Complexity Management in Engineer-To-Order Industry: A Design-Time Estimation Model for Engineering Processes

Christian Victor Brabrand[1](✉), Sara Shafiee[2], and Lars Hvam[1]

[1] Department of Management Engineering, Technical University of Denmark, Kongens Lyngby, Denmark
christian@brabrand.com

[2] Department of Mechanical Engineering, Technical University of Denmark, Kongens Lyngby, Denmark

Abstract. The engineer-to-order (ETO) industry's business environment constantly changes, which results in challenges related to project management, on-time delivery, quality, and market competition. Companies face pressure to optimize production while demand for personalized products, and accordingly the complexity level increases. To address these challenges, companies require to identify the most important complexity drivers to improve planning, get a better overview of the resource allocation, and improve internal processes. This study proposes a design-time estimation model based on the most important complexity drivers: 1) Functional requirement, 2) Number of technologies, 3) Level of connectivity, 4) Regulation and standards. This study presents key complexity drivers for assessing the expected hours to design a product in an ETO industry. Complexity drivers are explored qualitatively and quantitatively from (i) literature review; (ii) internal regular meetings and; (iii) data analysis. The gathered complexity drivers are weighted and combined in order to develop the mathematical design-time model. Finally, an IT-tool is prototyped to test the mathematical model at the case company. The application of the developed IT-tool is also tested at the case company to prove the usability.

Keywords: Design-time estimation model · Complexity management · Engineering design · Configurator · Optimization

1 Introduction

Designing the engineering processes is crucial in manufacturing companies. The design engineering tasks include many different perspectives as process modeling, design process, and product development [1]. The customer demands a more individualized product, which means that the manufacturer might need to increase its variety to keep the customer satisfied and retain/gain a competitive advantage in the market [2]. The challenges to be handled from the increasing variety are not always simple to identify and solve [3]. The design process can be rather complex, affecting the distribution, sales, and

A.-L. Andersen et al. (Eds.): CARV 2021/MCPC 2021, LNME, pp. 636–644, 2022.
https://doi.org/10.1007/978-3-030-90700-6_72

other general value chain processes. However, the increasing variety in products does not always create value or increase consumer quality [4].

Several methods propose different approaches to control complexity in today's industry. Substitution and product standardization are commonly known approaches [5]. Other methods are directed towards single products and harmonizing production and design to reduce setup costs [6] or dismantle the manufacturing system into subsystems, called modularization [6]. Using a configurator has also shown to be beneficial to standardize the process and reduce internal complexity [7–9]. Furthermore, linear regression proved to define complexity [10].

This study investigates how a complexity management approach can help estimate design-time for an ETO company. The changes in design-time are results based on market demands. Moreover, we quantify the complexity drivers to support the research with reliable data. The design-time estimation model is based on multilinear regression (MLR). The paper demonstrates a configurator's development, based on the mathematical model and the quantified complexity drivers, to help the designers estimate the required time for a project through a practical IT solution.

This paper extends the work on a design-time estimation model from 2014, which got published in 2019, where it showed how linear regression could estimate design-time [11]. Based on high-quality data from the case company, the new approach using MLR can better estimate the required design-time.

2 Research Method

Conducting a literature study to invistigate the engineering design phase's complexity drivers, the four groups of complexity in this case company [12] are identified. The complexity driver groups are: (1) engine, (2) product, (3) process, and (4) organization. The best available data were identified on the product level, referred to as sub-functions (SF), which we utilized to prove this study's concept. The first criteria were the size of the datasets to ensure the strength and depth of the analysis. The second criteria were the selection of one department to ensure consistency in the data. We evaluate our model and configurator at the case company. This type of empirical inquiry investigates a contemporary phenomenon within its real-life context [12]. Case study research enables profound observation of the phenomenon under investigation, and for a given set of available resources, fewer cases allow for more profound observation [13]. The case company is one of the leading suppliers of turbo machines specialized in marine engines. The selected case study method ensured accurate representation and enabled triangulation of the findings between various sources, thereby improving validity.

2.1 Data Collection

The enterprise resource planning (ERP) system contained the desirable data from previous design projects. Grouping the data in sub-functions (SFs) and using Power BI allowed us to visualize and identify the most feasible department and SFs based on data availability for the complexity drivers and the data consistency for the projects.

The complexity drivers were identified through literature and discussed in interviews to align them with the case company's experts' opinions. The interview process follows a systematic approach with six questions that proved to improve IT solutions [14]. All five interviewees are experts from the case department with 5–20 years of experience.

As mentioned, this case study followed up on previous research at this case company in 2014, where a paper was published in 2019 [11]. The previous study resulted in being cost-beneficial for the case company even with the limited amount of data. To follow-up on the first research, communications with the case company were initiated, where the case company agreed on delivering new data to update the previous research and tool. A design-time estimation tool is developed based on the complexity drivers to manage the complexity to improve the engineering design departments' performance.

3 Method and Configurator Demonstration at the Case Company

We know that developing a value-adding configurator requires high quality data from the previous research at the case company. Hence, we conduct this research, and the details will be discussed in the sub-sections below.

3.1 Identified Complexity Drivers

Five relevant complexity drivers for the case company got identified. Literature research provides four of them, and the interviews resulted in fifth one. Complexity drivers vary in different settings and departments and depend on the product types, working style, culture, and strategy. The complexity drivers are listed below.

(1) *Functional requirement* - The number of functional requirements demonstrates the number of functions one module can fulfill. Functional requirements can be weighted using a functional decomposition metric [15] and significantly influences the ETO industry [10].
(2) *Number of technologies* - The number of highly complex modules in a product. The number of technologies could describe how many fuel types an engine runs on. If main modules change, it can create a significant influence on the design-time [10].
(3) *Level of connectivity* - The connectivity level is the interdependencies between the modules in a product. The connectivity can have an influence on the design-time for a product [16]. The design of one SF proved to affect other SFs design, meaning small changes in one variant will lead to changes in other variants in that product.
(4) *Regulation and standards* - The regulations and standards from the ETO company's environment affect the design-time [10]. These regulations depend on the product category.
(5) *Depth of change* - Depth of change has a considerable influence on the design-time. The first four complexity drivers, just described, have been shown to influence the product's depth of change. The change level will depend on the customer requirements and is considered a dynamic factor.

3.2 Identifying the Complexity Drivers

Having found the complexity drivers for design estimation, we investigated the complexity drivers in the department. The experts responsible for the data were interviewed. The first round of interviews was to discuss the identified SFs, understand the design process, and discuss literature's complexity drivers to identify relevant complexity drivers. Based on first interview round, seven complexity drivers and their values showed to influence the product complexity: 1) Registered hours (number of registered hours on a SF), 2) Stroke (G and S), 3) Fuel (fuel oil, methane gas, ethane gas, liquefied methanol, and petroleum gas), 4) Number of cylinders (5–11 cylinders), 5) Technology (EGR and SGR – gas treatment system), 6) Mark (generation of engine – 8.5, 9.5, and 10.5), 7) and Depth of Change (percentage change 0–100%). Following the first round of interviews, follow-up e-mails were sent to the interviewees to gain data on the depth of change. Data for the six other complexity drivers were identified on a SF level from the Power BI linkage to the ERP system.

Before the second round of interviews with the experts, the configurator was developed based on the mathematical model. From the second interview, minor changes were made to customize it to the needs of that department. However, no changes in the identified complexity drivers were deemed necessary.

3.3 Design-Estimation Time Model

The identification of complexity drivers for the design-estimation model originates from literature and interviews with experts at the case company. The model includes the main complexity drivers, quantification of the complexity in the design process, the parts most affected by the complexity, and a design-time estimation tool at the case company.

The complexity drivers Fuel, Technology, and Stroke were identified as text values. Transforming this data from letter-based parameters into numbers (integer) was necessary to use MLR. Translating the acronyms into numbers took place during interviews with the case company's experts on each parameter to identify the values. Fuel is based on the design difficulty for the 5 different fuel types available, where they all were equally challenging to design except for the system running on oil. Technology is based on gas exhaust treatment. Two technologies included a gas treatment (SGR and EGR), where the last system did not include any gas treatment. Therefore, the system without any gas treatment was identified the easiest, and the remaining two were the same. Stroke is divided into two systems, S and G, where the system S showed to be the most complex given the highest value. However, Later on, the data got transformed into binary code, proving to be the optimal solution for this approach [10].

The design-time estimation model works on MLR. MLR is based on the same theory of simple linear regression. However, instead of one regressor, there will be multiple.

The goal of MLR is to minimize the residuals, which is the error between the data points and the plane, which is done by minimizing the residual sum of squares. To do so, a data frame including all parameters is created based on the complexity drivers in R Studio to identify design-time. First, the data is investigated using the histogram function in R Studio to identify if the data is skewed. If the data was skewed, it became logarithmically transformed to ensure more accurate data [17]. Next, we divide the data into a training set (80%) and a validation set (20%) to enable the possibility of a preliminary analysis. To analyze the data, the linear model function in R is used to fit a plan on the data to identify the y-intercept and slope values (β), which are used to identify the Adjusted R-squared value and p-value. Note that for MLR, the Adjusted R-squared value is used instead of the R-squared value. The summary function is used to calculate and show the Adjusted R-squared and p-values, which are shown in Fig. 1 and Fig. 2. This model uses backward elimination to remove non-significant values, which is executed in steps 1–3. The model runs on a SF-level and includes eight parameters: DSID Registered hours, Stroke, Fuel, Technology, Number of Cylinders, Mark, and Change. The summary function in R Studio allows us to identify the design-times dependency of parameters for this SF, shown in Fig. 1 and Fig. 2.

```
Coefficients:
           Estimate Std. Error t value Pr(>|t|)
change      0.16824    0.02535   6.636 1.84e-06 ***
stroke     -1.86371    0.62351  -2.989  0.00725 **
Technology -1.77340    0.55257  -3.209  0.00440 **
Cyl        -0.37498    0.17429  -2.152  0.04384 *
mark        0.32179    0.14203   2.266  0.03473 *
fuel       -0.17655    0.66829  -0.264  0.79435
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7855 on 20 degrees of freedom
Multiple R-squared:  0.8986,    Adjusted R-squared:  0.8681
F-statistic: 29.53 on 6 and 20 DF,  p-value: 6.266e-09
```

Fig. 1. Output with all parameters

```
Coefficients:
           Estimate Std. Error t value Pr(>|t|)
change       0.1631     0.0158  10.324  1.1e-09 ***
stroke      -1.7579     0.4671  -3.764  0.00114 **
Technology  -1.7189     0.5011  -3.430  0.00251 **
Cyl         -0.3541     0.1519  -2.332  0.02976 *
mark         0.3017     0.1172   2.573  0.01772 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7679 on 21 degrees of freedom
Multiple R-squared:  0.8982,    Adjusted R-squared:  0.874
F-statistic: 37.07 on 5 and 21 DF,  p-value: 9.954e-10
```

Fig. 2. Output, only relevant parameters

The results shown in Fig. 1 automatically runs through the following four steps. Step (1) considers the p-values for each parameter to see if any p-values are above 0.05. Step (2) removes the parameters with a p-value above 0.05. In this case, fuel is removed. Step (3) re-run the model to record how data responded to removing one or more parameters. The model repeats steps 1–3 until all the P-values for the model parameters are below 0.05, resulting in the optimal model shown in Fig. 2.

The identified β-values shown in Fig. 2, column “Estimate”, are the ones used to estimate the design hours using Eq. 1.

$$Design\ hours = exp(\beta_1 * x_1 + \beta_2 * x_2 \ldots \beta_n * x_n) \tag{1}$$

The x-values in the equation come from the complexity drivers, where the value is based on the requested engine design. The model has proven to explain up to 87.4% of the data’s variance, which got identified based on the adjusted R-squared value. Step (4) highlights the model’s test results with two separate plots in Fig. 3. The plot on the left shows the Residuals vs. Fitted values, and the plot on the right is an QQ-plot. The Residuals vs. Fitted plot model becomes unusable if a logical pattern occurs on the plot. Also, the plot shows if any data points might have an undue influence on the model fit. To identify the normal distribution pattern, an ideally straight line on a QQ-plot should show. However, some deviations are accepted for the QQ-plot. On the next page Fig. 3 shows the two different plots for one DSID.

The plots in Fig. 3 visualize the data. It is essential to remove the data deviating to an extend where it damages the model rather than improving it. Removing any data, the model always re-runs steps 1–4. In this case, no data points were necessary to remove. This iterative process continues to the point that all nonacceptable data were eliminated.Evaluating the cleaned data for estimating the design-hours now happens by observing the deviation between the predicted and registered hours. The analysis shows that the more data available from the projects, the more accurate the model will be.

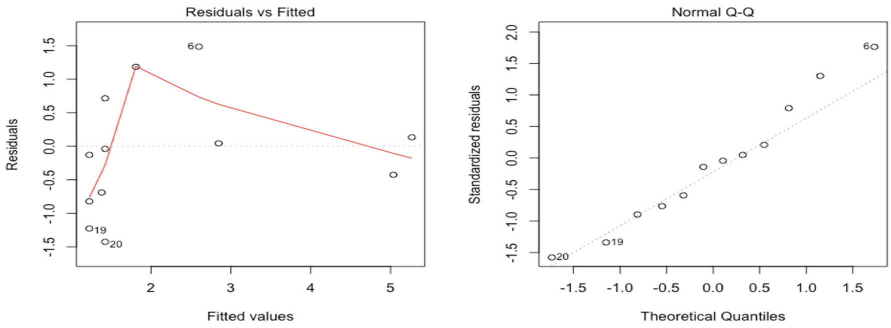

Fig. 3. Plot of the residuals vs. fitted values (left) and the QQ-plot (right)

Small projects with few hours will deviate fast percentage-wise compared to large projects. Hence, small projects do not benefit the model's accuracy. This showed from analyzing all projects percentage-wise, so only bigger projects got included, resulting in better results looking at the correlations and deviation in hours shown in Table 1.

Table 1. Difference in hours

	SF1	SF2	SF3	SF4
Mean hours off	13	28	7	14
Mean percent off	89	16	62	46

Table 2. Correlation for the training and validation data in hours

	SF1	SF2	SF3	SF4
Training data	0.88	0.73	0.96	0.96
Validation data	0.86	NA	0.90	0.72

Analyzing the differences in hours for the SFs, the correlation between estimated and registered hours is examined by creating a training and validation set. The training set has 80% of the observations and helps the model to learn. The remaining 20% of the observations are the validation set, predefining observations to compare them with the training set. Table 2 shows the correlations for the training data and validation data. If the training and validation values are close enough, the model is valid for the given SF. Not applicable (NA) means that the dataset is insufficient for the model to run it. Given the data accessibility, the demonstrated results are positive and validate this method.

3.4 Configurator

The configurator represents the mathematical model as a practical and user-friendly solution for the case company's engineers. The model's simplicity is essential to increase understanding and usage of the configurator [18], and therefore the minimum number of required inputs was included. Hence, no unnecessary parameters after selecting a SF appear in the configurator. This configurator is designed in Excel, where the available variables for each parameter are based on the inputs. The configurator runs on Eq. 1, where the β-values are identified for each parameter based on the mathematical model for the chosen SF. The x-values are added manually to the configurator using a dropdown list for each parameter, where the engine specification is selected. It is believed that the configurator will improve over time as data improves. Figure 4 demonstrates the user interface of the configurator, where the design-time is calculated for a SF.

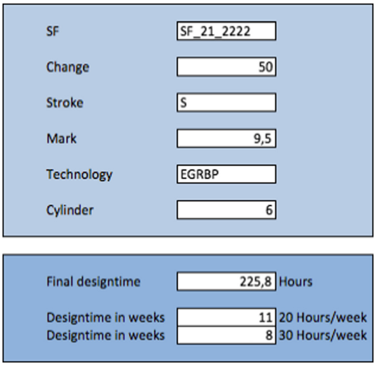

Fig. 4. Final configurator presented for the case company

4 Discussion and Conclusion

This paper developed a mathematical model and a configurator to estimate the design-time based on MLR using complexity drivers. This paper demonstrates the identification of the complexity drivers for the engineering design phase for a highly complex engineered product. The importance of having two types of complexity drivers, static and dynamic got shown by A. Griffen [19]. Static complexity drivers are stable for the SFs data, while experts will decide about the dynamic drivers, such as depth of change. Hence, the dynamic drivers are more subjective. Data availability scoped the project, which led us to a proof of concept. More data can help to determine new complexity drivers. Implementing new parameters in the model is easy. This study focused on identifying complexity drivers based on the various parameters, where the product level had the best data available between the four complexity groups.

The study showed that external complexity drivers highly influence the company through regulations shown in the parameters. The company strategy results in a wide variety of products trying to meet demands in the market. Hence, the focus of the research is on the internal complexity drivers. We identified the complexity drivers in the engineering process and developed a mathematical model to develop the configurator. The configurator for the case company is a fast and easy solution to estimate the design-time for a specific SF, as all the complexity drivers are stable and known for all SFs. However, there are limitations to this configurator. Firstly, the depth of change will be subjective based on the expert's knowledge. Secondly, the inputs require manual work. For the configurator to automatically collect data for each SF, it would require unique ID-numbers for the SFs.

In conclusion, the configurator can be used as an effective solution, but further work is recommended when higher quality data is available. With good data, the configurator can estimate design-time on engine level. Knowing the design-time of every engine will improve the accuracy of capacity planning. Currently, the configurator helps the top management to understand the workload on a SF level. This research extends the existing study by developing a new method based on MLR instead of linear regression. Moreover, this paper identified internal complexity drivers at a case company.

References

1. Earl, C., Johnson, J., Eckert, C.: "Complexity," Des. Process Improv. A Rev. Curr. Pract., 174–197, 2005. https://doi.org/10.1007/978-1-84628-061-0_8
2. Chaitow, L.: Variety. J. Bodyw. Mov. Ther. **8**(1), 1 (2004). https://doi.org/10.1016/S1360-8592(03)00086-X
3. Kreimeyer, M., Lindemann, U., Complexity Metrics in Engineering Design (2011)
4. Wang, Y., Wu, J., Zhang, R., Shafiee, S., Li, C.: A 'user-knowledge-product' Co-creation cyberspace model for product innovation. Complexity **1**, 2020 (2020). https://doi.org/10.1155/2020/7190169
5. Mortensen, N.H., Bertram, C.A., Lundgaard, R.: Achieving long-term modularization benefits: a small- and medium-sized enterprise study. Concurr. Eng. Res. Appl. **27**(1), 14–27 (2019). https://doi.org/10.1177/1063293X18803145
6. Bosch, E., Metternich, J.: Understanding and assessing complexity in cutting tool management. Procedia CIRP **72**, 1499–1504 (2018). https://doi.org/10.1016/j.procir.2018.03.108
7. Hvam, L., Kristjansdottir, K., Shafiee, S., Mortensen, N.H., Herbert-Hansen, Z.N.L.: The impact of applying product-modelling techniques in configurator projects. Int. J. Prod. Res. **57**(14), 4435–4450 (2019). https://doi.org/10.1080/00207543.2018.1436783
8. Shafiee, S.: Conceptual Modelling For Product Configuration Systems. Technical Univerisity of Denmark (2017)
9. Haug, A., Shafiee, S., Hvam, L.: The costs and benefits of product configuration projects in engineer-to-order companies. Comput. Ind. **105**, 133–142 (2019). https://doi.org/10.1016/j.compind.2018.11.005
10. Grabenstetter, D.H., Usher, J.M.: Determining job complexity in an engineer to order environment for due date estimation using a proposed framework. Int. J. Prod. Res. **51**(19), 5728–5740 (2013). https://doi.org/10.1080/00207543.2013.787169
11. Shafiee, S., Nadja, Z., Herbert-hansen, L., Hvam, L.: Development of a Design-time Estimation Model for Complex Engineering Processes (2019)
12. Yin, R.K.: Case study research: Design and methods (applied social research methods). Sage, Thousand Oaks, CA (2009)
13. Voss, C., Tsikriktsis, N., Frohlich, M.: Case research in operations management. Int. J. Oper. Prod. Manag. **22**(2), 195–219 (2002). https://doi.org/10.1108/01443570210414329

14. Shafiee, S., Hvam, L., Haug, A., Dam, M., Kristjansdottir, K.: The documentation of product configuration systems: a framework and an IT solution. Adv. Eng. Inf. **32**, 163–175 (2017). https://doi.org/10.1016/j.aei.2017.02.004
15. Bashir, H.A., Thomson, V.: Estimating design complexity. J. Eng. Des. **10**(3), 247–257 (1999). https://doi.org/10.1080/095448299261317
16. Orfi, N., Terpenny, J., Sahin-Sariisik, A.: Harnessing product complexity: Step 1establishing product complexity dimensions and indicators. Eng. Econ. **56**(1), 59–79 (2011). https://doi.org/10.1080/0013791X.2010.549935
17. Wang, H., Lu, N., Chen, T., He, H., Lu, Y., Tu, X.M.: Changyong FENG Log-transformation and its implications for data analysis. Biostatistics in psychiatry (20), Shanghai Arch. Psychiatry, vol. 26, no. 2, pp. 105–109 (2014). https://doi.org/10.3969/j.issn.1002-0829.2014.02.009
18. Greifeneder, R., Scheibehenne, B., Kleber, N.: Less may be more when choosing is difficult: Choice complexity and too much choice. Acta Psychol. (Amst) **133**(1), 45–50 (2010). https://doi.org/10.1016/j.actpsy.2009.08.005
19. Griffin, A.: Metrics for measuring product development cycle time. J. Prod. Innov. Manag. **10**(2), 112–125 (1993). https://doi.org/10.1016/0737-6782(93)90003-9