Analysing the effect of lean on the performance of NPD projects using system dynamics modelling

Saeed Taheri (saeed.taheri@ntu.ac.uk)
Nottingham Business School

Baback Yazdani
Nottingham Business School

Michael Zhang
Nottingham Business School

Abstract

To be able to survive in today's fast-changing market environment companies are looking for innovative ways to improve the performance of their new product development (NPD) processes. However, uncertainty and rework are among characteristics of NPD which make them difficult to manage. Implementing lean in NPD is an innovative approach to address this issue. Using system dynamics approach to model set-based concurrent engineering as a fundamental element of lean product development, this paper shows the positive effect on of adopting this strategy on the time, cost and quality of NPD projects, in comparison with the traditional point-based design.

Keywords: Lean, SBCE, System dynamics

Introduction

New product development (NPD) projects are the main source of competitive advantage for companies as successful firms have the capability to introduce high quality products at a faster rate and lower cost than their competitors. The main motive for studying NPD from the process perspective is to find a linkage between the process structure and its behaviour which affects overall performance of the system. According to new perspectives, improving the process quality is in direct relationship with the time and cost improvement in NPD processes (Suss and Thomson, 2012). Reducing defects and reworks by better sharing, acquiring, integrating, and applying knowledge in NPD processes leads to quality improvement.

Ahmadi and Wang (1999) defined the design confidence as the degree of compatibility and closeness between the design and its target specifications. As the design process progresses the level of design confidence increases due to producing more information about the design and the knowledge gained from this information. This results in reducing uncertainty in the process. The progress of design is accompanied by iteration cycles as inherent parts of NPD processes (Lévárdy and Browning, 2009). The purpose of these deliberate iteration cycles could be experimentation (Kennedy et al., 2014), information transfer and converging on a satisfactory solution (Meier et al., 2015). However, there is a distinction between these planned iteration cycles and reworks, which are caused by revisiting prior decisions due to new learnings, making technically infeasible decisions early and before having desired knowledge (Kennedy et al., 2014), and defects (Lévárdy and Browning, 2009). Using the definition by Kennedy et al. (2014: P279), rework is "the work that occurs when a prior decision that was assumed to be final for that project is changed because it was later found to be defective". It is reported that between 30-80% of development capacity in a project are consumed by reworks (Terwiesch et al., 2002; Kennedy et al., 2014). Fewer reworks result in faster time-to-market, lower costs, better quality and customers' satisfaction, and higher profitability.

Both planned iterations and reworks consume valuable resources of the project, so finding a way to reduce them results in reducing the time and cost of projects. However, as the design confidence is increased through iteration cycles, eliminating them could result in having a new product with inferior quality. Lean product development (LPD) provides an answer to this contradiction by adopting Set-Based Concurrent Engineering (SBCE). Yet, the reasons of outperforming SBCE compared with traditional Point-Based Concurrent Engineering (PBCE) approach are not well articulated in literature. It is the aim of this paper to study the ways SBCE affect the performance measures of an NPD project, including project duration, cost and process quality.

Theoretical development

Reviewing the publications in the field of LPD reveals that most of them put SBCE as its pivotal element (Sobek et al., 1999; Morgan and Liker, 2006; Hoppmann et al., 2011; Khan et al., 2013). According to Kerga et al. (2016) SBCE is one of approaches to handle uncertainty in NPD. Ford and Sobek (1995) called it a key decision strategy for NPD managers about the way to converge from a set of initial conceptual ideas to one concept to become the final design. In contrast with the traditional PBCE which includes selecting a unique concept as early as possible during the development process, SBCE is grounded on front-loading the project by considering a set of design solutions from the beginning which are gradually narrowed down by evaluating the performance parameters and eliminating inferior options which do not satisfy design requirements (Sobek et al., 1999). The effectiveness of PBCE is dependent on the ability of companies to distinguish the quality of different design alternatives and select the best one early in the design process (Schulze, 2016). However, changes in requirement during design phases are unavoidable, and reworks due to these changes increase the design cost exponentially towards the later stages of the project (Kennedy et al., 2014). Implementing SBCE has the advantage of substituting late reworks with early planned iteration cycles. Although at the first glance, delaying decisions about design solutions and considering multiple solutions seems counterintuitive, Toyota product development system proves that it reduces costly design reworks through decreasing development risks (Morgan and Liker, 2006).

To compare the effectiveness of these two approaches a number of studies are conducted using case studies and modelling as their research methodology. Case studies conducted by Sobek et al. (1999) and Morgan and Liker (2006) focused on Toyota as the origin of LPD and tried to find the benefits of SBCE through collected evidences. They found reliable communication among teams, rapid innovation in design, better decision making, and enhanced learning as the main benefits of SBCE (Kerga et al., 2016). In cases other than Toyota, Raudberget (2010) reported more than 75% decline in project cost and 50% reduction in project time, as well as more than 50% improvement in technical performance as the result of implementation of SBCE in Swedish companies. Using an analytical model, Ford and Sobek (2005) showed the improving effect of SBCE on project cost and duration. Belay et al. (2014) compared the shift in resource allocation towards the front-end of projects in SBCE with PBCE and found more than 50% reduction in project cost and 20% improvement in project duration as the result of adopting SBCE.

Nevertheless, case studies did not compare SBCE and PBCE directly and under similar project conditions, such as inputs to the projects and existing constraints. In fact, it is hardly possible to run two similar projects which just differ from the point of view of these two approaches in order to compare the outcomes. This results in inability of previous researches in proving the advantages of SBCE (Kerga et al., 2016). Yet, understanding the superiority of this approach is significantly important for managers who are willing to adopt lean product development. On the other hand, simulation studies missed to investigate some key factors in SBCE approach such as the number of initial concepts, and the type of project. Malak et al. (2009) clearly suggested additional research to investigate the partitioning of design space into concepts. Consequently, this paper aims to fill this gap by finding the answers to the following question:

- How adopting SBCE as a fundamental element of LPD helps companies to develop new products with high quality in less time and with lower investment?
- How does the type of project influence the effectiveness of adopting SBCE?

Research methods and data collection

System dynamics as a method for building simulation models to study the management of dynamically complex systems is adopted for this paper. Several researchers used system dynamics models to study the continuous progress of NPD projects and the impact of different factors on the projects performance. The power of system dynamics modelling is in making a linkage between the observable behaviour of a system, its structure and decision-making processes (Qudrat-Ullah and Seong, 2010).

This paper follows Rahmandad’s approach (2015) to build a general model based on existing theoretical construct in literature. The model is then used to test different policies after being calibrated using real project data. Historical data related to the project schedule, cost and quality problems over the entire period of four types of projects, from the complexity perspective, in a major car-manufacturing company are collected to calibrate the model. Project complexity reflects the novelty of the project. Four types of projects are selected based of the classification by Morgan and Liker (2006) as followed:

Type 1- radically new projects with breakthroughs in the products or technology

Type 2- product platform-development projects that have fundamentally new systems and components and utilize improved versions of existing products.

Type 3- derivative products which are built on existing product platforms

Type 4- incremental product improvement projects with the lowest degree of novelty

Modelling process

The model is structured around three general phases on an NPD project, namely conceptual design, and detail design, and tooling. Although the project performance is the result of interactions between many development processes, features of project, participants, and resources, as the purpose of the model is investigating the effect of SBCE on the performance of a project, only features and processes which describe this effect are included in the model. Therefore, the model outputs are relative and only useful for developing insights and increasing the understanding about SBCE.

The information-based view of NPD (Clark and Fujimoto, 1991) is followed in the model which defined each development activity as an information-processing unit which receives information from upstream activity, transforms it to the new information, and then send it to the downstream activity. The first part of the model is workflow sector, which is based on rework cycle as introduced by Lyneis and Ford (2007). In this structure tasks first reside in *tasks not completed* stock and after initial completion move to *tasks pending test* stock. A fraction of tasks is discovered faulty after test and moves to *tasks pending rework* stock. After rework, these tasks move back again to tasks pending test stock for rechecking, as rework could create additional defects. Tasks which show no defect and pass testing step are approved and move to *tasks pending decision* stock. Completion rate, rework detection rate, rework rate and approve rate are flows between stocks in the structure which their values depend on the capacity, the number of tasks available and the minimum time required to perform the job. Rework probability controls the portion of tasks which are approved or sent for rework.

The unique expansion to this model compared with previous models of NPD projects is adding the *iteration* rate to make a distinction between rework and iteration in each phase of the project. This flow connects tasks pending decision and tasks not completed stocks, so tasks could either be released to the next phase or sent back to pass the iteration cycle again. In conceptual design phase, another flow is added for concepts to be sent to the *scope reduced* stock. This is to model the convergence period in conceptual design phase during which inferior concepts are narrowed down gradually. Each iteration cycle increases the level of design confidence, and the decision made at the end of each cycle to either release tasks to the downstream phase or reiterate them depends on the level of uncertainty as the difference between the design confidence and the target confidence. Increase in design confidence is modelled as a co-flow with a single stock and flow structure. *Confidence increase* rate in conceptual design phase depends on the project complexity, the number of initial concepts and the number of iteration cycles which are completed (Eq. 1). In other phases, project complexity is substituted by the final design confidence of the final concept receives from the conceptual design phase (Eq. 2).

$$R_{DC_{CD}} = FC.I^{PI}.C^{PC}.S^{PS} \tag{1}$$

$$R_{DC} = FC.I^{PI}.{DC_{CD}}^{PD}.S^{PS} \tag{2}$$

Where $R_{DC_{CD}}$ and R_{DC} are the rates of increase in design confidence during conceptual design phase and downstream phases, I is the number of iterations, C is the number of initial concepts, S is project complexity, PI is the strength of number of iterations effect on the rate of confidence increase, PC is the strength of number of concepts effect on the rate of confidence increase, PS is the strength of project complexity effect on the rate of confidence increase, DC_{CD} is design confidence in conceptual design phase, PD is the strength of the effect of the design confidence in conceptual design phase on the rate of confidence increase in downstream phases, and FC is a scaling factor. FC, PI, PC, PS and PD are project-independent constants which their values are calculated through the calibration of the model.

Second part of the model is resource management sector which allocate recourses to completion, test and rework activities in proportion to their current demands, based on direct proportional policy (Joglekar and Ford, 2005). The capacity of performing each type of activities depends on the number of allocated resources and the productivity of resources. The productivity is defined as the amount of work each resource could handle in the unit of time, and assumes to be constant throughout the project.

The final part of the model is outputs sector. The duration and cost of single phases and the whole project, the number of iteration cycles in each phase, the rework percentage throughout the project, and the total design confidence, as an indicator of process quality, are monitored by the model. The cost metric is estimated by multiplying the total number of allocated resources to different types of activities by a unit cost parameter as a company-specific constant which is calculated through the model calibration.

Feedback structure

Fig. 1 illustrates the workflow sector of the model and its feedback structure. Iteration and rework are two reinforcing loops in the system, responsible for delay and cost overrun in projects. Rework loop withdraws tasks from their direct path after discovering defects and return them again to the tasks pending test stock after rework. This adds to the duration of the project because of multiple execution of some activities on the same task. In addition, the cost of project is increased due to allocating resources to unplanned rework activities. Similarly, iteration loop increases the time and cost of the project by increasing the number of tasks in tasks not completed stock and repeating completion, testing and rework activities which consume resources.

Feedback loop B1 is directly influenced by iteration loop; first iteration loop has the dominance which increases the time and cost of the project, but when the uncertainty decreases as the result of the increase in design confidence, dominance shifts to feedback loop B1. This loop is balancing which withdraws tasks from tasks pending decision stock by increasing the release rate, and drives the project to completion. In addition, reducing uncertainty triggers feedback loop B2 and results in a decline in rework probability, decreasing the fraction of tasks which goes through the rework loop.

Iteration loop plays a critical role in this feedback structure by triggering B1 and B2 feedback loops. However, due to the initial increase in the time and cost of the project, exogenous manipulation of the dominance of this loop by managers results in unexpected effects on the overall project performance. This manipulation is represented as the schedule pressure feedback loop in Fig. 1. Approaching to a fixed deadline and increase in the number of tasks in tasks pending completion stock due to the reinforcing effect of iteration loop increases the time required to complete the project, and results in the schedule pressure. While this could have several negative side effects, changing the target confidence in conceptual design phase is a firefighting remedy by managers to meet the deadlines. Consequently, the uncertainty reduces with lower number of iterations and feedback loop B1 dominates and finishes the project sooner and with lower cost. This shifts the policy adopted in the project from SBCE to PBCE.

Model testing

Using standard system dynamics validation tests, as introduced by Sterman (2000), the model is validated for its usefulness for the intended purpose. The model is structurally valid because it is mostly based on previously tested models, such as Jalili and Ford (2016), and literature about lean product development, concurrent engineering and SBCE. In addition, there is unit consistency in all sectors of the model. The behaviour validity of the model is tested using extreme condition tests through setting model inputs to extreme values. Under these conditions, the model still shows reasonable behaviours. Finally, consistent with the actual project behaviour described in other researches, in each iteration cycle, model shows an S-shaped trajectory of tasks approved over time which supports its behaviour validity.

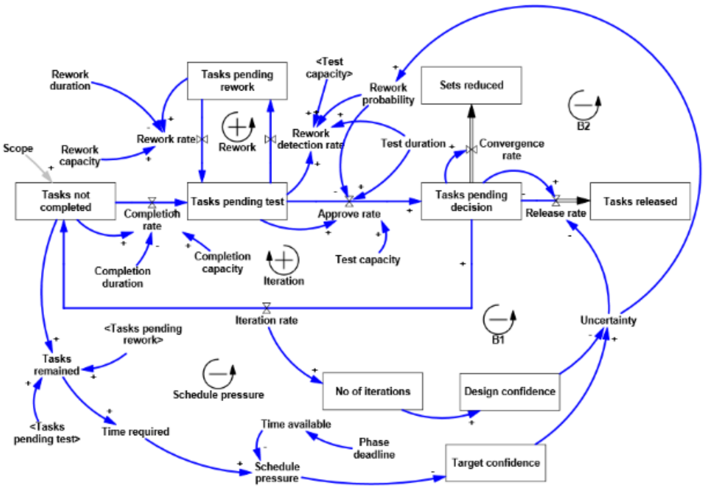

Fig. 1: the workflow sector of the model and feedback structure

Model calibration

Calibration is the process of estimating the values of model parameters in a way that allows the model to generate behaviours which fit the real-world data. Mathematically, the calibration is a numerical optimization process to minimize the distance between the model outcome and the actual data by searching for the best model parameters (Parvan et al., 2015). Model parameters are divided into two groups of project-dependent parameters which change from one project to the other, and project-independent parameters which are constant among all types of projects. The payoff function is a linear combination of the differences between projects data and model outcomes in phase finish time, total cost, and rework curve as three sources of error across all three phases of the project. It is defined as the weighted sum of the squared percentage errors (SSPE) according to the concept of the least square method (Parvan et al. 2015) as shown in Eq. 3:

$$Payoff\ function$$
$$= \sum_{ij} W_{tp} \left(\frac{TP_{ij} - tp_{ij}}{t_{ij}}\right)^2 + \sum_i W_{co} \left(\frac{CO_i - co_i}{co_i}\right)^2 + \sum_i W_r \int_0^{T_i} \left(\frac{R(t)_i - r(t)_i}{r(t)_i}\right)^2 \qquad (3)$$

Where TP and tp are simulated and actual phase finish time, CO and co are simulated and actual project total cost, R(t) and r(t) are simulated and actual rework curve, i and j are project index and phase index, and W is weight values. The weight values (W_{tp}, W_{co} and W_r) balance the numerical sizes of different variables, and their values are based on the confidence in the data. Different combinations of weight values are tried and results are compared to have the least payoff value and consequently, the minimum calibration error. The best results achieve with W_{tp}=0.5, W_{co}=0.5 and W_r=0.25.

To perform the calibration, projects are linked together through project-independent parameters using subscripts in Vensim. In the first run, the model is calibrated for each project separately to provide an estimation for parameters. Then, using these estimated parameters, the model is fine-tuned simultaneously for all projects. As there is limited access to projects data this process is used to increase the statistical power in estimation of parameters and to result in a lower final error values (Parvan et al., 2015). Two groups of model parameters and their estimated values are shown in Table 1. Fig. 2 is an example of the model outputs after calibration for project A, in comparison with the actual project data, which shows a close fit between the actual and simulated behaviours. This supports the ability of the model to investigate the dynamic of SBCE in single NPD projects.

Table 1: model parameters and their estimated values

	Parameter	Description	Phases	Project A	Project B	Project C	Project D
Project-dependent parameters	**SC**	Phase Scope	**Conceptual design**	100	50	50	50
			Detail design	96.0959	83.27	83.26	81.2057
			Tooling	149.999	107.9	106	99.766
	IRes	Initial available resources	**Conceptual design**	500	500	500	500
			Detail design	1098.51	999.2	362.9	301.5
			Tooling	822.559	504	284.3	284
	IPR	Initial rework probability	**Conceptual design**	0.432113	0.43	0.2997	0.289241
			Detail design	0.25255	0.25	0.1	0.099964
			Tooling	0.610349	0.55	0.3995	0.389761
	ICom	Initial completion duration	**Conceptual design**	0.2	0.1794	0.1656	0.127973
			Detail design	0.0247927	0.02479	0.02478	0.0247
			Tooling	0.026279	0.02989	0.02989	0.0297353
Project-independent parameters	**FC**	Scaling factor in design confidence equation	16.1302				
	PC	Strength of number of concepts effect on the rate of confidence increase	0.556888995				
	PD	The strength of the effect of the design confidence in conceptual design phase on the rate of confidence	2				
	PG	The strength of the effect of design confidence on rework probability	4.89721				
	PI	The strength of number of iterations effect on the rate of confidence increase	1.5				
	PS	The strength of project scalability effect on the rate of confidence increase	-3.12565				
	F1	Coefficient in corruption probability equation	0.001				
	Unit cost	For estimation of the total project cost	53572.6				

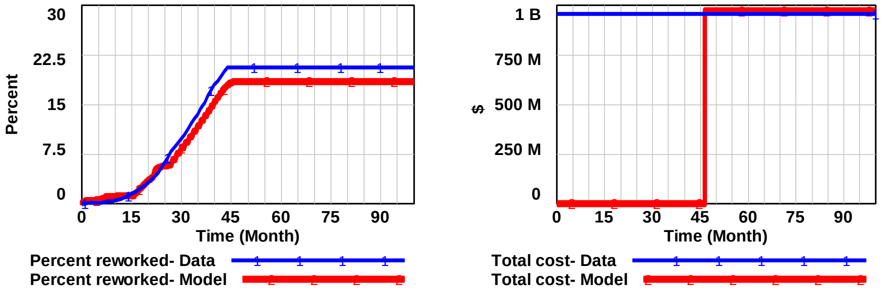

Fig. 2: simulation results for cost and rework in project A (calibration vs. real data)

Results and discussion

Model is run based on two scenarios; in PBCE scenario, conceptual design phase starts with two initial concepts and after two iterations one of them is selected as the best alternative to be sent for detail design. For this scenario schedule pressure loop is active, through which the 100 percent target confidence level is reduced when the schedule pressure is high. This results in less iteration cycles and elimination of the convergence period. In SBCE scenario, conceptual design phase starts with design teams concurrently working on five initial concepts. The convergence period starts when the level of confidence reaches to half of the target confidence when teams narrow down the design space by eliminating inferior concepts, while still iterating to increase the confidence in remaining concepts to the target level. In this scenario, the schedule pressure loop is deactivated and the target confidence is fixed at 100 percent, so iteration cycles in conceptual design phase continue until design confidence exceeds target confidence, and uncertainty approaches to zero. Fig. 3 compares what is happening in tasks pending decision stock in project A for PBCE scenario and SBCE scenario. Whereas in SBCE scenario, the convergence period starts after three iterations and lasts for four iterations, in PBCE scenario the convergence period is absent and the conceptual design phase lasts for only two iterations.

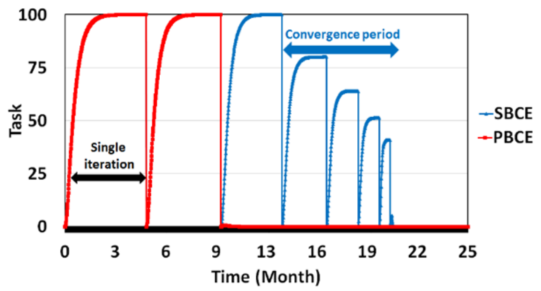

Fig. 3: difference in conceptual design phase iterations for PBCE and SBCE scenarios

Fig. 4 shows the duration and cost of conceptual design phase for two scenarios. Because of reduction in the number of iteration cycles, the duration of conceptual design phase decreases from SBCE scenario to PBCE scenario. In addition, more initial concepts results in higher rate of increase in design confidence per iteration cycle and reduce the time takes to eliminate the uncertainty. This decrease is more than 67% for project A, and about 50% for project D, while for projects B and C is in between. In addition to the duration, as the number of initial concepts is higher for SBCE scenario, the number of design teams work in parallel on these concepts increase. As the result, the cost of conceptual design phase increases more than 90 percent for SBCE scenario compared to PBCE scenario in project A. There is a similar trend for projects B, C and D while the amount of increase is 70, 60 and 37 percent, respectively. These findings are supported by Malak et al. (2009).

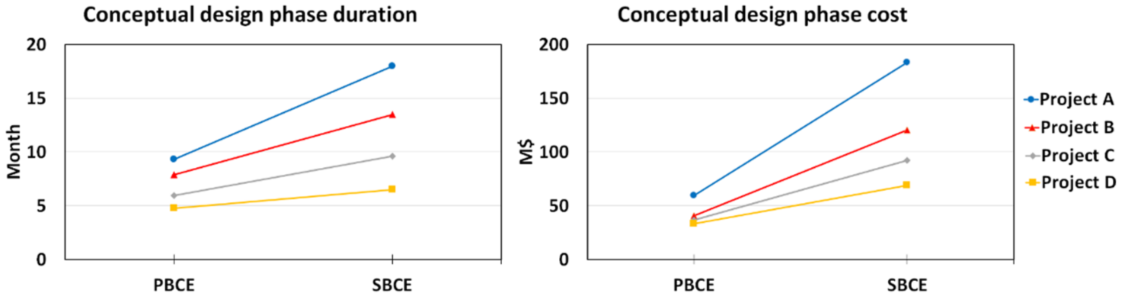

Fig. 4: the duration and cost of conceptual design phase for two scenarios

However, the total project duration and cost follow a different trajectory. According to Fig. 5, for all projects, total project duration and cost is lower in SBCE scenario. The decrease in project duration and cost is more dramatic for project A (more than 86% lower in duration and around 81% in cost), but the slope decreases towards project D where both cost and duration are 51% lower for SBCE scenario compared with PBCE scenario.

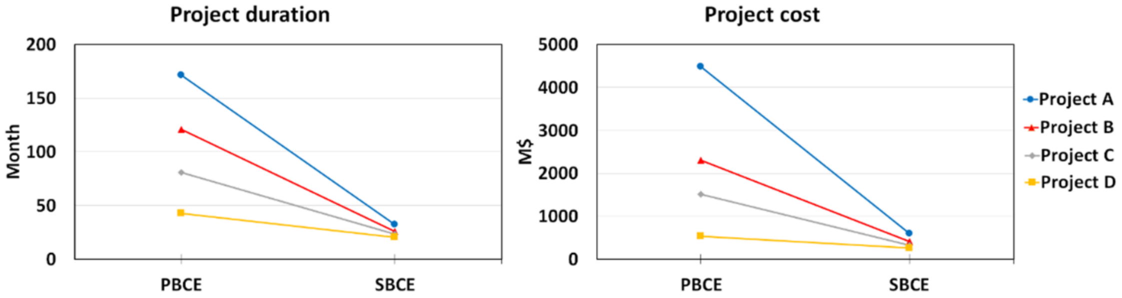

Fig. 5: effect of reducing iterations in conceptual design phase on project duration and cost

Although in SBCE scenario the cost of conceptual design phase is higher, the effect of the frontloading of projects prevents costly rework cycles during the execution of downstream phases, which in turn, reduces the total cost of projects. Likewise, the rate of increase in the design confidence is higher in SBCE scenario compared to PBCE scenario. It is due to the higher confidence in the final concept which passes more iteration cycles during conceptual design and its effect on the rate of increase in design confidence according to Eq. 2. Consequently, the detail design and tooling phases of all projects in SBCE scenario finish with less number of iterations. The iteration loop, as previously mentioned, is a reinforcing loop in the system which is one of the reasons for project delay and cost overrun, so decreasing the total number of iterations results in the reduction of project duration in SBCE scenario. However, the final design confidence as an indicator of the quality of new product is significantly higher in SBCE scenario, due to higher rate of confidence increase in each iteration during detail design (Fig. 6). Similar to the iteration loop, the rework loop is also reinforcing (Fig. 1). As shown in Fig. 6, the percentage of tasks which need rework is much lower in SBCE scenario, compared to PBCE scenario. In addition to less number of iteration cycles, this is another reason for the decrease in the duration and cost of projects in SBCE scenario. Finally, comparing the outcomes of two scenarios reveals that although implementation of SBCE has positive effects on the project performance measures for all projects, the effects are more distinguishable when the level of project complexity is higher (Project A compared with project D).

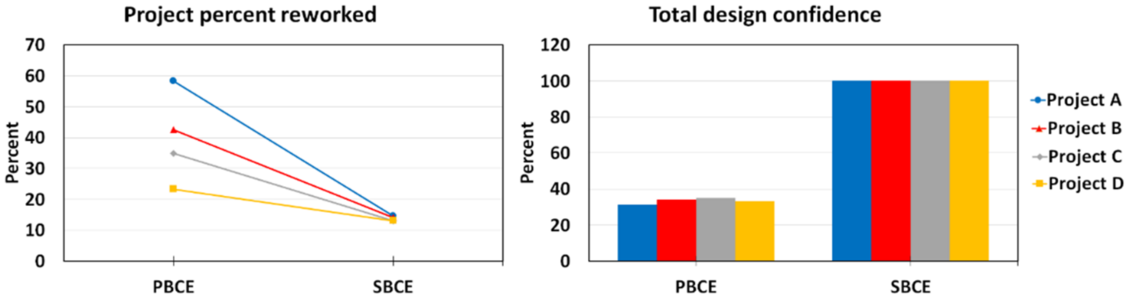

Fig. 6 : differences in the rework percentage and final design confidence for two scenarios

Conclusion

The rate of implementation of SBCE approach in companies is low (Kerga et al., 2016), mainly due to the lack of clear evidence about its superiority over PBCE approach. For this reason, in this paper a system dynamics model is developed based on the rework cycle which uniquely distinguished between rework and iteration loops in different phases of the project. SBCE and PBCE scenarios are differentiated based on the duration of conceptual design phase, the number of initial concepts and the existence of convergence period to eliminate inferior concepts. Iteration loops are responsible for increasing the design confidence towards a target level and reworks are to correct defects found in activities. Although both loops increase the cost and duration of the project, having more iteration loops in conceptual design phase increases the confidence in the final concept and results in higher increase rate of design confidence during subsequent phases. Thus, total number of iterations in the project declines while the final design confidence is significantly higher, results in the higher quality of final product. In addition, the rework probability is lower when the confidence in the final concept is high, reducing the number of tasks flow through the rework cycle.

References

Ahmadi, R., and Wang, R.H., 1999. Managing development risk in product design processes. *Operations Research*, 47 (2), 235-246.

Belay, A.M., Welo, T. and Helo, P., 2014. Approaching lean product development using system dynamics: investigating front-load effects. *Advances in Manufacturing*, 2 (2), 130-140.

Browning, T., Deyst, J., Eppinger, S. and Whitney, D., 2002. Adding value in product development by creating information and reducing risk. *IEEE Transactions on Engineering Management*, 49 (4), 443-458.

Clark, K.B., and Fujimoto, T., 1991. *Product development performance: Strategy, organization, and management in the world auto industry.* First ed. Boston: Harvard Business School Press.

Ford, D.N., and Sobek, D.K., 2005. Adapting real options to new product development by modeling the Second Toyota Paradox. *IEEE Transactions on Engineering Management*, 52 (2), 175-185.

Hoppmann, J., Rebentisch, E., Dombrowski, U. and Zahn, T., 2011. A framework for organizing lean product development. *EMJ - Engineering Management Journal*, 23 (1), 3-15.

Jalili, Y., and Ford, D.N., 2016. Quantifying the impacts of rework, schedule pressure, and ripple effect loops on project schedule performance. *System Dynamics Review*, 32 (1), 82-96.

Joglekar, N.R., and Ford, D.N., 2005. Product development resource allocation with foresight. *European Journal of Operational Research*, 160 (1), 72-87.

Kennedy, B.M., Sobek, D.K. and Kennedy, M.N., 2014. Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process. *Systems Engineering*, 17 (3), 278-296.

Kerga, E., Schmid, R., Rebentisch, E. and Terzi, S., 2016. Modeling the benefits of frontloading and knowledge reuse in lean product development. *In: Proceedings of PICMET '16: Technology Management for Social Innovation, Portland, US, 2016.* pp. 2532-2542.

Khan, M.S., Al-Ashaab, A., Shehab, E., Haque, B., Ewers, P., Sorli, M. and Sopelana, A., 2013. Towards lean product and process development. *International Journal of Computer Integrated Manufacturing*, 26 (12), 1105-1116.

Lévárdy, V., and Browning, T.R., 2009. An adaptive process model to support product development project management. *IEEE Transactions on Engineering Management*, 56 (4), 600-620.

Lin, J., Chai, K.H., San Wong, Y. and Brombacher, A.C., 2008. A dynamic model for managing overlapped iterative product development. *European Journal of Operational Research*, 185 (1), 378-392.

Lyneis, J.M., and Ford, D.N., 2007. System dynamics applied to project management: a survey, assessment, and directions for future research. *System Dynamics Review*, 23 (2-3), 157-189.

Malak, R.J., Aughenbaugh, J.M. and Paredis, C.J., 2009. Multi-attribute utility analysis in set-based conceptual design. *Computer-Aided Design*, 41 (3), 214-227.

Meier, C., Browning, T.R., Yassine, A.A. and Walter, U., 2015. The Cost of Speed: Work Policies for Crashing and Overlapping in Product Development Projects. *IEEE Transactions on Engineering Management*, 62 (2), 237-255.

Morgan, J.M., and Liker, J.K., 2006. *The Toyota product development system: integrating people, process, and technology.* New York: Productivity Press.

Parvan, K., Rahmandad, H. and Haghani, A., 2015. Inter-phase feedbacks in construction projects. *Journal of Operations Management*, 39, 48-62.

Qudrat-Ullah, H., and Seong, B.S., 2010. How to do structural validity of a system dynamics type simulation model: the case of an energy policy model. *Energy Policy*, 38 (5), 2216-2224.

Rahmandad, H., 2015. Connecting strategy and system dynamics: an example and lessons learned. *System Dynamics Review*, 31 (3), 149-172.

Raudberget, D., 2010. Practical applications of set-based concurrent engineering in industry. *Journal of Mechanical Engineering*, 56 (11), 685-695.

Schulze, A., 2016. Developing products with set-based design: How to set up an idea portfolio and a team organization to establish design feasibility. *In: Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2016.* Cambridge University Press, pp. 235-249.

Sobek, D.K., Ward, A.C. and Liker, J.K., 1999. Toyota's principles of set-based concurrent engineering. *MIT Sloan Management Review*, 40 (2), 67-84.

Sterman, J.D., 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World.* 1st ed. Boston: McGrow Hill Higher Education.

Suss, S., and Thomson, V., 2012. Optimal design processes under uncertainty and reciprocal dependency. *Journal of Engineering Design*, 23 (10-11), 829-851.

Terwiesch, C., Loch, C.H. and Meyer, A.D., 2002. Exchanging preliminary information in concurrent engineering: Alternative coordination strategies. *Organization Science*, 13 (4), 402-419.