INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

REDUCING WASTE IN PRODUCT DEVELOPMENT BY USE OF MULTI-DOMAIN MATRIX METHODOLOGY

F. Elezi, M. Graebsch and U. Lindemann

Keywords: lean, waste, product development, complexity, multi-domain matrix, value stream mapping, design structure matrix

1. Introduction

Product time to market proves to be one of the most important factors for the success of the technology-driven companies in modern-day markets. This is particularly true for the automotive and high-tech industries. Reducing time to market is very important for the companies operating in these industries not only for establishing themselves in their respective markets, but also for increasing their market share. Furthermore, decrease of time to market impacts the learning curve, which on the other hand can lead to considerable cost reduction compared to the competitors. Therefore, it has become essential for companies to develop high quality products faster than ever before.

In most cases, the biggest contributor to time to market is the duration of the design phase. This phase contains a lot of rework and iterations due to many causes such as defective information, upstream changes, insufficient communication, and poor planning and coordination, all resulting in the extension of the development time and uncertainty [Browning, 1998]. In technology-driven industries, time to market basically represents the time spent for product development. Therefore reducing product development cycle time appears to be a plausible solution to reduce time to market. Shortening the time companies spend in developing their products has two main benefits. First it allows companies to be more flexible and respond faster to market demands. Second, the developing costs can potentially be reduced.

Lean thinking, as a way to improve processes in an efficient and effective manner, was originally developed by Toyota after the Second World War and since the end of last century is widely used in manufacturing and service industries with considerable success. It consists of principles and techniques that support the process flow by increasing value-adding tasks and by reducing what is considered as waste in the process (non-value adding tasks). The most common waste elimination tool used for this purpose is Value Stream Mapping (VSM) [McManus, 2005]. As this way of process improvement proved to be successful in manufacturing environment, researchers started applying Lean Thinking in other business areas, such as Product Development.

2. Problem

Product Development processes differ considerably from manufacturing processes, primarily in the objects that flow between tasks. Namely, in manufacturing process there is physical material flowing, while in Product Development mainly information in various formats are sent and received between tasks. Information flow, on the other hand, is much more difficult to manage. It is not clear even to date, how to identify tasks of the process where the value is added and where not [Browning, et al., 2000]. Many information quality issues may arise even between tasks (interfaces), when the

information is sent/received from one task to the other. Moreover, Product Development process flow comprises of a large number of branching, iterations and rework, which further complicates the management and improvement of flow. Consequently, the processes in Product Development are much more complex than the ones found in manufacturing.
For these complex PD processes, there is a good process mapping tool available: the Design Structure Matrix (DSM). Although DSM lacks high level of detail, it illustrates complex information flows in a simple and useful way for both qualitative and quantitative analysis [Eppinger, 2001]. However, DSM methodology is limited into the analysis of a single domain and does not take into consideration the other important domains that according to Lean Thinking influence directly the process (in particular product and organizational domains).
Multiple-Domain Matrices (MDM) represents the newest matrix-based approach for complexity management that can be used for the purpose of waste elimination in PD. The term MDM was first introduced in 2007 by Maurer and Lindemann [Maurer, Lindemann, 2007]. MDM belongs to matrix-based models, and as a complexity management tool has several advantages:

- Enhance visibility of the structure of systems and are very suitable for recognizing specific structural attributes, which gives a unique opportunity to improve the processes from the system view perspective.
- By using MDM it is not only easy to depict relations between elements of single domains, but also between elements of different domains, thus have the potential to provide a holistic view of the entire system.

This research explores the possibilities of using MDM methodology for reducing complexity and support flow by eliminating waste from the PD process.

2.1 Complexity in PD process and its relation to waste

In a survey conducted by Slack [Slack, 1999], where he inquired PD experts in industry for a single example of waste in their PD processes, the complexity appears as one of the possible wastes, although not as a relevant one (ranked 7th out of 10 wastes identified in order of occurrence). But can complexity be seen as a waste in PD?
Before answering this question, let’s have a better look into how the waste is generated in PD process. We take first a simple example of generation of inventory waste in a PD process.

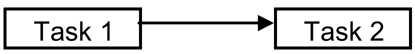

Figure 2.1 Inventory-free flow

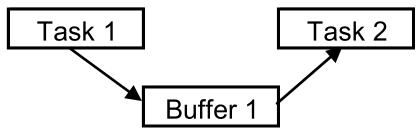

Figure 2.2 Flow of information through a buffer

As seen in the figure 4.1 the inventory-free flow is between two tasks which have only one dependency – the flow of information from task 1 to task 2. The introduction of buffer in figure 4.2 makes the system more complicated since an additional element is introduced to the system (Buffer 1 – where the information is stored) and there is an additional dependency. In the new configuration we have three elements and two dependencies, which is clearly a more complex system.
Over-processing exhibits also the same characteristics – the increase of complexity, as information goes through unnecessary tasks.

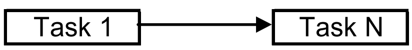

Figure 2.3 Over-processing-free flow

Figure 2.4 Over-processing in (N-1) elements

As in previous examples, it can be seen that after introduction of over-processing waste, the system becomes more complicated by (N-1) elements and (N-1) dependencies.
In the same fashion, rework (sometimes referred as defects) generates additional complexity in the process:

Figure 2.5 Rework-free flow

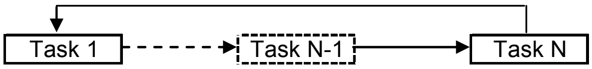

Figure 2.6 Rework in N elements

From the the examples above it can be seen that complexity does not represent waste *per se*, but rather presence of waste increases the complexity in PD processes. On the other hand, its known that complexity is inherent to PD projects. **Inherent Complexity** will be defined as **"the minimum process complexity required to deliver a customer requirement"**. In other words, inherent complexity of the PD processes emerges from requirements of the customers. These requirements are translated into the product architecture, which finally are reflected into the process. Some of major limitations that contribute to inherent complexity are given bellow:

1. The product, organizational, technological, financial and other domain limitations
2 Lack of knowledge and creativity of process-owners.

However, some of its PD process complexity is generated by the presence of waste in the system. This is additional complexity that according to Lean Thinking should be eliminated from the process as it does not add value. Hence, the **Excessive Complexity** will be defined as **"the complexity in the process that emerges mainly due to presence of waste, and does not add value in respect to customer requirements"**.

In essence, excessive complexity and waste seem to be two faces of the same coin, or in other words, excessive complexity is the systems view of the waste.

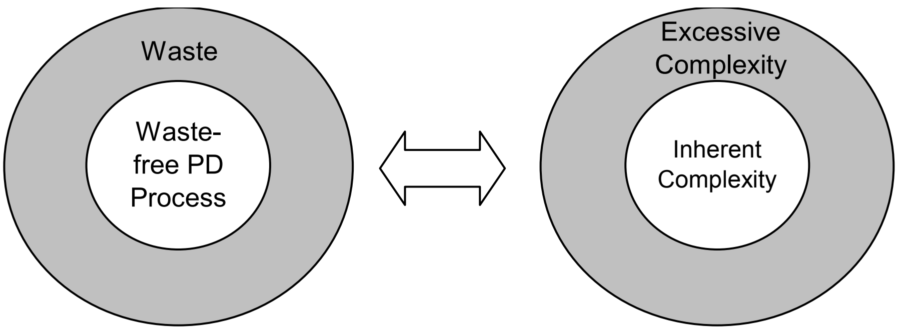

Figure 2.7 Analogy between waste and excessive complexity

This relation of waste to complexity is a good starting point to ask whether waste elimination is possible by attacking the complexity of the system, and if yes, what would the appropriate methodology be. The traditional way of waste elimination involves PD Value Stream Mapping where waste is visualized and eliminated layer after layer. This methodology involves going deeper and deeper into tasks to find waste and eliminate it. The bottom line of this traditional waste elimination methodology is to reduce the PD cycle time by making the process more efficient and effective. Overlapping activities is widely applied to achieve reduction of PD cycle time, but this usually leads to additional rework or iteration, hence increases the complexity. To control this increase of complexity, a systems analysis can prove helpful. Furthermore, knowing that excessive complexity constitutes waste in the system, preliminary analysis and excessive complexity reduction seem as a good start for waste elimination from PD processes. Although the usefulness of traditional waste reduction techniques are not put into question, the need to attack waste from a system point of view should be examined. As the traditional waste elimination is focused only on the waste generated in the process domain, a more holistic tool that examines the PD in many domains (especially product, organization and process domains) has to be utilized.

3. Methodology

In this section we propose a step-by-step methodology to analyze and improve the PD process by taking into consideration the relationship of waste and complexity discussed above. The approach should utilize systems view of all relevant domains (e.g. product, process, organization) and have the capacity to analyze the complex relationships between system elements in the same and in different domains. MDM tool provides this inter- and intra-domain insight [Lindeman et al, 2007]. The proposed methodology is designed to improve incremental innovation PD processes running in single-project environment.

Lean Product Development requires not only process improvement, but an integrated and aligned target of all other domains such as product and organizational domains. As the product domain is a representation of customer requirements, the methodology should start by analyzing first the product domain. Then, the process domain is analyzed and rearranged in a way that supports the flow of information by taking into account certain product structural constellations. At the end, the organizational domain is calculated (an MDM feature) and teams are envisaged to support the flow of information for the given process.

From what is said above, it can be concluded that the input for the proposed methodology is an MDM that contains at minimum the product and process DSM. In order to calculate the organizational DSM, the product-organization or process-organization DMM should also be provided.

This research proposes an MDM methodology to analyze and improve the PD efforts. This methodology involves 6 steps as depicted in following figure.

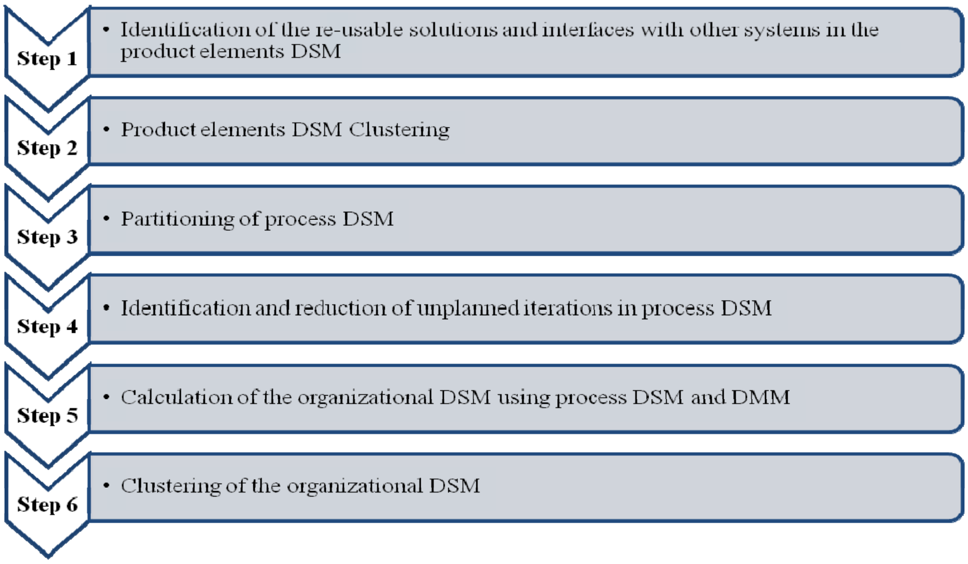

Figure 3.1 Proposed steps of the MDM methodology

Step 1: Scope management in Product Domain

The first step of the methodology aims to reduce the excessive complexity of the product architecture by identifying the already existing design solutions and the interfaces with other systems that are already defined. This step requires a meeting with the project team to list and identify the interfaces that connect the developing system with the rest of the system, as well as the parts or modules that are going to be reused in order to reduce complexity and facilitate analysis. By doing so, it is expected that a number of elements and dependencies of the product element DSM will be reduced, and the potential to elevate the quality and efficiency of the PD process is increased. The output of this step is an product elements DSM with excluded reusable parts and interfaces already developed in other systems.

Step 2: Product Elements DSM Clustering

The remaining product elements DSM is then clustered by using a clustering algorithm. This step aims to find structural patterns that could help in identifying interfaces between modules, and interfaces within the product architecture. For example, if an interface is found, then the parameters of this interconnection can be decided upon early in the PD process, so the development of the two parts can proceed in parallel and unnecessary iteration is avoided. In this step, these interfaces are only identified, while the reflection in the process itself will be considered in step 4.

Step 3: Process DSM Partitioning

Process DSMs represent a concise, visual format for analyzing information flow in PD. Process DSMs are time-based. They are a very useful process models for highlighting iterations and coupled activities, a feature that traditional PERT/CPM models do not posses. Process DSMs describe the dependencies (input-output) between activities, thus showing the dependency structure of the process based on the information requirements. By rearranging the activities a better process architecture can be achieved, such that the creation of information is at the right time and the unplanned iteration is minimized. The technique that makes this rearrangement of the DSM is called partitioning, which sometimes is referred as triangularization. It involves an algorithm that tries to move the dependency marks in the upper-triangular form by rearranging the sequence of the tasks, thus reducing the feedback loops.

In certain cases, there are some limitations in the process, such as certain phases have to be finished before the next one can start (waterfall model). The partitioning algorithm can be applied to the DSM of such a process, but after the application the tasks have to be manually corrected in order to take into account these limitations. The other way of solving this, is to divide the process DSM into sub-DSMs that correspond to the phases of the process. This way, the mixing of tasks that belong to different process phases is avoided and the partitioning algorithm can be applied directly into sub-DSMs. However, this division of process DSM decreases the visibility of the whole process.

Step 4: Identification and reduction of unplanned iterations in process DSM

Unplanned iterations represent possible iterations in the process usually between key reviews of the designs or prototypes at the end of each PD phases. For example, if a prototype fails in some functions or has certain bugs, than the information is reworked through many previous task to try to eliminate these errors. Sometimes, when these iterations happen, the project management has even to decide whether to proceed with the project or simply abandon it. Therefore, these iterations are very important to be managed, and require preliminary analysis. The identification of unplanned iterations requires hands-on knowledge of the PD process and product architecture, therefore requires involvement of the project team.

In regard of reduction of unplanned iterations, there is a great deal of flexibility. Different techniques that involve coupling and decupling can be applied. This step is highly specific to different projects and involves both inter and intra-domain analysis. Also the knowledge gained from the output of step 2 of this methodology can be applied to reduce possible unplanned iterations. This is achieved by identifying the most interconnected parts of the product to be developed and define their attributes early in the development process.

Step 5: Calculation of the organizational DSM

After the previous steps are performed, the calculation of organizational DSM can be conducted. The input of this step is the process DSM and process-organization or product-organization DMM. The output of this step is an organizational (responsibilities) DSM that depicts the interaction between developers during the PD process.

Step 6: Clustering of the organizational DSM

In this step, clustering algorithm is applied to the organizational DSM. This step follows the same integration analysis as in step 2. The aim of this step is to cluster teams based on the interaction of the developers. The organizational DSM proposes a map of how the teams should be organized in the PD project, so the information flow between them is improved. By improving the information flow, the coupled tasks and planned iterations converge faster. The output of this step yields an organizational DSM that can serve as possible solutions of how to create dynamically teams from different functions of the organization throughout the PD process.

3.1 The impact of the propsed MDM methodology on different types of waste

The proposed methodology is expected to affect waste in two ways: Direct and Indirect impact. Directly impacted wastes are those wastes that are directly reduced with the proposed analysis. The authors identified 3 directly impacted wastes that will be more elaborated in the following section.

Indirectly impacted wastes are those types of wastes that the proposed methodology affects through reducing one or some of their causes.
Following is the table that summarizes the directly and indirectly impacted waste.

Table 3.1 Direct and indirect impacted waste by the proposed methodology

Direct impact	Indirect impact
Rework	Waiting
Re-invention	Over processing
Transport and unnecesary movement	

Reduction of Rework
Rework is in many cases the biggest contributor of extended PD cycle time. Rework in PD is caused mainly by two types of iterations: planned and unplanned iterations. Both these types of iterations can be eliminated or at least reduced by applying the re-sequencing (partitioning algorithm). In some cases, simple re-sequencing does not bring satisfactory results, therefore each iteration should be analyzed separately in order to understand why it occurs.

- Planned iterations – are usually iterations between coupled tasks. These iterations are beneficial since increase the design quality and can/should not be eliminated from the process. However, steps should be taken that planned iterations are supported in such a way that they converge as fast as possible. Faster information flow, hence a faster planned iteration can be achieved by for example co-locating the teams involved in the iteration. In the proposed methodology, a possible solution can be deducted by using the clustering algorithm in the organizational DSM (step 6).
- Unplanned iterations – are the prime causes of major delays in PD cycle time, therefore they should be given a special attention. They usually occur in downstream activities that cause an upstream activity's input to change, as errors are discovered and tests are not passed. However, such iterations can occur also earlier in the PD project. The proposed methodology tackles unplanned iterations in step 3 and 4.

Reduction of Re-invention
In PD projects it is very important to re-use already existing design solutions for certain modules of the product from previous projects, as it has a high potential to increase the quality and efficiency of PD (addressed in step 1 of the proposed methodology). Unfortunately, the developers and engineers seem to neglect this possibility and have the tendency to re-invent product parts or modules from a scratch. MDM provides a good visibility of the product structure (product DSM), and clustering algorithm can further help the user to find structural constellations (modules). By identifying these constellations or even elements of the product that can be used early, an improved PD process can be deducted.
On the other hand, after applying clustering algorithm in the product DSM, the user can identify interfaces of the product and thus standardize them early, so possible iteration at the end of project is minimized.
Reduction of Transport and Unnecessary Movement
This waste occurs mainly due to inefficient transmission and unnecessary data movement. This waste has many sources, and the most important ones are the unclear data forwarding and physical distance between developers. MDM can support the elimination of this waste in two ways. Firstly, the calculation of organization (responsibilities) DSM can help a lot in visibility of the data forwarding, therefore supports information flow (step 5 of the proposed methodology). Secondly, by clustering the responsibilities matrix, a structural constellation that can serve as basis for creation PD temporary teams is obtained (step 6). Clustering algorithm groups responsible individuals (departments) in teams based on the frequency of interaction, thus facilitates information flow in PD. The proposed methodology impacts transportation and unnecessary movement also indirectly, as it reduces iteration – which is one of the root causes of transport and unnecessary movement waste.
Waiting
Kato [Kato, 2005] argues that some tasks of the process wait (stay idle) for the input information from the tasks that are engaged in the iteration loops. This effect is even larger in case of unplanned

iterations, and the successive tasks have to wait for the information until the feedback loop converges and the information is released. By reducing rework with the proposed methodology, the waiting waste is indirectly reduced.

Over-processing

Kato identified over-processing as waste caused by transportation, rework and re-invention among others. Rework, reinvention and transportation are tackled directly with the proposed methodology, thus their reduction also potentially reduces over-processing.

It has to be noted however, that the reduction of indirectly impacted waste occurs in specific cases. Namely, waiting and over processing have many other root causes, not only the ones mentioned above. In the cases when waiting and over-processing are not caused by iteration, re-invention and transportation, but rather from some other root causes, the proposed methodology is not expected to have any effect.

4. Conclusion

This research addresses two important research questions. The first research question deals with the relation between PD complexity and waste. It is found out that introduction of waste in the process elevates the complexity of the PD process. This was clearly demonstrated in several examples with several types of waste. This conclusion is the first cornerstones of the proposed MDM methodology.

The second research question addresses the possibility flow of information improvement by attacking complexity. Before this question could be answered, a better insight of the complexity in PD processes had to be formed. Hence, the complexity was categorized into 2 types: the inherent and excessive complexity. Inherent complexity is the minimum complexity required to develop a product (given the limitations) and is not a subject of this research. On the other hand**, excessive complexity is caused by the waste in the process, therefore its reduction improves the information flow in a PD process.** This conclusion represents the second cornerstone for the proposed methodology, as it gives a clear target of what the methodology aims to achieve.

MDM methodologies were not used until now in PD waste elimination. This research represents a pioneering effort in this direction. Although most of the analysis techniques used in the methodology (inter/intra domain analysis, and deduction of dependencies) are well documented in the existing literature, the use of MDM for this particular objective is new. Reduction of complexity with the proposed MDM methodology impacts waste in two ways, directly and indirectly. The direct impacted types of waste include **rework, re-invention and transport/movement** of information. Indirectly impacted types of waste include **waiting** and **over-processing.** However, the authors want to stress clearly that the methodology proposed in this research is by no means a replacement of the traditional PD Value Stream Mapping waste reduction techniques, but rather complement these techniques with a new systemic view. In respect to the proposed methodology, a list of open questions is presented below:

Usefulness of the proposed methodology in radical innovation PD project

In radical innovation PD projects, creativity is one of the main drives of the process. Developers start with one idea, find it deficient and then decide to change it. This leads to increased number of iterations. Having in mind that the proposed MDM methodology tries to reduce these iterations into minimum, the effect on the creativity is not clear and needs further research.

Methodology is designed for single-project environment

The methodology is designed to handle one PD project at a time. However, there are very few companies with a single PD project running; rather they have several PD projects running at the same time. The methodology could be expanded to take into account these situations as well.

Universality of the proposed methodology

The methodology involves MDM analysis in 3 domains. In certain cases, there might be other relevant domains that should be analyzed. For example, in the some industries there are different technologies available and companies would like to analyze which technology would fit best to their PD process. This would involve analysis in the technology domain, and clearly the methodology should be expanded. On other hand, some small companies with few personnel involved in PD, might skip the organizational analysis (step 5 and 6).

References

Browning, T.R.. et al, "Complex Systems Product Development: Adding Value by Creating Information and Reducing Risk", Proceedings of the Tenth Annual International Symposium of INCOSE, Minneapolis, 2000

Browning, T.R..,"Use of Dependency Structure Matrix for Product Development Cycle Time Reduction", Proceedings of the Fifth ISPE International Conference on Concurrent Engineering: Research and Applications, Tokyo, Japan, 1998, pp. 2

Eppinger S., "Innovation at the Speed of Information", Harvard Business Review, Cambridge, USA, 2001

Graebsch, M.,"Information and Communication in Lean Product Development", Diploma Thesis, Massachusetts Institute of Technology and Technical University of Munich, 2005

Kato, J.,"Development of a Process for Continuous Creation of Lean Value in Product Development Organizations", Master Thesis, MIT, Cambridge, USA, 2005

Lindemann U., Maurer M., Braun T.,"Structural Complexity Management – An Approach for the Field of Product Design", Springer-Verlag, Berlin – Heidelberg, 2009

Maurer M., Lindemann U."Facing Multi-Domain Complexity in Product Development", Cidad Working Papers Series 3, 2007

McManus, H.,"Product Development Value Stream Mapping (PDVSM) Manual", Cambridge: MIT Lean Aerospace Initiative, 2005

Slack, Robert A, "Application of Lean Principles to the Military Aerospace Product Development Process", Masters thesis in Engineering and Management, Massachusetts Institute of Technology, USA, 1998

Dipl. Ing., MSc Fatos Elezi
Scientific Assistant
Institute for Product Development, Technische Universität München
Boltzmannstr. 15, D - 85748 Garching, Germany
Telephone: +49.89.289.151-32
Telefax: +49.89.289.151-44
Email: fatos.elezi@pe.mw.tum.de
URL: http://www.pe.mw.tum.de/