ABSTRACT

Product development is multidisciplinary activity which is dynamic, creative, uncertain, highly iterative and strongly reliant on communication, making it particularly complex. Research on the topic shows that having a well-defined development framework, whether it be process-based or information-based, is a critical success factor for new products. Research also shows that some sort of product development structure is already being used in large manufacturing firms, particularly in the aerospace and automotive industries.

This work proposes a methodology for structuring the development process of small engineering firms or teams which relies on modelling to describe the processes and support their improvement. It was applied and continuously revised in a practical environment consisting of a team of 30 engineering students designing and building a race car.

Results show that implementing the methodology is time-consuming, particularly for small, understaffed teams, but that useful insights such as new organizational structures, new process approaches and better knowledge of influence of external factors can be drawn from its implementation.

Keywords: Product Development; Process Modelling; Design Structure Matrix; Applied Signposting

1 Introduction

The iterating nature of complex design processes makes it necessary for project managers to have a comprehensive overview of the whole system in order to enhance the advantages of iteration (convergence towards better solutions) without creating rework that delays the launch of the new product to market.

The rising expectations from customers, which request individual solutions tailored to their problems, the increasing competition due to globalization and the international build-up of scientific and technological skills (Bullinger & Warschat, 1996) make product development an increasingly competitive and demanding activity for companies. Three key targets were defined then, which are still valid today, for gaining the competitive edge in product development processes:

- Decrease time to market to gain market share
- Improve overall quality to meet the demands of customers
- Decrease product development cost to decrease the pay-off period

Delivering a better product, in less time, and with less cost may seem obvious, but these insights from more than 20 years ago still hold. Reaching these targets, according to Bullinger & Warschat, required the parallelization of tasks, the integration of departments and personnel via matrices and the standardization of the development process.

A survey on various aerospace firms and agencies (Osborn, 2009) shows that concurrent engineering – the first requirement stated above - is widely implemented in the aerospace industry. The integration of personnel via matrices is also widely implemented, with research from as early as 1980 (Kolodny, 1980) showing the correlation between stronger cross-functional integration in project teams and higher rates of product innovation. The main subject of this article is the structuring of the product development process, with a strong emphasis on the role of process modelling. The main goal for this article is the development of a methodology to support development process standardization or structuring in teams with unstructured processes.

2 Review of development process structuring approaches

Successful development projects which lead to the profitable sale of a product perform on the dimensions (Ulrich & Eppinger, 2011) of Product Quality, Product Cost, Development Time, Development Cost and Development Capability. These dimensions
are critical for the profitability of a product development effort, but other criteria can be important in some cases to different stakeholders.

In a benchmarking study comprising 135 manufacturing firms (Cooper & Kleinschmidt, Benchmarking the Firm's Critical Success Factors in New Product Development, 1995), some critical success factors were determined. The most critical factor is the existence of a product development structure. A few challenges arise due to basic characteristics of new product development, such as its inherent creativity, multidisciplinarity, uncertainty and risk (Fricke, et al., 1998). Some of these characteristics are not necessarily prejudicial to the development process. Iteration, for instance, has been shown to be highly beneficial to the overall outcome of the process in examples like the design of aircraft engine components (Jarrett, Dawes, & Clarkson, 2006), where evolutionary development (product development based on improving previous models) trendlines suggested that product requirements were impossible to meet.

2.1 Prescriptive approaches to structuring development processes

Prescriptive approaches are suggestions of what the process should be based on experience and best practices. Examples of prescriptive approaches include the Product Development Institute’s Stage-Gate (Cooper, Stage-Gate Systems: A New Tool for Managing New Products, 1990), “Product Design and Development” (Ulrich & Eppinger, 2011) and “Engineering Design: a Systematic Approach” (Pahl, Beitz, & Wallace, 1988). These examples are different variations of phase-gate approaches, which define a generic timeline for product development.

An article by McKinsey (Holman, Kaas, & Keeling, 2003) argues that phase-gate development processes brought dramatical efficiency increases all throughout the 90s, but that improvements were plateauing in the turn of the century. They suggest that a new approach to product development is needed which switches from a fixed, process-based to a dynamic, information-based framework. Particularly for software development, alternative approaches exist which are not linearly organized into stages, like Scrum, with good results reported (Mann & Maurer, 2005). Most research has focused on the implementation of Scrum to software development projects, but some exceptions exist that test its validity for mechanical engineering applications (Reynisdóttir, 2013).

Another approach which does not structure the development process in a sequential way is Dynamic Product Development (DPD) (Ottosson, 2004). This approach argues that long term planning and budgeting based on the assessment of market needs should not be required before starting the development process, because for any product beyond a certain complexity there is too much uncertainty behind this assessment. Particularly for the generation of product ideas.

The main problem with these approaches is that they give little insight into the tasks that must be performed in the process, lacking low-level task description for new and inexperienced product development teams.

2.2 Descriptive approaches to structuring development processes

Descriptive approaches attempt to understand the process as it is in order to standardize and improve it. Some of the most common tools which are used to construct descriptive models of design processes are the Critical Path Method (CPM) and the Process Evaluation and Review Technique (PERT). These tools allow the development process to be represented as a sequence of tasks and their dependencies. Both can be used to determine the sequence of tasks in which a delay in any task will represent a delay in the overall project completion (known as the critical path).

In complex development project with hundreds or thousands of tasks, network diagram tools such as CPM or PERT can become difficult to display. For better visualization, both these task network tools can be represented by Gantt charts. Despite being the most commonly used tool for scheduling projects, Gantt charts are not ideal to represent iteration, because task dependency can only be represented as precedence.

One of the most common tools to represent task dependency and is the Design Structure Matrix (DSM) (Eppinger & Browning, 2012), which is a visual representation of a system with N elements in a N x N matrix form. The DSM enables applying analysis algorithms such that the components/tasks/system elements can be reordered. The most common analysis algorithms are partitioning, clustering (Figure 2-1) and banding.

Other matrix representations besides the DSM exist. The Domain Mapping Matrix (DMM) was developed as an extension of the DSM concept, and is a rectangular rather than a square matrix, allowing relationships between different domains. Five domains are considered to exist (Danilovic & Browning, 2007): Goals, Product, Process, Organization and Tools. The same analysis algorithms discussed for the DSM can be used.
A third matrix mapping approach can be used which is called Multi Domain Matrix (MDM), as seen on Figure 2-2. This method combines the two previously discussed approaches and allows multiple DSM and DMM to be combined in a single matrix (Eppinger & Browning, 2012). The same analysis algorithms can be used to organize the development process according to the different domains.

3 Methodology

The main goal of this work is the development of a methodology, based on process modelling, comprised of a series of steps aimed at structuring the design process of product development teams, particularly those with a high degree of iteration. It is argued that this facilitates understanding the design process and enables the team to improve it, ultimately resulting in better designed products, with reduced development cost and time-to-market.
3.1 Software Tools used for modelling and simulation

The main software tool used for modelling and simulating the design process for this methodology is the “Cambridge Advanced Modeller”. Modelling a system and simulating its outcome can be done using 2 different platforms: Design Structure Matrix (DSM) or Applied Signposting Model (ASM). Because the former has already been described in detail, only the former is further explained.

The Applied Signposting Model (Wynn, 2007) in CAM allows for the accurate definition of each task in the design process, as well as its interactions with other tasks. The graphic presentation is a flowchart of tasks and deliverables, as seen on Figure 3-1.

![Figure 3-1 - Simple process that modifies Deliverable 1, using Simple, Compound and Iterative Tasks](image)

Regarding simulation, the Applied Signposting Model uses the Monte Carlo Method (Metropolis & Ulam, 1949) to plot the probability distribution function of the overall process duration. By default, results are displayed in the form of a histogram and a cumulative distribution function.

3.2 Modelling parameters.

The first thing to be modelled is the duration of a task. Every task is subject to uncertainty, so duration always should be modelled using a probability distribution function. The Triangular PDF is the suggested function. Tasks which are repeated due to iteration should be completed faster as more iterations are performed, modelling the gained experience. In a certain iteration, each duration should be (using the expected value as an example):

\[ T_{\text{Expected}}(N) = \left( \frac{T_{\text{Expected initial}} - T_{\text{Expected shortest}}}{N^B} \right) + T_{\text{Expected shortest}} \]

Where \( N \) is the number of iterations, \( B \) is the learning factor, and \( T_{\text{Expected initial}} \) is the duration of the task the first time it is attempted and \( T_{\text{Expected shortest}} \) is the minimum possible duration of a task after it is completely mastered.

A similar thought process to that of task duration is applied in estimating the likelihood of iteration. The difference is that varying the likelihood of iteration according to number of iterations is done using the `rand()` function, which outputs a random number between 0 and 1. Iteration occurs if:

\[ \text{rand}() < \frac{\text{ProbMax} - \text{ProbMin}}{N^B} + \text{ProbMin} \]

Where \( \text{ProbMax} \) is the likelihood of having to repeat the task after its first attempt, \( \text{ProbMin} \) is the likelihood of having to repeat the task after it has been attempted an infinite amount of times, \( N \) is the number of iterations and \( B \) is the learning factor. To model an iterative task, one requires 10 parameters (the last 3 are not used for simple tasks):

- Minimum Duration (For initial attempt)
- Expected Duration (For initial attempt)
- Maximum Duration (For initial attempt)
• Minimum Duration (Asymptote)
• Expected Duration (Asymptote)
• Maximum Duration (Asymptote)
• Task Duration Learning Factor (which is assumed constant for all durations)
• Maximum Likelihood of Iteration (For initial attempt)
• Minimum Likelihood of Iteration (Asymptote)
• Iteration Likelihood Learning Factor

The final parameter is simple: how many resources from a certain resource group are required to perform the task. Any number of groups and individual resources can be defined. The default availability of a certain resource is defined and is a global variable. Each task then is assigned one or more of the created resources.

3.3 Methodology Steps

The methodology is comprised of a sequence of steps which can be divided into four phases. The “Understanding Interactions” phase is comprised of the following steps:

1. List all components/functions
2. List interactions between components/functions, based on interviews
3. Build a Design Structure Matrix representing component interactions
4. Run clustering algorithms the DSM.
5. Discuss results with the designers and adjust the component list
6. Run the same procedure until the designers agree with the clustering.

After determining the most logical division of the overall system, one must list, characterize and sequence the tasks that lead to the design of each individual sub-system. This is done through an interview with the person or team responsible for the design. The proposed steps for the “Process characterization” phase are:

1. Draw the process flowchart in some medium which allows for quick corrections
2. Ask relevant questions to validate the flowchart
3. Define a set of parameters which is relevant to characterize duration and likelihood of iteration; assign values to these parameters to represent each task, as well as resource usage
4. After the agreement on the task sequence, iterations and characteristics, translate the drawings into the software tool

In the third phase, the model is built and its validity is assessed. Due to the probabilistic nature of the model, this reassessment itself is very subjective unless the team has a good historical records of similar task durations. The proposed steps for this phase are:

1. “Translate” the process into the computational model according to the interviews
2. Perform simulation on each subsystem process model individually and compare with designers’ expectations. Adjust according to feedback.
3. Repeat the two previous steps for the whole system.
4. Add the different subsystems continuously adjusting values
5. Perform simulation on the whole model and compare to historic data and designers’ intuition.

After performing these steps, the computational model should accurately represent the development process, and the simulation results should be consistent with actual practice.

The final phase of this methodology is to model changes and evaluate their effect on the outcomes of simulation. This can involve changing global variables, changing the task network itself or changing individual task parameters.

4 Case Study – Applying modelling approach

The engineering design process chosen to implement the methodology discussed in Chapter 3 was the design of a Formula Student car, specifically the design of its aerodynamic package.

A Formula Student car is a complex product which is developed by a team of students from many engineering backgrounds. The team from Instituto Superior Técnico, FST Lisboa, has been in existence since 2001 and has, since 2010, built 4 fully electric
cars. This is a suitable environment for this case study, because it shares some of the characteristics of complex product development processes in the aerospace or automotive industries.

During the time of this case study, several people were interviewed with different goals in mind: every team member was asked which sub-systems impacted each of their departments, and vice-versa; the Aerodynamics Department head was interviewed for department specific design process tasks.

4.1 Subdividing the system

First, the complete product architecture was determined. 8 sub-systems were already identified by the team. The DSM was filled with all the major components which were still not attributed to a sub-system (Figure 4-1). 7 additional subsystems were identified. Of the 15 car sub-systems, the Aerodynamic Devices sub-system was chosen for this case study.

![Figure 4-1 - System identifying DSM](image)

4.2 Building the model

The design process for the car’s aerodynamic devices depends on fluid dynamics analysis and on structural analysis, for both machined parts and carbon fibre parts. Also, in particular for the carbon fibre parts, tooling has to be designed before parts can be made.

In 2016/2017, the Aerodynamics Department decided to work on three features: front, rear and sidepod wings. In the foreseeable future, other features that can be worked on include the undertray, which is placed beneath the car in order to accelerate flow and create downforce. A low-resolution image of the complete model can be found in Figure 4-2.
4.3 Baseline simulation results and validation

The baseline process is, as far as could be understood, the current practice for the development of the Aerodynamic Devices subsystem. It is reminded that some of the branches are optional, namely the “Sidepods” and “Undertray” development processes, reflecting the design practice in the past and in the foreseeable future. The baseline simulation is only concerned with the 2017 design process, which features only the “Sidepods” optional branch. This makes it much easier to compare to actual practice and validate the model. Simulation results for the baseline process were the following:

- Most likely duration is 20 weeks and there is 23% probability that the project has a duration lower than 20 weeks
- Worst case duration is 46 weeks and there is 100% probability that the project has a duration lower than 46 weeks
- Best case duration is 12 weeks and there is 2% probability that the project has a duration lower than 20 weeks
- There is 80% probability that the project has a duration lower than 31 weeks

The real duration of the Aerodynamic Devices design process, which included rear wing, front wing and sidepods, was roughly 16 weeks, which is less than the most likely duration expected based on the simulation. This indicates that the design did not require intensive iteration and re-work and therefore was close to a best-case scenario. Discussing these results, it became clear that the actual design process “cut some corners”, sacrificing quality for time management. Therefore, the real process, which was understood as far from ideal in terms of iteration and re-work, exceeded expectations in comparison to the baseline process simulation results. Although this analysis makes accurately validating the model very difficult, the overall perception of the people involved is that, if the process was more accurately followed, the total duration would be coherent with the simulation results. This is not ideal as scientific validation, but it was considered sufficient to validate the model and carrying on with process change analysis.

4.4 Simulating changes to the model

With the baseline process defined and some results drawn from simulation, the standard design process is defined. It is now possible to simulate changes to this process in order to understand their impact. Many examples were tested, but only 2 are presented. The baseline process model was designed with two global variables in mind, which control the existence of the “Sidepods” and “Undertray” processes. The baseline simulation results consider the existence of the “Sidepods” process, but
not the “Undertray” process. What would happen if both processes were considered, for the same resources? The results can be seen on Table 4-1, and suggest that adding a new feature with the given resource constraints would delay development.

Table 4-1 - Effects of adding the “Undertray” process

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>C. Frequency</th>
<th>Sidepods + Undertray</th>
<th>C. Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Likely Duration</td>
<td>20 weeks</td>
<td>23%</td>
<td>26 weeks</td>
<td>35%</td>
</tr>
<tr>
<td>Best Case Duration</td>
<td>12 weeks</td>
<td>2%</td>
<td>18 weeks</td>
<td>8%</td>
</tr>
<tr>
<td>Worst Case Duration</td>
<td>46 weeks</td>
<td>100%</td>
<td>54 weeks</td>
<td>100%</td>
</tr>
<tr>
<td>80% C. Frequency Duration</td>
<td>31 weeks</td>
<td>80%</td>
<td>37 weeks</td>
<td>80%</td>
</tr>
</tbody>
</table>

Another interesting insight to gain would be to know how many more aerodynamic analysis resources can be added to the baseline process (just wings + undertray) before it stops making a difference. Table 4-2 shows that adding any more aerodynamic analysis resources only makes a slight difference in worse scenarios. Accounting for the increased complexity of managing more people, one concludes that no more than 3 aerodynamic analysis resources should be part of the team.

Table 4-2 - Effects of adding further aerodynamic analysis resources

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Baseline +1</th>
<th>Baseline + 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Likely Duration</td>
<td>20 weeks</td>
<td>15 weeks</td>
<td>15 weeks</td>
</tr>
<tr>
<td>Best Case Duration</td>
<td>12 weeks</td>
<td>10 weeks</td>
<td>10 weeks</td>
</tr>
<tr>
<td>Worst Case Duration</td>
<td>46 weeks</td>
<td>41 weeks</td>
<td>39 weeks</td>
</tr>
<tr>
<td>80% C. Freq. Duration</td>
<td>31 weeks</td>
<td>25 weeks</td>
<td>24 weeks</td>
</tr>
</tbody>
</table>

4.5 Direct impact of this case study

The proposed methodology and its practical application had a direct impact on the Formula Student team and its design process, though limited in terms of process standardization, which is the main goal of the methodology. Some of the impacts were adjustments to the team structure, including the creating a new department following the subsystem analysis, and insights regarding the design process as some duration values were reassessed.

The most relevant impact, which would be for the team to follow a structured design process according to what was modelled, did not occur as of yet. The main reason for this may be that the overall model was only concluded after the 2017/2018 team started its design process, and it was well over halfway through when simulation results were drawn. Also, due to time constraints, the methodology was only discussed with the previous department head, not with the current one.

5 Conclusions and future work

In the introductory notes to this document, three challenges to the improvement of product development processes were stated:

- Parallelization of sub processes and tasks to eliminate delays
- Integration of departments and persons via matrices or better dedicated cross-functional team work
- Standardization of the product development process to minimize confusion and misunderstanding.

The standardization of the product development process was chosen as the main focus of this work, particularly relying on process modelling and simulation to achieve it. However, and largely by accident, the proposed methodology addresses all three challenges. The practical application of the methodology to the case study showed that:

- The impact of parallelization can be estimated after the baseline process is described and modelled, as can the resource requirements to achieve this parallelization.
- Standardizing the development process is only possible when the cross-functional relationships between product components and the people designing them are known. As a direct consequence of attempting to model the process and these relationships, a better understanding of the requirements for cross-functional team integration is gained, making it easier to implement matrix or projectized organizational structures.
- After the methodology is implemented, not only is the standard development process defined, but also alternatives can be considered and simulated. This means that there is a clear, visual definition of the standard process, making it much easier to communicate, and that there is data in the form of simulation results showing why alternative processes are not chosen as standard.
### 5.1 Challenges to implementing the methodology

Although the implementation of the methodology is considered to improve results for product development teams, it is also clear that there are some challenges to its implementation. These became apparent during the case study, and were particularly relevant due to the author’s role as project manager for the team. Some of these challenges include the difficulty in estimating and validating parameters without accurate historical data, the time required for building the model and performing simulation, and the opposition to change when the current “process” appears to be working.

Addressing these challenges is the responsibility of management, which should ensure that proper documentation of the existing process is being produced, that the project is adequately staffed for achieving group/company goals, and that change can only be rejected because of correct reasoning and not because of inertia.

### 5.2 Future work

During the time that this work was developed, a working approach to process standardization was defined and applied in a real project environment which had enormous influence on the final step-by-step methodology. Time constraints and the nature of this work itself did not allow for further application of the methodology to other examples, which would without a doubt improve it. The first action the author would suggest to anyone willing to pick up from this work is to test the methodology in other product development environments, which will surely bring up different challenges to the ones met in this case study.

Regarding the team in which the case study was performed, the application of the suggested changes and the documentation of their impact could prove useful, and if this is the case, the extension of the model to gradually include more subsystems would allow the project manager to start viewing the design process from a more high-level perspective. This high-level perspective should be important to define long term strategy – another interesting research topic.

Besides further testing and validation of the methodology, a better understanding of the CAM software is almost certain to bring improvements in the application of the methodology.

### 6 Works Cited


