INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016
Dubrovnik - Croatia, May 16 - 19, 2016.

A SIMULATION-BASED ANALYSIS ON THE INTEGRATION OF PROGRAM MANAGEMENT AND SYSTEMS ENGINEERING

L. Becerril, E. Rebentisch, N. Chucholowski, E. Conforto and U. Lindemann

Keywords: program management, systems engineering, system dynamics

1. Introduction

Successful Engineering Programs or large-scale projects, such as the Oresund bridge linking eastern Denmark and southern Sweden and NASA's Mars Pathfinder [Locatelli et al. 2013], create value for their stakeholders, the involved organizations and countries. The cost overrun within Engineering Programs in the last ten years was, according to Oehmen et al. [2010], around 40%.

Previous studies associate this poor performance partly with a growing cultural barrier between Program Management (PM) and Systems Engineering (SE) [Conforto et al. 2013a,b], [Conforto and Rebentisch 2014]. This lack of integration between PM and SE leads to duplicated work and effort, separate tracking of requirements and poor coordinated schedules – among other issues [Langley 2011]. For developing effective recommendations that lead to better performing programs, an understanding on how the integration of these two disciplines (referred to as 'integration' in the following) affects program performance is still missing. In this context, 'integration' "results from the team's ability to combine Program Management and Systems Engineering practices, tools and techniques, experience and knowledge in a collaborative and systematic approach […] achieving a common goal/objective in complex program development environments" [Conforto and Rebentisch 2014, p.20].

The study presented in this paper examines the relationship between this 'integration' and program performance. This is achieved by decomposing the causal relationships between previously identified organizational, processes-related, and people-related factors that contribute to 'integration' and the behavior of engineering programs. For this purpose, System Dynamics - an established approach to analyze development projects - is applied to model and analyze program behavior under different levels of 'integration'. The Model is developed based on (1) previous and current research regarding the relationship between PM and SE, (2) existing project and multi-project system dynamics models, and (3) literature on Program and Project Management, Systems Engineering, and related disciplines. The objectives of this study are to contribute to the understanding of the relation between 'integration' and program performance and to identify leverage points that influence program behavior, specifically those related to PM and SE integration. First, this paper presents an overview of existing research on the integration of Program Management and Systems Engineering. Thereafter, the high-level structure of the System Dynamics model developed is presented, as well as the modelling methodology. The fourth section presents the simulations and analysis results based on the model. Finally, the last part discusses the contribution of the study and presents an outlook.

2. Program management and systems engineering in engineering programs

Engineering programs are defined as “a group of interrelated projects, subprograms, and program activities that are managed in a coordinated way to obtain benefits not available from managing them individually” [PMI 2013, p.4]. Other important characteristics of programs are their long duration, their complex value proposition, and the creation of new enterprises through forming a (new) network of participants [PMAJ 2008], [Oehmen et al. 2010], [Lucae et al. 2014].

The value of PM consists in managing, integrating and coordinating this interrelated group of projects (the program). Hence, benefits and control, which are not available by managing projects individually, can be obtained. In contrast, SE is defined as an interdisciplinary approach to enable the realization of successful systems [INCOSE 2011]. SE focuses on providing a “quality product that meets the user needs” [INCOSE 2011, p.7] and considers both technical and managerial aspects, with the latter addressing governance of technical work [Ferris 2007], [Locatelli et al. 2013].

In Engineering Programs, program managers and systems engineers work closely together, especially in defining the systems life cycle and planning key decision gates to meet their specific needs [INCOSE 2011]. While Program Management has overall program and strategic accountability, Systems Engineering has accountability for the technical and systems elements of the program [Langley 2011]. However, there is an overlap between SE and PM activities and responsibilities, which is evident by comparing the lead practitioners’ literature (c.f. [INCOSE 2011], [PMI 2013]). Additionally, a survey conducted by Conforto et al. [2013b] identified shared responsibilities between program managers and systems engineers from the respondents' experience. These included quality management, risk management life cycle planning and external supplier relations. However, the exact role of the program manager and the systems engineer may vary from organization to organization.

According to [Conforto et al. 2013b], an overlap of responsibilities between PM and SE is rather common. However, a growing cultural barrier contributes to programs' poor performance since some systems engineers and program managers believe that their work activities are separate from each other rather than complementary [Langley 2011], [Conforto and Rebentisch 2014]. Consequently, they apply fully different approaches to the key work such as planning, defining components and their interactions, and integrating these components [Langley 2011]. According to Langley [2011], this lack of integration results in unreliable schedules that are developed independently of the technical scope, duplicated work and effort, program team members receiving conflicting direction, and separate management and tracking of requirements, often leading to program outcomes that are different from what the customer or end user expects.

Until now, a number of organizational, process-related and people-related factors that contribute to the integration of PM and SE has been identified [Conforto and Rebentisch 2014]. Additionally, Reiner [2015] provides insights into how performance may be affected by PM and SE integration. This study consists of a survey that links these organizational, process-related, and people-related factors, to decision-making, collaborative work, and information sharing practices, and these in turn to program performance. As a result, a positive correlation between 'integration' and program performance has been revealed [Reiner 2015]. However, these empirical studies [Conforto and Rebentisch 2014], [Reiner 2015] only provide a static view of completed programs. Thus, the mechanisms how the organizational, process-related and people-related factors impact integration and program behavior are missing. To identify where management actions can be more effective, a dynamic perspective of the system is essential. Furthermore, in empirical studies differences in performance which resulted from external factors, such as regulations, are difficult to recognize. Additionally, confidentiality issues and long program times make it difficult to capture the endogenous behavior of specific programs. In contrast, a model - as proposed in the present paper - allows the simulation of different levels of integration between PM and SE under the same program circumstances in a controlled environment. These simulations improve our understanding of what contributes to more effective integration between PM and SE, and in turn, how 'integration' impacts overall program performance.

3. System dynamics model

According to Browning et al. [2006], complex processes and behaviors can be understood by examining their relatively simpler parts and those parts’ endogenous and exogenous relationships. Building on these premises, a modeling approach that dissects the influence of PM and SE integration into simpler and previously analyzed causal relationships was chosen.

System Dynamics (SD) is an approach for modeling and analyzing dynamic systems. Its fundamental concepts are information feedback systems and decision-making processes [Forrester 1961], [Sage and Rouse 2009]. In the context of Project Management, System Dynamics models are able to capture critical project features, such as concurrence within and between phases, schedule and budget pressure, iteration and rework, and impact of quality and productivity [Ford 2009]. Thus, this approach has been widely applied to project management, including models of large scale projects for strategy assessment and in the context of litigation in shipbuilding, aerospace and civil construction [Sterman 1992].

The model developed is based on a selection of project and multi-project System Dynamics models[1] and it comprises of a set of interrelated projects within a program. The fundamental enhancement to existing models is that PM and SE integration influences program behavior in two aspects: it regulates the exchange of information among projects by affecting the communication and collaboration capabilities of the enterprise; and it influences the resource allocation process by regulating how the program manager and the chief systems engineer make decisions together.

These adaptations and the overall structure of the model will be further described below. Moreover, the model presented in this work focuses on the mid-early and middle phases of engineering programs, i.e. on the development of the system to deliver. The very early phases, such as the definition of the programs scope, and the later phases, such as system operation and disposal are outside the scope of this work. The influences of specific programs' external factors, for example political environment or customer relationships, are also excluded from this study. Furthermore, the model's validity is ensured through the methodological development based on proven models and through a series of structural and behavioral tests, such as extreme conditions tests.

3.1 Methodology

This study's methodology is based on the General Approach for Modeling Projects by Kohn [2014]. This approach defines four essential steps for the modeling, namely: model purpose definition, model development, model validation and verification, and model use – in this order, however it allows jumps and iterations [Kohn 2014]. First, during model purpose definition, the model objectives, the system-of-interest, and its boundaries were defined. The second step "model development" comprised from selecting the model type to the actual modeling activity. System Dynamics was selected early in the process since it deals with the dynamic complexity of projects, where most traditional project management tools and mental models are inadequate [Lyneis et al. 2001].

The model development step includes - besides the modelling activity - examining available resources and acquiring information as basis for model building. An extensive literature review indicated a lack of studies and sources to develop the model based on literature since the proposed issue is very specific. However, the information gathered from the studies mentioned above (c.f. Conforto et al. [2013a,b] and Conforto and Rebentisch [2014]), existing System Dynamics models and discussions with experts, provided sufficient information for developing the model's structure illustrated in Figures 1, 2 and 3. Meanwhile, an empirical study by Reiner [2015] delivered the necessary information for calibrating and validating this structure as explained in the section below. Following, the system dynamics model itself was developed according to the modeling process suggested by Sterman [2000].

During the model development, the model's validity was ensured through its systematic development process. Furthermore, the integration-related factors and parameters were derived from a survey among 157 practitioners - mostly program managers and systems engineers. The survey participants had an average of 8-10 years of experience in the sectors of Technology and Information (32%), Aerospace and defense (25%), Consulting (7 %) and Construction (6%), among other industries [Reiner 2015]. During the test phase, discussions and reviews with subject-matter and System-Dynamics experts provided valuable feedback. Additionally, the model was tested according to the recommendations of Sterman [2000] with regard to its parameters and assumptions through sensibility tests, extreme conditions tests,

---

[1] The models for modeling project management this work is based on are depicted in Cooper [1993], Ford et al. ([Ford 1995], [Lyneis and Ford 2007]), and Herweg and Pilon [2001] as shown in Figure 3

and other structural and behavioral tests. The last step, namely model use, involves the simulations and analysis of the results. Further uses of the model are discussed in the outlook.

3.2 Model structure

One of the advantages of modeling and simulation is the possibility to examine the effects of the organizational, people-related and process-related factors under constant external conditions and program properties. Since the main interest of the model is the integration between PM and SE, the model focuses on examining the differences in performance in response to changing levels of integration. Therefore, a stable development environment, process and organization are assumed throughout the program life.

As mentioned above, 'integration' is a result of a number of organizational, process-related and people-related factors. In this study, we also introduced three aggregated factors as shown in Figure 1. Furthermore, 'integration' itself is not explicitly modelled but measured through an "integration indicator" which comprises all integration-related aggregated factors according to the definition of 'integration'. Figure 1 gives an overview of these factors and their relationships. This structure[2] was first developed based on the Integration Framework for Program Management and Systems Engineering [Conforto et al. 2013] and discussions with experts, and enhanced by a literature survey. The structure was then refined and verified through regression analysis and ANOVA testing performed on the industry survey results of Reiner [2015]. Factors with no statistical significance were removed from the structure. For the auxiliary variables "PM and SE decision involvement" and "Decision making information" no empirical data was available.

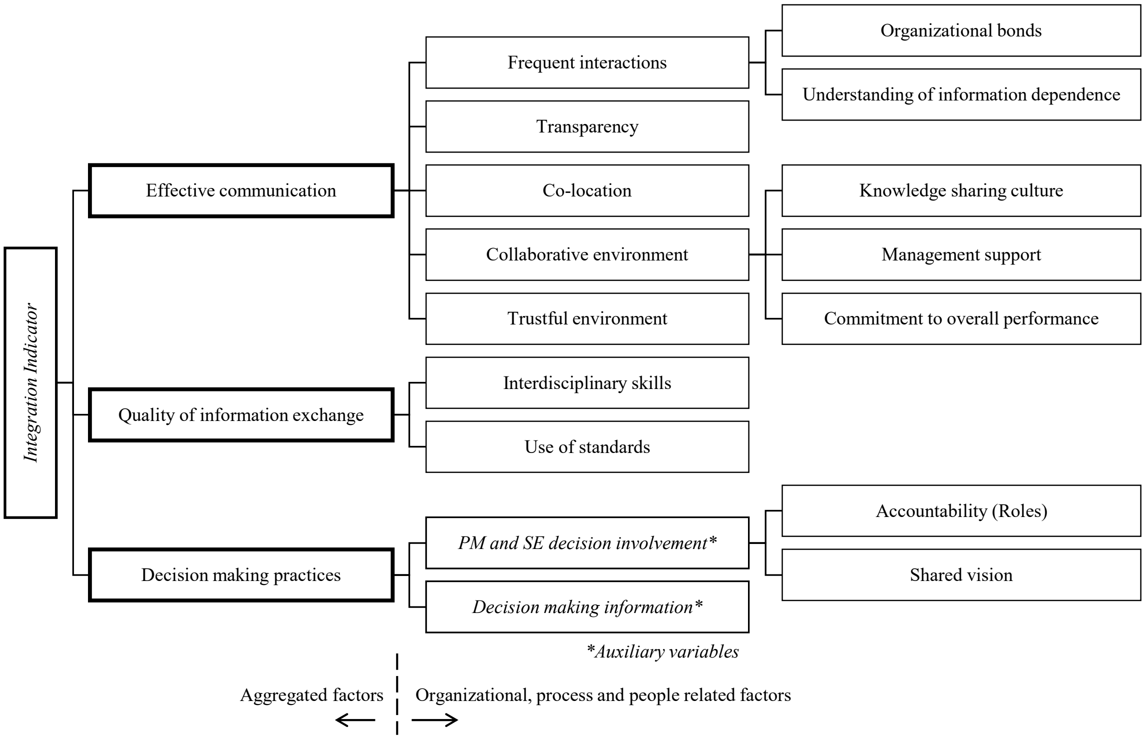

Figure 1. Structure of the organizational, process-related and people-related factors that contribute to 'integration'

The model simulates five interdependent development projects within a program (Figure 2, right). Each of these projects is represented through a rework cycle. The rework cycle [Cooper 1980, 1993a,b,c] is a simplified model of a project or a project phase, where a given number of tasks are being done. It has been traditionally used in project models since its introduction [Lyneis and Ford 2007]. The tasks are

---

[2] The term structure is used here to distinguish it from the systems dynamics model.

the fundamental units of work that flow through the project. They can be done either correctly or defectively; defective tasks have to be reworked causing backlogs. Furthermore, resources are represented as manpower. Additionally, the model presented in this study includes changes, which are triggered by iterations within a development program or by other changes. The number of changes depends on the program's scope and on an uncertainty constant. Moreover, the five projects represented within the program are interrelated through information dependencies, change propagation, and common resource sharing (Figure 2, left). The number of projects and their dependencies on each other can be adjusted easily to represent a specific program.

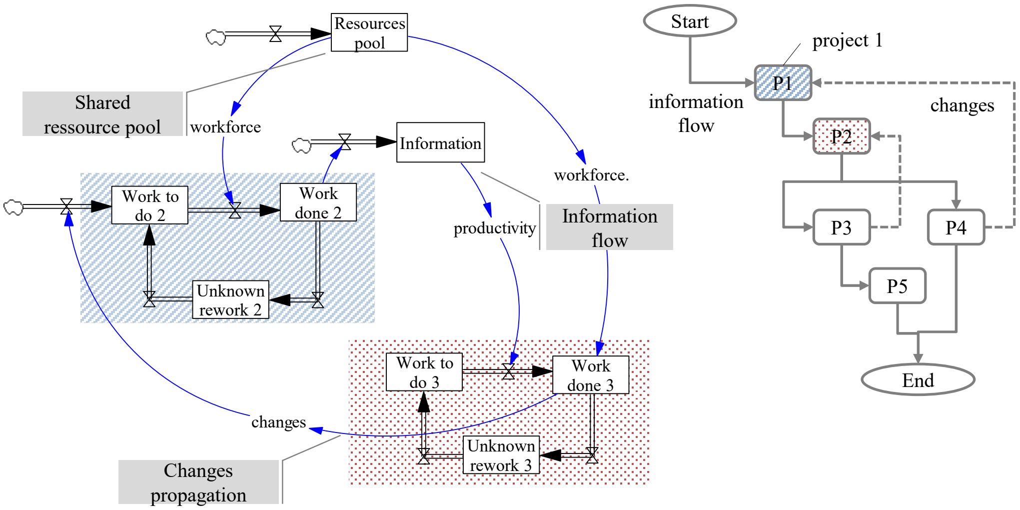

Figure 2. Relationships among sub-projects in SD program-model

The model's structure is divided into four modules, as illustrated in Figure 3: the single project model (based on the rework cycle), the information flow process, the decision making process, and the resource allocation process. The single project model and the resource allocation process have been adapted from existing models. The information flow and the decision-making process have been developed within the scope of this study.

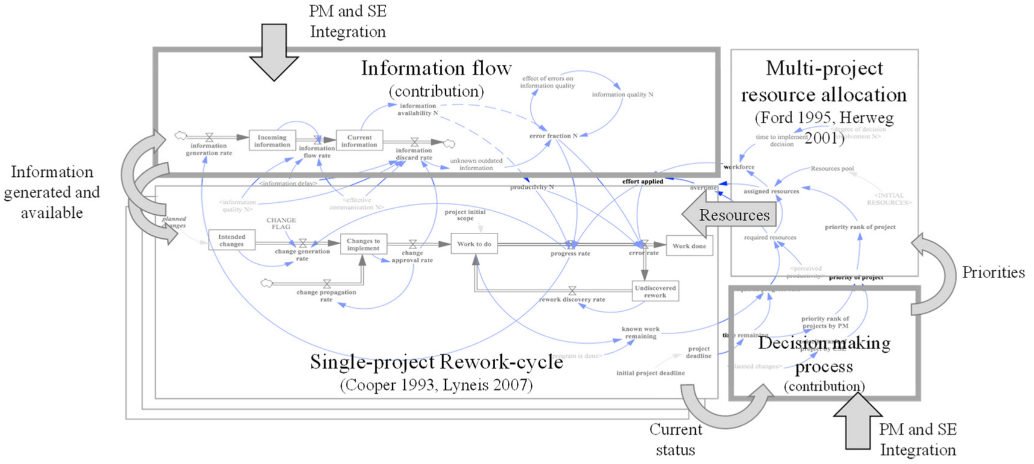

Figure 3. Model structure and modules

As shown in Figure 3, each project exchanges information with other Projects through the information flow, which is influenced by the integration-related aggregated factors "Effective communication" and "Quality of information exchange", and their corresponding sub-factors depicted in Figure 1. The available information impacts the productivity and error rate of the subsequent project. Furthermore, each project gives a status update to a central decision-making process, which subsequently gives priorities to the projects based on the following criteria: the project's delay, their priority from the systems perspective (constant), integration-related factors "Accountability" and "Shared vision", and available information. Based on these priorities, the resources-allocation process determines the available workforce for each project from a shared resource pool. In other words 'integration' influences the modelled program by regulating the information flow between projects and by affecting the priority of each project. Moreover, the equations within these modules contain the functions, decision rules and assumptions that define the model.

4. Model simulations and analysis

In order to analyze and understand the impact of PM and SE integration, a systematic approach for the simulation of programs with different degree of integration was needed. For this purpose, sensitivity simulations analysis was applied. These sensitivity simulations examine the following four hypotheses:

- H1 - 'integration' leads to better program performance
- H2 - Some organizational, people-related, and process-related factors have more influence on performance than others
- H3 - Integrated enterprises can deal better with changes than non-integrated enterprises
- H4 - The influence of integration depends on the program properties

Since integration between PM and SE is characterized by a number of organizational, processes-related, and people-related factors, the model specifies an "integration indicator" which measures integration between PM and SE as a function of the aggregated factors (see Figure 1). Moreover, time performance is measured as the deadline overrun relative to the initial deadline. Program costs are represented as the cumulative man-effort. Non-project work is not accounted for in this variable. Lastly, the base case is a program that comprises 4000 activities (scope) and has a planned duration of 32 months. The modeled program does not represent a specific program but a generic program with properties calibrated to reflect the median of the 157 programs in the survey from Reiner [2015].

4.1 Framework

The sensitivity analysis method runs a high number of simulations with random values within a defined range as inputs for the variables to examine, named the active parameters. In this case, the factors that are being varied are indicated as inputs (active parameters) for each specific analysis and have random values between “0” and “1” (in 0.1 increments), since they are normalized. As shown in Table 1 these simulations analyze program performance against integration (Group 1), factors that contribute to integration (Groups 2), and integration under altered program properties (Group 3).

Table 1. Simulations framework

GROUP	HYPOTHESIS TESTED	INPUT PARAMETERS	PROGRAM PROPERTIES	OUTPUT
1	H1	Factors that contribute to integration	Base Case	'integration' (indicator) vs. performance
2	H2	Factors (individually and aggregated)	Base Case	'integration'-related factors vs. performance
3	H3 and H4	Factors that contribute to integration	Variable scope, initial deadline, and amount of changes	'integration' (indicator) vs. performance vs. program property

For the groups 1 and 3 all integration-related factors (see Figure 1) are variated randomly each simulation, thus a new integration indicator emerges each run. In contrast, for each simulation within group 2 one of the factors is variated while the others have constant values in order to assess the impact of the factors individually. Additionally, the factors were variated in groups (as in the aggregated factors in Figure 1) to examine interactions among them.

Moreover, the program properties correspond the base case for groups 1 and 2. Since the third group examines integration under altered program properties, the scope and initial deadline are variable, as well as the number of changes. The sensitivity simulations were carried out with an embedded tool in Vensim® with Monte Carlo simulations. Each of the analysis comprised 10000 runs, i.e. 10000 simulated programs[3]. As shown in Figure 1, the integration indicator comprises the three aggregated factors. Each of them consists of a number of weighted sub-factors, which weights add to 100%. Thus, the aggregated factors have values between 0 and 1 as well. Since the integration indicator is the sum of these three factors it is a continuous number between 0 and 3, being 3 the highest integration level.

4.2 Simulation results

The following Figure 4 presents the results for the first group. The diagram plots the deadline overrun (in %) relative to the initially planned deadline against the value of the integration indicator. In the diagram a cluster of data points appears at around 200% overrun since the simulation is interrupted at this time point. It is assumed that in real life most of these programs would be cancelled before reaching that magnitude of delay. These data points show a trend to a declining time overrun with higher integration. Moreover the with increasing value of the integration indicator the data points are less dispersed, indicating that performance is more homogenous with higher PM and SE integration.

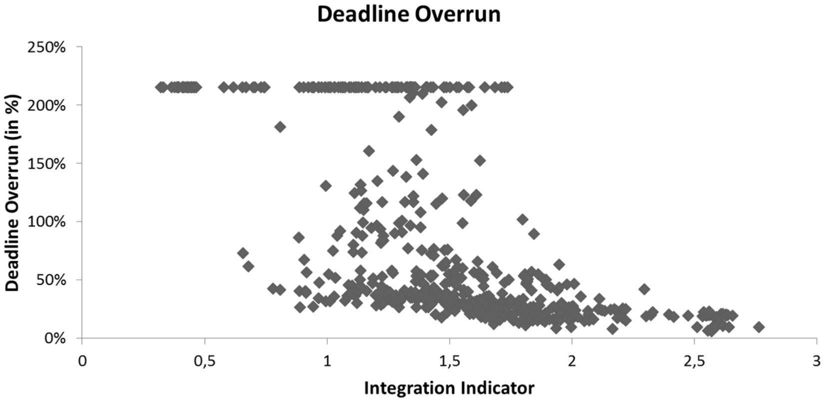

Figure 4. Deadline overrun to 'integration'

Furthermore, Figure 5 provides further support for these observations. As shown, the deadline overrun for the base program is between 50% up to 200% for very low integration (integration indicator <1), while for highly integrated (integration indicator >2) programs it is between 0% and 50%. This behavior can be observed with other performance indicators such as costs.

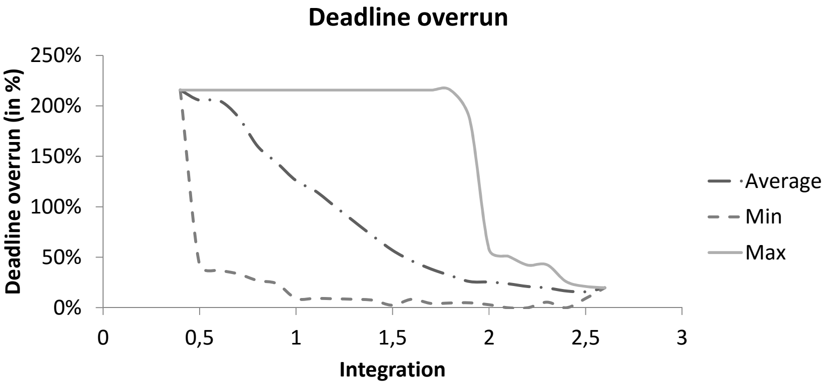

Figure 5. Deadline overrun versus ' integration' (average, min. and max. values)

---

[3] For illustration purposes the simulation depicted in Figure 4 contains only 500 runs.

Furthermore, the third group of simulations comprises the analysis of the influence of integration on programs with different initial conditions. The variable program properties analyzed are:

- variable scope with proportional initial deadline and amount of changes;
- variable initial deadline with constant scope / amount of changes;
- variable amount of changes with constant deadline / scope.

The most explicit results from comparing the curves were obtained in the analysis of variable initial deadlines. The three scenarios illustrated in Figure 6 are named "no external schedule pressure", "high external schedule pressure" and "very high external schedule pressure". External schedule pressure represents the pressure to complete the program from diverse stakeholders at the program definition, thus it is assumed to have only an impact on the initial deadline. With higher time pressure the planned time is shorter by constant scope.

In this case, the Base Case's initial deadline was 32 weeks, while the deadlines of high pressure and the very high pressure were 25.6 and 19.2 weeks respectively. These deadlines were calculated through a linear extrapolation with the help of existing project models. Within this model, the deadline defines the available time for each sub-project and has an effect on the overtime and haste variables - as in traditional SD project-models - thus on the error rate, as well as on the resource allocation. Generally, the shorter the remaining time, the more resources are allocated to a sub-project, whereas this relationship is non-linear and depends on integration-related factors.

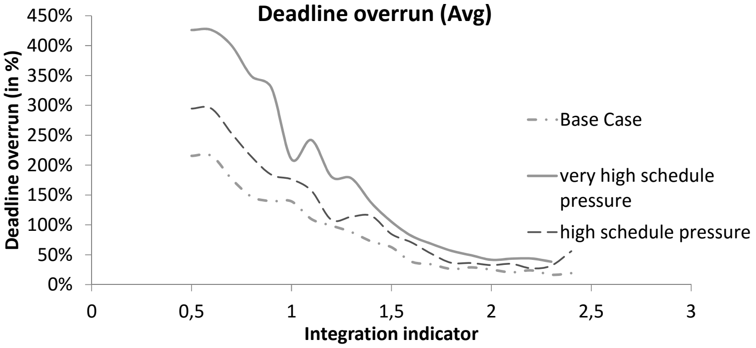

Figure 6. Deadline overrun versus 'integration' with variable initial deadline

In Figure 6, the curves for “high” and “very high” pressure show greater deadline overruns than the base case for low integration programs, while the overrun for high integrated ones is almost the same in all three cases. These trends suggest that in the low integration cases the schedule pressure has a higher impact on the schedule overrun. Thus, these results indicate that for programs with high external schedule pressure, i.e. a tight deadline, 'integration' plays a major role in program performance.

4.3 Discussion of results

To summarize the results, Table 2 presents an overview of the hypotheses and the results of the corresponding analysis (cf. Table 1). These results provide new insights on the influence of PM and SE integration on program performance. One of the main results of the study is shown in Figure 4. In contrast to the first hypothesis ('Integration' leads to better program performance), programs with a medium level of integration may achieve results comparable to 'high-integrated ones'. However, some of these programs with a medium level of integration resulted in deadlines overruns of over 100%. Thus, programs with higher integration appear to not necessarily have less deadline overruns but they present a more reliable performance. Moreover, the 'effective communication' and the 'quality of information exchange' variables show a positive correlation with better performance. Additionally, the 'quality of information exchange' shows a tipping point around 40% of its maximum value. This indicates that a minimum quality of the information transmission is necessary to maintain the information flow. This is a product of a reinforcing loop that characterizes how errors within the information provoke errors in the work being done, which in turn provoke more errors. When the information transmission quality is below the threshold value mentioned above, the program enters in a deadlock of low information quality and low work quality.

Table 2. Hypotheses and results summary

HYPOTHESIS	RESULTS
'Integration' leads to better program performance (H1)	Simulations indicate that 'integration' leads to more reliable performance
Some of the identified factors have more influence on performance and 'integration' than others (H2)	6 individual factors appear to have greater influence; further analysis is needed. Simulations suggest a critical minimum value for aggregated factor 'quality of information exchange'
Integrated enterprises can deal better with change than non-integrated (H3)	No results suggest that the influence of 'integration' is greater in programs with more changes. Further analysis is needed.
The influence of 'integration' depends on program properties (H4)	Results suggest that integration is especially relevant for programs with high schedule pressure.

In addition, the following factors were identified to have more impact on program completion time and effort: transparency, trustful environment, use of standards, interdisciplinary skills, accountability, and shared vision. However, the factors’ weights in the model, which are input from [Reiner 2015], have a strong influence on this analysis, thus the implication of these results is limited. Additional statistical methods are needed in future research to measure the impact of individual factors with help of the model. Finally, the results suggest that the initial deadline has a great influence on the impact of 'integration' on time performance. The implications of these findings for future research are discussed in the outlook.

5. Conclusion and outlook

The study presented here has two main contributions: the SD model itself and the simulation results based on the model analysis. The model’s value is realized through explicitly representing the mental models, hypotheses and assumptions of the different cited researchers and experts on how programs function especially regarding PM and SE integration. Therefore, the model (1) enhances the understanding of the big picture by untangling the complex system into simpler parts and relations and (2) serves as foundation for discussions among researchers and practitioners. Furthermore, the analysis presented examines the impact of the level of integration between PM and SE in a program. The main outcomes of the simulations' results are that, although more integrated programs perform better on average, a higher level of integration does not necessarily mean less deadline and cost overruns and less remaining work – but it does lead to more homogeneous program performance and hence reducing planning risks. Therefore the results suggest that integration leads to more reliable program performance Based on this result a new hypothesis can be introduced, namely: integration between PM and SE leads to more reliable program performance. Although this hypothesis should be tested with empirical work, it contributes to the understanding of the consequences of low integration between PM and SE.

Further simulations examined the impact of integration in programs with altered external conditions or constraints. The objective was to investigate if 'integration' is more critical under certain environmental conditions. As concluded above, 'integration' seems to be decisive in programs with high time pressure. Therefore, one of the results of the analysis was that the planned deadline has a great impact. Hence, the influence of integration on the program plan should be analyzed in the future by expanding the model boundaries to include the program definition process. Additionally, effective recommendations for management actions to improve 'integration' can be developed with the insights provided. They can be integrated into the model directly linked to the factors that they engage in or to the program behavior’s leverages. Finally, through examining and modeling the secondary effects of these recommendations, their overall impact can be assessed.

Until now, key insights for SE and PM practitioners in regard of PM and SE integration are that: (1) efforts in integrating PM and SE probably lead to better planning security, (2) especial attention in regard to 'integration' is required for projects with a tight deadline, and (3) a minimum quality of information transmission between disciplines should be ensured through standards and interdisciplinary training. In conclusion, the general objective of this work, namely to understand how the relationship between PM and SE integration and program performance functions, was largely fulfilled. The model presented in this paper is a step towards developing effective recommendations that lead to better performing programs based on the new knowledge and insights.

References

Browning, T. R., Fricke, E., Negele, H., "Key concepts in modeling product development processes", Systems Engineering, Vol.9, No.2, 2006, pp. 104-128.

Conforto, E., Rebentisch, E., "Improving the integration of program management and systems engineering – case study analysis", Executive report. Cambridge, USA, 2014.

Conforto, E., Rebentisch, E., Rossi, M., "Improving the integration of program management and systems engineering – case study analysis", Executive report. Cambridge, USA, 2013.

Conforto, E., Rossi, M., Rebentisch, E., Oehmen, J., Pacenza, M., "Improving integration of program management and systems engineering", 23rd incose annual international symposium. Philadelphia, USA, <http://hdl.handle.net/1721.1/79681>, 2013.

Cooper, K. G., "Naval ship production: A claim settled and a framework built", Interfaces, 10(6):20-36, 1980.

Cooper, K. G., "The rework cycle: Benchmarks for the Project Manager", Project Management Institute, 1993a.

Cooper, K. G., "The rework cycle: How it really works… and reworks", PM network, 1993b, pp. 25-28.

Cooper, K. G., "The rework cycle: Why projects are mismanaged", PM network, 7 (2), 1993c, pp. 5-7.

Ferris, T. L., "Systems engineering process standardization and cultural diversity", In: Engineering Management Conference, IEEE, 2007, pp. 90-95.

Ford, D. N., "System dynamics for large complex projects", Cambridge, USA, 2009.

Ford, D. N., "The Dynamics of Project Management: An Investigation of the Impacts of Projects Process and Coordination on Performance", Massachusetts Institute of Technology, <https://dspace.mit.edu/handle/1721.1/11105>, 1995.

Forrester, J. W., "Industrial dynamics: MIT Press", Cambridge, MA, 1961.

Herweg, G. M., Pilon, K. E., "System Dynamics Modeling for the Exploration of Manpower Project Staffing Decisions in the Context of a Multi-ProjectEnterprise", MIT SM Thesis, February 2001, <https://dspace.mit.edu/handle/1721.1/82695>, 2001.

INCOSE, "Incose Systems Engineering Handbook V", 3.2. 2. INCOSE, 2011.

Kohn, A., "Entwicklung Einer Wissensbasis Für Die Arbeit Mit Produktmodellen (Knowledge Base for Supporting the Handling of Product Models in Engineering Design)", Doctoral Dissertation. TUM, 2014.

Langley, M., Robitaille, S., "Toward a new mindset: Bridging the gap between program management and systems engineering", PM Network, 2011.

Locatelli, G., Mancini, M., Romano, E., "Systems engineering to improve the governance in complex project environments", International Journal of Project Management, Vol.32, No.8, 2014, pp. 1395-1410.

Lucae, S., Rebentisch, E., Oehmen, J., "Understanding the front-end of large-scale engineering programs", Procedia Computer Science, Vol.28, 2014, pp. 653-662.

Lyneis, J. M., Cooper, K. G., Els, S. A., "Strategic management of complex projects: A case study using system dynamics", System Dynamics Review, Vol.17, No.3, 2001, pp. 237-260.

Lyneis, J. M., Ford, D. N., "System dynamics applied to project management: A survey, assessment, and directions for future research", System Dynamics Review, Vol.23, No.2-3, 2007, pp. 157-189.

Oehmen, J., Rebentisch, E., Kinscher, K., "Program management for large scale engineering programs", LAI Whitepaper Series "Lean Product Development for Practitioners", 2010.

PMAJ, "P2m: A guidebook for project and program management for enterprise innovation", Tokio, 2008.

PMI (Project Management Institute), "The standard for program management", 3rd ed., 2013.

Reiner, T., "Determination of Factors to Measure the Effective Integration between Program Management and Systems Engineering", Master's Thesis, RWTH Aachen, 2015.

Sage, A. P., Rouse, W. B., "Handbook of systems engineering and management", John Wiley & Sons, 2009.

Sterman, J. D., "Business dynamics: Systems thinking and modeling for a complex world", Irwin/McGraw-Hill Boston, 2000.

Sterman, J. D., "System dynamics modeling for project management", Massachusetts Institute of Technology, Cambridge, USA, 1992.

Lucia Becerril, Research Assistant
Technical University of Munich, Institute of Product Development
Boltzmannstraße 15, 85748 Munich, Germany
Email: lucia.becerril@pe.mw.tum.de