., Urbanec, J. and Lüdcke, R. (2014), "Knowledge Management in Product Generation Development – an empirical study", *Proceedings of the 24th Symposium Design for X, Bamberg, October 1-2, 2014*, TuTech Verlag, Hamburg, pp. 1–12.
- Anderson, E.W., Fornell, C. and Lehmann, D.R. (1994), "Customer Satisfaction, Market Share, and Profitability. Findings from Sweden", *Journal of Marketing*, Vol. 58 No. 3, pp. 53-66. <https://doi.org/10.2307/1252310>
- Breuer, J.J. (1998), "Analysis of driver-vehicle-interactions in an evasive manoeuvre: Results of "moose test" studies", *Proceedings of 16th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*.
- Eppinger, S.D. (2001), "Innovation at the Speed of Information", *Harvard Business Review*, Vol. 79 No. 1, pp. 1–11.
- ISO (2010), *Quality management - Customer satisfaction - Guidelines for monitoring and measuring*, Vol. 03.120.10 No. ISO 10004:2010, Beuth Verlag GmbH, Berlin.
- ISO (2015a), *Quality management systems - Fundamentals and vocabulary*, 01.040.03; 03.120.10 No. ISO 9000:2015, Beuth Verlag GmbH, Berlin.
- ISO (2015b), *Quality management systems - Requirements*, Vol. 03.120.10 No. ISO 9001:2015, Beuth Verlag GmbH, Berlin.

- Kiritzis, D. (2011), "Closed-loop PLM for intelligent products in the era of the Internet of things", *Computer-Aided Design*, Vol. 43 No. 5, pp. 479–501. <https://doi.org/10.1016/j.cad.2010.03.002>
- Magniez, C., Brombacher, A.C. and Schouten, J. (2009), "The use of reliability-oriented field feedback information for product design improvement. A case study", *Quality and Reliability Engineering International*, Vol. 25 No. 3, pp. 355–364. <https://doi.org/10.1002/qre.973>
- Pahl, G., Beitz, W., Blessing, L., Feldhusen, J., Grote, K.-H. and Wallace, K. (2007), *Engineering Design: A Systematic Approach*, 3rd ed., Springer-Verlag London Limited, London.
- PROMISE consortium (2015), *PROMISE PLM - New Generation Lifecycle Management*. [online] Available at: <http://promise-innovation.com/promise-plm> (accessed 11.11.2017).
- Saaksvuori, A. and Immonen, A. (2008), *Product Lifecycle Management*, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Tuarob, S. and Tucker, C.S. (2016), "Automated discovery of product preferences in ubiquitous social media data A case study of automobile market", *2016 International Computer Science and Engineering Conference (ICSEC)*, Chiang Mai, Thailand, December 14-17, 2016, IEEE, pp. 1–6. <https://doi.org/10.1109/ICSEC.2016.7859912>
- Unger, D. and Eppinger, S. (2011), "Improving product development process design. A method for managing information flows, risks, and iterations", *Journal of Engineering Design*, Vol. 22 No. 10, pp. 689–699. <https://doi.org/10.1080/09544828.2010.524886>
- Wellsandt, S., Hribernik, K. and Thoben, K.-D. (2015a), "Content analysis of product usage information from embedded sensors and web 2.0 sources: A first analysis of practical examples", *2015 IEEE International Conference on Engineering, Technology and Innovation/ International Technology Management Conference (ICE/ITMC)*, Belfast, Ireland, June 22-24, 2015. <https://doi.org/10.1109/ICE.2015.7438641>
- Wellsandt, S., Hribernik, K. and Thoben, K.-D. (2015b), "Sources and Characteristics of Information about Product Use", *Procedia CIRP*, Vol. 36, pp. 242–247. <https://doi.org/10.1016/j.procir.2015.01.060>
- Wellsandt, S., Nabati, E., Wuest, T., Hribernik, K.A. and Thoben, K.D. (2016), "A survey of product lifecycle models. Towards complex products and service offers", *International Journal of Product Lifecycle Management*, Vol. 9 No. 4, pp. 353–390. <https://doi.org/10.1504/IJPLM.2016.10001874>
- Wellsandt, S. and Thoben, K.-D. (2016), "Approach to Describe Knowledge Sharing between Producer and User", *Procedia CIRP*, Vol. 50, pp. 20–25. <https://doi.org/10.1016/j.procir.2016.04.193>
- Westfechtel, B. (1999), *Models and Tools for Managing Development Processes, Lecture Notes in Computer Science, Vol. 1646*, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Wüst, C. (1997), *Tanz um die Gummihütchen*. [online] Der Spiegel. Available at: <http://www.spiegel.de/spiegel/print/d-8813322.html>
- Wynn, D.C. and Clarkson, P.J. (2017), "Process models in design and development", *Research in Engineering Design*, Vol. 29 No. 2, pp. 1–42. <https://doi.org/10.1007/s00163-017-0262-7>
- Wynn, D.C. and Eckert, C.M. (2017), "Perspectives on iteration in design and development", *Research in Engineering Design*, Vol. 28 No. 2, pp. 153–184. <https://doi.org/10.1007/s00163-016-0226-3>
- Wynn, D.C., Grebici, K. and Clarkson, P.J. (2011), "Modelling the evolution of uncertainty levels during design", *International Journal on Interactive Design and Manufacturing (IJIDeM)*, Vol. 5 No. 3, pp. 187–202. <https://doi.org/10.1007/s12008-011-0131-y>
- Yassine, A. and Braha, D. (2003), "Complex Concurrent Engineering and the Design Structure Matrix Method", *Concurrent Engineering*, Vol. 11 No. 3, pp. 165–176. <https://doi.org/10.1177/106329303034503>

Stefan Wellsandt, Scientific Researcher  
 BIBA - Bremer Institut für Produktion und Logistik GmbH, IKAP  
 Hochschulring 20, 28359 Bremen, Germany  
 Email: [wel@biba.uni-bremen.de](mailto:wel@biba.uni-bremen.de)