

Understanding task execution time in relation to the multilayer project structure: Empirical evidence

Sebastiano A. Piccolo¹, Jakob Trauer², Julian Wilberg², Anja M. Maier¹

¹Technical University of Denmark, Department of Management Engineering

²Technical University of Munich, Department of Mechanical Engineering

Abstract: Estimating task execution time is essential for planning and managing engineering projects. Many process scheduling and optimisation tools and methods require precise task execution time estimates. However, estimates are often too optimistic, potentially harming the usefulness of such tools. In this paper, we develop a methodology to aggregate multiple data sources into a Multiple Domain Matrix and show that its structural properties correlate with task execution time. Specifically, using data from a real-world engineering case, we show that the *size* of a task, the *number of people* assigned to it, and the number of *interfaces* directly correlate with task execution time. We discuss how these measures are available during the planning stage of the process and how people can use them to obtain better estimates.

Keywords: multilayer networks, MDM, task execution time estimation, design project, data science

1 Introduction

In late 2005, the Hamburg Parliament decided to start the construction of a new concert hall in the centre of the city – the “Elbphilharmonie”. Several independent consulting companies estimated € 186.7 million in line with a feasibility study for the completion of this ambitious construction project. The targeted opening date was the 30th March 2010 (Parliament of the Free and Hanseatic City of Hamburg, 2014). By the 4th of November 2016, the building was officially finished – a delay of more than six years with a budget overrun of more than € 679 million. The “Elbphilharmonie” is just one example of project mismanagement and exemplifies the potentially catastrophic consequences of unrealistic and undersized estimations of budget and time. Good time estimates are crucial to project success (Murmman, 1994; Thamhain and Wilemon, 1986) and many tools have been developed in the attempt to support experts in their estimates and project planning (Bashir and Thomson, 2001; O’Donovan et al., 2005).

Why do experts underestimate project completion time? Humans have a tendency to underestimate the difficulties of the tasks for which they are providing estimations (Flyvbjerg, 2006; Kahneman and Lovallo, 1993). In addition, the tasks to estimate are often considered in isolation without a systemic understanding of the whole (Kahneman and Lovallo, 1993). For this reason, calls to action for using historical data to correct and/or inform time estimations have been made (Flyvbjerg, 2006; Halkjelsvik and Jørgensen, 2018).

In this paper, leveraging the intersection between engineering design and network science, we combine three different data sources from a large-scale design project of a biomass power plant in order to understand task completion time in relation to the project structure. We show that task completion time correlates positively with the number of documents produced within the scope of a task, the number of tasks to which a task is connected, and the number of people assigned to it. Our results are in line with previous research and show that the analysis of historical or archival data can generate a useful understanding of factors that can affect a project. We discuss how such an approach can offer a more global view and support project planners in estimating task completion time. After a brief overview of the background and related literature (section 2), we introduce the datasets and the analysis methods (section 3). We report the results (section 4) and discuss their implications, connections with extant literature, and avenues for future research (sections 5 and 6).

2 Background

Estimating project completion time is a crucial task in the life of a project. Time estimates are important not only for financial reasons such as to present the project to possible investors, time estimates are an input variable of many project management tools. Project scheduling techniques such as the Process Evaluation and Review Technique (PERT) and the Critical-Path Method (CPM) (Project Management Institute, 2017) or techniques based on Design Structure Matrices (DSM) (Eppinger and Browning, 2012) require entering completion time for each task. As a result, errors in the estimations of tasks completion time can seriously harm the subsequent project planning and management.

Despite the models developed (for instance, Bashir and Thomson, 2001; Srinivasan and Fisher, 1995), expert estimation seems to be the most common way to estimate effort and completion time (Halkjelsvik and Jørgensen, 2018; Project Management Institute, 2017). On the one hand, expert estimation has its advantages, as experts may have important domain knowledge that the model does not include (Jørgensen, 2004). On the other hand, expert estimations are inherently prone to human and situational biases (Jørgensen, 2004; Kahneman and Lovallo, 1993) that make them too optimistic. This optimism bias happens as experts tend to consider problems as unique, not accounting for similar cases. That is, expert estimations rely on the “inside” view, which only takes the structure and the impediments of the specific case into account. An “outside” view, on the other hand, takes *distributional* information of similar cases into account (Kahneman and Lovallo, 1993).

Studies that investigate what factors relate to execution time offer useful insights to take a more “outside” view to time estimates. Lanigan, (1994) showed that task effort is a function of the *nature* of the task itself and the number of people working on it. Kakimoto et al., (2018) showed that maximum team size to estimate effort of a project is effective and robust to perturbations when the error rate is equal or less than 50%. In software engineering, different studies relate the size of software, captured by number of lines of code, function points, or number of files, to the execution time or development effort (Albrecht and Gaffney, 1983; Boehm, 1984; Symons, 1991).

In this paper, we connect the previous insights with the domain of Engineering Design, testing the overall hypothesis that task execution time can be predicted, to some extent, from the properties of the networked structure of the project.

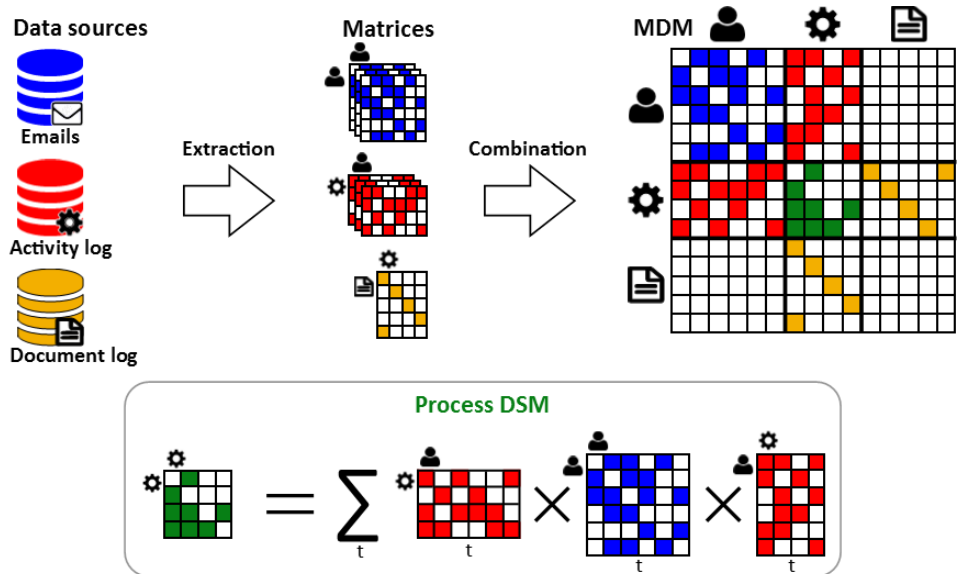


Figure 1: Process of data extraction and combination to build the Multiple Domain Matrix (MDM) used to understand task execution time in relation to project structure.

3 Data and Methods

3.1 Data

The data used in this paper refers to a large-scale design project of a biomass power plant conducted by a multi-project Scandinavian company (Parraguez et al., 2015). Three different data sources are available:

- An activity log, which records the activities performed by the company's personnel throughout the duration of the design process. The activity log describes the relations between 100+ people and ~150 unique activities. Each activity is identified by an activity code assigned by the software that the company uses to manage the project.
- A document log, which contains metadata for the 3000+ documents created during the design process. The metadata include information about document creation and last modification dates, external companies involved in the document editing process, and the code of the activity to which each document is related.
- The complete email exchange between all the people involved in the project (employees, suppliers, external consultants, etc.). The complete email archive amounts to ~54000 emails.

3.2 Methodology to build the MDM automatically from data sources

In order to understand a design project in relation to its multilayer network structure, we need to extract the fundamental networks (matrices) from the data sources, in a way that makes them combinable into one Multiple Domain Matrix (MDM) (Figure 1). From the document log, we extract a matrix that maps each document to the activity it refers to. From the activity log, we extract a series of monthly bipartite networks, also known as Domain Mapping Matrices (DMM), that represent the assignment of people to activities, connecting each person to the activities performed in one month. Similarly, from the email archive, we extract a series of directed networks that connect the company's employees based on the monthly email conversation. As the design process under analysis is closer to a Systems Engineering process rather than an agile one, monthly aggregation is appropriate. We tried other more refined aggregations, such as weekly, but the results remained unchanged.

The activity network that describes the information dependencies between the activities that compose the process is obtained by applying relational algebra for networks. Let PA_t be the matrix describing the assignment of people to activities at time t , and PP_t the communication between people, as captured by the email communication, at time t , the activity network at time t is computed with the following formula: $AA_t = PA_t^T \cdot PP_t \cdot PA_t$. The final activity network for the MDM is computed by aggregating (summation) all the snapshots AA_t into a single one. The matrices extracted as described above are then aggregated and combined to form the MDM (Figure 1). Considering the evolution over time for PP and PA is important to avoid an unrealistic process DSM that is too dense, where each activity may be connected to nearly any other (Figure 2).

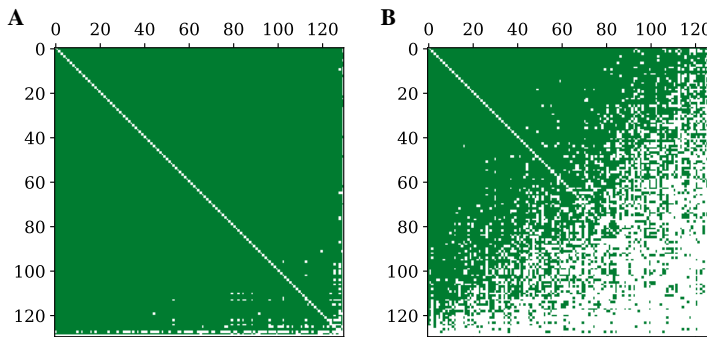


Figure 2: Comparison between the process Design Structure Matrix (DSM) obtained by aggregating all the temporal information into one single snapshot (A) and by using monthly snapshots (B).

3.3 Modelling

In this paper, we focus on understanding task completion time in relation to the multilayer structure of the project. Guided by the insights discussed in the literature review and in accordance with our hypothesis that structural properties of the project can predict, to a certain extent, completion time, we extract the variables of interest from the MDM.

As the completion time for the tasks is expressed in number of days, thus is a positive integer, we use models of the following form:

$$\log(y_i) \sim \alpha + \beta X_i \quad (1)$$

Where y_i is the completion time for the i -th activity, α is a constant term, X_i is the vector of explanatory variables, and β its relative vector of coefficients. To fit the model we use the ordinary least squares method (OLS) with robust standard errors to account for possible heteroscedasticity. The logarithm transformation of the completion time is useful to reduce the skewness of the distribution. To evaluate the goodness of the models we use the following measures: the R^2 , the adjusted R^2 , the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and the root mean square error (RMSE). For R^2 and adjusted R^2 , the higher the better; for the other measures, the lower the better. Finally, to check for multicollinearity, we computed the condition number and the variance inflation factors (VIFs). In the following, we describe and discuss the variables that we use in our modelling approach to explain task completion times. The dependent variable, i.e. the variable that we seek to explain using structural properties of the MDM, is the activity execution time. We use activity and document logs to compute the completion time for each activity. As the activity log has data on a daily granularity, we count the number of days elapsed between the first and last time a person worked on an activity or a document connected to it. To account for the *size* of each activity, we compute the number of documents connected to it (#Documents). In the MDM, this corresponds to the degree of the activities in the DMM activity-document (see Figure 1). As each document deals with a set of functional requirements, the number of documents can be interpreted as an approximate measure of the functional requirements of an activity. Furthermore, the number of documents can give a first estimate of the workload of the teams involved (Piccolo et al., 2017). We expect a positive relation between the number of documents and completion time.

For each activity, we compute the number of people (#People) allocated to it as the degree of the activities in the DMM activity-people (see Figure 1). This DMM proved to be highly relevant to understand the role of people in the robustness of a design process (Piccolo et al., 2018). The number of people connected to an activity can be interpreted as an approximation of the workforce needed by the activity. In addition, activities with high number of people assigned to them can be more error prone (Piccolo et al., 2018); thus, we expect a positive relation between the number of people connected to an activity and its completion time.

We account for the structure of the activity network and the amount of information dependencies affecting each activity by computing a set of measures: 1) the degree of each activity (#Activities), i.e. the number of ingoing and outgoing edges; 2) the indegree, i.e. the number of ingoing edges and quantifies the dependency of an activity from the preceding ones; 3) the outdegree, i.e. the number of outgoing edges and quantifies the influence of an activity on the following ones; 4) the product of indegree and outdegree, here termed criticality, which accounts for a synergistic relation between in- and outdegree. We expect a positive relation between these structural properties and the completion time.

Finally, we compute the number of external companies involved in each activity as a possible confounder for the measures computed above. The rationale for including this

confounder is that a higher number of external companies involved in an activity could produce more difficulties in coordination and thus, increase overall completion time.

All variables, before the statistical modelling, were normalised by removing their averages and dividing them by their standard deviations. Table 1 shows the correlation between the explanatory variables. We note that the correlation between degree, indegree, outdegree, and criticality is very high (almost perfect correlation). Thus, we present only the models with #Activities, without the other correlated variables to avoid inconsistencies due to multicollinearity. Interpretation for the other variables is the same as for the degree.

Table 1. Correlations between explanatory variables. High correlations ($r \geq 0.7$) highlighted.

	2	3	4	5	6	7
1. # Documents	0.45	0.38	0.37	0.35	0.39	0.4
2. # People		0.36	0.7	0.7	0.71	0.78
3. # Companies			0.42	0.45	0.41	0.47
4. # Activities				0.98	0.98	0.95
5. Indegree					0.95	0.96
6. Outdegree						0.96
7. Criticality						

4 Results

We present the results of our analysis in Table 2. First, we develop a baseline model that accounts only for the effect of the number of documents, the number of people, the number of activities, and the number of external companies involved. All the terms are positive and significant, with the exception of the number of people and the amount of companies. We develop a second model to account for the possibility of non-linearity in the number of people and the activities' degree. The complete model represents an improvement over the baseline with an increase of ~25% for the explained variance (R^2 and Adjusted R^2). The coefficients confirm the expectations of positive relations between the number of people, documents, activities, and completion time.

The number of people and activities are associated non-linearly and monotonically with completion time (see Figure 3 for a visualisation of the relations). Finally, observing that the number of external companies is not significant, we remove it obtaining a reduced model that has the same explanatory power as the previous one. The coefficients remain significant, describing the same positive associations of the variables with the completion time. Our models do not suffer of multicollinearity, as confirmed by the Variance Inflation Factors (VIFs) and condition numbers smaller than 10.

Table 2. Regression table. Dependent variable: execution time

Coefficients	Baseline	Complete	Reduced
Constant	5.49*** (0.13)	6.10*** (0.19)	6.12*** (0.19)
log(#Documents)	0.56*** (0.14)	0.60*** (0.14)	0.54*** (0.13)
#People	0.14 (0.13)	1.02*** (0.25)	1.02*** (0.25)
#People ²		-0.38*** (0.10)	-0.37*** (0.10)
#Activities	1.00*** (0.20)	0.36 (0.25)	0.31 (0.24)
#Activities ²		-0.24* (0.11)	-0.26* (0.11)
#Companies	-0.16 (0.10)	-0.15 (0.09)	
R ²	0.43	0.52	0.52
Adjusted R ²	0.41	0.50	0.50
AIC	493.85	474.36	473.40
BIC	508.19	494.43	490.60
RMSE	7.72	6.95	7.59
#Observations	130	130	130
*** p < 0.001, ** p < 0.01, * p < 0.05		Standard errors in parentheses	

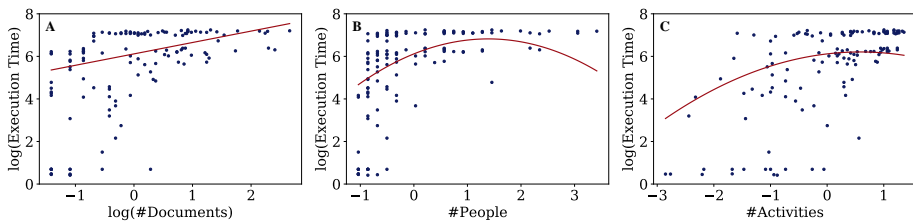


Figure 3: Relations between task execution time and #Documents, #People, and #Activities. The negative numbers are due to variable standardisation.

5 Discussion

Estimating task execution time is an important activity for planning and managing engineering projects, as many scheduling tools require task completion time estimates as one input variable. However, time estimates are often too optimistic because of cognitive biases that prevent experts to realise and consider the many factors influencing task execution. Here, we proposed to understand execution time in relation to the multilayer structure of a project through the use of a Multiple Domain Matrix (MDM). Differently from traditional approaches that rely on interviews, we developed a method to build the MDM automatically from three data sources: email communications, activity logs, and document logs.

While we analysed only one project and the specific value of the regression coefficients pertain only to this case, our analysis produced results in line with current practice in software engineering and insights that we believe are useful to improve the practice of time estimation and project management. We discuss them in the following. Our modelling strategy shows that the task completion time can be modelled as a function of the number of documents produced in the context of the task (task size), the number of people allocated to it (resource allocation), and the number of interfaces with the other tasks (task interfaces).

Task size: The number of documents, here, is a proxy for the *size* of a task, as the lines of code or the number of function points are in software development (Albrecht and Gaffney, 1983; Boehm, 1984; Symons, 1991). The positive relation between the number of documents and execution time (see Figure 3A) shows that “task sizing” can be useful also outside software engineering. We found that the logarithm of the number of documents performs better than the crude number, which means that a perfect estimation of the size is not necessary and a measure of the order of magnitude would perform well. Understanding which measures of task size are the most suitable for engineering design is a topic for future research and we suspect that a measure derived from the functional requirements, as it happens in software engineering (ISO, 2007), can be a good starting point.

Resource allocation: We have also found a positive relationship between the number of people assigned to an activity and its execution time. In Figure 3B, it is clear that the relation is monotonic. The quadratic curve starts decreasing after ~90.5% of data points and does not represent a good fit anymore. The positive relation between the number of people allocated to an activity and its completion time shows that the amount of people assigned to a task should be used to make time estimations as tasks with higher number of people require more time. This is especially important as it has been documented, under the name *team scaling fallacy*, that underestimation of completion time increases as team size increases (Staats et al., 2012). Furthermore, activities with a high number of people assigned to them are more important for process robustness as errors or changes originating through such tasks can spread faster and affect more activities (Piccolo et al., 2018).

Task interfaces: The number of interfaces an activity has with other activities is also positively associated with the completion time. The relation is monotonic and no turning point is observed (see Figure 3C). Thus, in case of the relation between completion time and number of interfaces, we do not find a curvilinear relation, as for example claimed in Gokpinar et al., (2010) for the relation between the number of interfaces of a subsystem and the number of defects. We also found that the number of interfaces in input (indegree) has almost the same explanatory power as the total count of interfaces. This means that the completion time is in direct relation with the number of inputs that a task has to integrate. The relation between the degree and completion time reminds us of the importance of integrative activities during error propagation processes (Braha and Bar-Yam, 2007; Piccolo et al., 2018).

With a measure of activity size, people assigned to activities, and number of interfaces we were able to explain 50% of variance in the completion time. We argue that these measures, such as the number of people allocated to a task, are readily available or can be estimated during the planning stage of a project. The number of interfaces per activity

can be obtained by building the DSM for process sequencing. The measure of activity size could be estimated from the amount of functional requirements. One could be tempted to explain more variance by adding more variables to the models. While there are definitely many more factors that can affect task completion time, it is worthy to remember that the use of irrelevant information hinders good time estimates (Halkjelsvik and Jørgensen, 2018). We believe that the process of data analysis and the measures used here can be used to support experts in making better estimates, while helping them to take a more outside view (Kahneman and Lovallo, 1993). Studying how to integrate these and other metrics as well as the process of data analysis into the practice of project management is a topic for future research.

6 Conclusions

Estimating task completion time is difficult and often results in underestimates due to optimism bias and other human and situational biases and a lack of meaningful information on which to base the estimates. To provide a ground for better estimates, this paper combined and analysed multiple data sources from a large-scale design project, showing that task completion time relates to the structure of the project as captured by a Multiple Domain Matrix (MDM). Statistical analyses showed that task execution time correlates positively with the *size* of the task, the number of *interfaces* with other tasks, and the number of *people* allocated to the task. In our case, we were able to explain 50% of the variance. We discussed implications of the findings and gave pointers on how the three metrics used can be made available to managers during the planning stage of a project. Moreover, this study also showed a possible use of historical data to inform future decision-making.

References

- Albrecht, A.J., Gaffney, J.E., 1983. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Trans. Softw. Eng.* SE-9, 639–648. <https://doi.org/10.1109/TSE.1983.235271>
- Bashir, H.A., Thomson, V., 2001. Models for estimating design effort and time. *Des. Stud.* 22, 141–155. [https://doi.org/10.1016/S0142-694X\(00\)00014-4](https://doi.org/10.1016/S0142-694X(00)00014-4)
- Boehm, B.W., 1984. Software Engineering Economics. *IEEE Trans. Softw. Eng.* SE-10, 4–21. <https://doi.org/10.1109/TSE.1984.5010193>
- Braha, D., Bar-Yam, Y., 2007. The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. *Manage. Sci.* 53, 1127–1145. <https://doi.org/10.1287/mnsc.1060.0617>
- Eppinger, S.D., Browning, T.R., 2012. *Design Structure Matrix Methods and Applications*. The MIT Press.
- Flyvbjerg, B., 2006. From Nobel Prize to Project Management: Getting Risks Right. *Proj. Manag. J.* 37, 18–19.
- Gokpinar, B., Hopp, W.J., Iravani, S.M.R., 2010. The Impact of Misalignment of Organizational Structure and Product Architecture on Quality in Complex Product Development. *Manage. Sci.* 56, 468–484. <https://doi.org/10.1287/mnsc.1090.1117>

Part IV: Using Data

- Halkjelsvik, T., Jørgensen, M., 2018. Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life. Springer International Publishing, Cham. <https://doi.org/10.1007/978-3-319-74953-2>
- ISO, 2007. Information technology – Software measurement – Functional size measurement. Part 1: Definition of concepts, International Standard ISO/IEC 14143-1.
- Jørgensen, M., 2004. A review of studies on expert estimation of software development effort. *J. Syst. Softw.* 70, 37–60. [https://doi.org/10.1016/S0164-1212\(02\)00156-5](https://doi.org/10.1016/S0164-1212(02)00156-5)
- Kahneman, D., Lovallo, D., 1993. Timid Choices and Bold Forecasts: A Cognitive Perspective on Risk Taking. *Manage. Sci.* 39, 17–31. <https://doi.org/10.1287/mnsc.39.1.17>
- Kakimoto, T., Tsunoda, M., Monden, A., 2018. Should Duration and Team Size Be Used for Effort Estimation?, in: Lee, R. (Ed.), *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. Springer International Publishing, Cham, pp. 91–105. https://doi.org/10.1007/978-3-319-62048-0_7
- Lanigan, M.J., 1994. Task estimating: completion time versus team size. *Eng. Manag. J.* 4, 212. <https://doi.org/10.1049/em:19940508>
- Murmann, P.A., 1994. Expected Development Time Reductions in the German Mechanical Engineering Industry. *J. Prod. Innov. Manag.* 11, 236–252. <https://doi.org/10.1111/1540-5885.1130236>
- O'Donovan, B., Eckert, C., Clarkson, J., Browning, T.R., 2005. Design planning and modelling, in: *Design Process Improvement*. Springer, pp. 60–87.
- Parraguez, P., Eppinger, S.D., Maier, A.M., 2015. Information Flow Through Stages of Complex Engineering Design Projects: A Dynamic Network Analysis Approach. *IEEE Trans. Eng. Manag.* 62, 604–617. <https://doi.org/10.1109/TEM.2015.2469680>
- Piccolo, S.A., Lehmann, S., Maier, A., 2018. Design process robustness: a bipartite network analysis reveals the central importance of people. *Des. Sci.* 4, e1. <https://doi.org/10.1017/dsj.2017.32>
- Piccolo, S., Lehmann, S., Maier, A., 2017. Using data- and network science to reveal iterations and phase-transitions in the design process, in: *Proceedings of the 21st International Conference on Engineering Design (ICED17)*, Vol. 2: Design Processes | Design Organisation and Management. Design Society, pp. 11–21.
- Project Management Institute, 2017. A guide to the project management body of knowledge (PMBOK® guide), Project Management Journal.
- Srinivasan, K., Fisher, D., 1995. Machine learning approaches to estimating software development effort. *IEEE Trans. Softw. Eng.* 21, 126–137. <https://doi.org/10.1109/32.345828>
- Staats, B.R., Milkman, K.L., Fox, C.R., 2012. The team scaling fallacy: Underestimating the declining efficiency of larger teams. *Organ. Behav. Hum. Decis. Process.* 118, 132–142. <https://doi.org/10.1016/j.obhdp.2012.03.002>
- Symons, C.R., 1991. *Software Sizing and Estimating: Mk II FPA (Function Point Analysis)*. John Wiley & Sons, Inc., New York, NY, USA.
- Thamhain, H.J., Wilemon, D.L., 1986. Criteria for controlling projects according to plan. *Proj. Manag. J.* 17, 75–81.

Contact: Sebastiano A. Piccolo, Technical University of Denmark (DTU), Department of Management Engineering, Produktionstorvet, Building 424 DK-2800, Kongens Lyngby, Denmark, sebpi@dtu.dk