We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Using Discrete Event Simulation for Evaluating Engineering Change Management Decisions

Weilin Li

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/50579

1. Introduction

Today's hyper-competitive worldwide market, turbulent environment, demanding customers, and diverse technological advancements force any corporations who develop new products to look into all the possible areas of improvement in the entire product lifecycle management process. One of the areas facing both practitioners and scholars that have been overlooked in the past is Engineering Change Management (ECM).

On the one hand, even though the demand has increased for more effective ECM as an important competitive advantage of product development companies, the existing ECM literature focuses mainly on the following topics: i) administrative evaluation that supports the formal EC approval, implementation, and documentation process, ii) ECM in product structure and material resource planning, and iii) change propagation and knowledge management. In addition, with a few exceptions [1, 2, 4, 12, 18, 19, 20, 26], almost all the previous research or empirical studies were qualitatively discussed in a descriptive nature.

On the other hand, despite of a rich body of concurrent engineering literature that emphasizes the iterative nature of New Product Development (NPD) process, "these models see iterations as *exogenous* and *probabilistic*, and do not consider the source of iteration" [23], which causes the identified rework too general, and therefore not sufficient for an effective ECM study. As a result, there is a lack of research–based analytical models to enhance the understanding of complex interrelationships between NPD and ECM, especially from a systems perspective.

The vision behind this chapter is to ultimately bridge this gap between these two bodies of literature by recognizing the main characteristics of both New Product Development (NPD) and ECM processes, quantifying the interrelated connections among these process features in a Discrete Event Simulation (DES) model (Arena), experimenting with the model under different parameter settings and coordination policies, and finally, drawing decision-making suggestions considering EC impacts from an overall organizational viewpoint.

2. Background

2.1. Problem definition

ECM refers to a collection of procedures, tools, and guidelines for handling modifications and changes to released product design specifications or locked product scope [4, 6, 22, 35]. ECs can be classified into two main categories [4, 5, 11, 13, 27]:

- *Emergent EC* (EEC) originates from the problems or errors detected from activity outcomes (i.e., design data and information) that have already been frozen and formally released to the downstream phase. EECs are assumed to occur according to a certain probability determined by the conceptualized *solution uncertainty*,
- *Initiated EC* (IEC) requested by sources outside the project's control such as changing market conditions, arising customer requirements, new legislation, or emerging technology advances any point along the NPD process in response to the conceptualized *environmental uncertainty*.

Under this classification scheme, design iterations within an NPD process and *problem–induced* EECs are very similar, but occur in different situations. Both of them aim at correcting mistakes or solving problems through repetitively achieving unmet goals that have been set initially. EECs are requested rework to prior activities whose outcomes have already been finalized and released to the next phase. However, NPD iterations take place before any design information is formally released to downstream phases, and therefore it generally takes less time to handle iterations due to both a smaller rework scope and a shorter approval processing time. For simplicity, term "*rework*" will be used to refer to both iterations and EECs, unless specific distinction is required. From another standpoint, *opportunity–driven* IECs arise from new needs and requirements, which result in the adding of functionality to a product [10], or enlargement of the original design solution scope. A formal assessment and approval process is desirable in handling both types of ECs due to the associated complexity and potential risks [13, 35].

2.2. Context

ECM problems cannot be studied in isolation. But rather, they need to be addressed within a broader context, including the following three principle facets: i) complex systems, ii) current engineering and uncertainty, and iii) rework and change propagation.

2.2.1. Complexity

A new product is designed and developed via an NPD process through the efforts from a group of specialists under dynamic internal and external environment. This DES model brings together the four main elements of complexity associated with design and product development [10], namely, product, process, team (/designer), and environment (/user), on the decision of how iterations and ECs emerge and thus impact NPD project performance, and how should they be effectively managed by applying different coordination policies.

Highly engineered *product* is a complex assembly of interacting components [21, 25]. In automobile industry, a fairly typical modern vehicle is composed of more than ten thousand manufactured component pieces, supplied by thousands of outside suppliers. In the face of such great quantities of components, complex products are impossible to be built all at once. They are decomposed into minimally coupled major systems, and then further broken into smaller sub–systems of manageable size and complexity, and finally down to separate components or parts for individual detailed engineering design. On the other hand, the integration of interdependent decompositions within and across system(s) into the final overall solution as well adds up to the level of complexity and requires substantial coordination efforts [31].

Similarly, a large complex Product Development (PD) *process*, through which all the stages of a product's lifecycle occur, is itself a complex system involving hundreds or thousands of interrelated or interacting activities which transforms inputs into outputs. As shown in the PD literature, tremendous research effort has been devoted into exploring the complexity of PD processes, especially in studying both of the advantages and disadvantages of parallel development process (also known as concurrent engineering) or spiral development process (which is applied more often in software industry) as compared with the traditional staged (also known as waterfall or sequential) development process. Some prior research particularly stressed structuring and managing the process through efforts in minimizing the interdependencies among tasks via process sequencing optimization [8, 9, 34].

Also, multi–disciplinary *teams* participating in an NPD project are typically composed of numerous decision makers from different functional areas (e.g., marketing, engineering, manufacturing, purchasing, quality assurance, etc.) with varied skill sets (e.g., degree of specialization, depth of knowledge, qualifications, work experience, etc.), responsibilities, and authorities working together and contributing to the achievement of the final product solution. These teams exhibit another set of complex and non–linear organizational behaviors in communication, collaboration, and integration when considering local task decisions as well as task interactions in determining aggregate system performance [28].

Last but not least, an NPD project interacts with its internal (e.g., simultaneous concurrent development of other products within the same organization) and external (e.g., customers/market, competitors, suppliers, and other socio–economic factors such as government regulations, etc.) *environments* throughout the project cycle. The dynamic and sometimes even chaotic competitive environmental factors also contribute significantly to the complexity in the coordination of NPD projects.

2.2. Concurrency and uncertainty

The concept of *concurrent engineering* is characterized by i) the execution of PD tasks concurrently and iteratively, and ii) the cross–functional integration through improved coordination and incremental information sharing among participating groups. It has been widely embraced by both academia and industry for the well documented advantages of NPD cycle acceleration, risk minimization by the detections of design errors in early stages,

and overall quality improvement (e.g., [3, 17, 27]). It is one of the process features that are captured and thoroughly analyzed by the DES model proposed here.

Complexity drives *uncertainty*. Uncertainty is an inherent nature of NPD projects stemming from all aspects of complexity associated with efforts creating a new product as discussed above. The presence of inherent uncertainty in NPD processes is much greater and, interestingly, much more complicated than those in processes of other kinds (e.g., business or manufacturing processes), even though the latter also possess certain degree of inherent unpredictability. Types of uncertainty in engineering design include *subjective uncertainty* derived from incomplete information, and *objective uncertainty* associated with environment [37]. Moreover, concurrent processing of NPD activities will further increase the uncertainty of an NPD project by starting activities with incomplete or missing input information. In this model uncertainty is explicitly differentiated into three types: i) low–level *activity uncertainty* represented by the stochastic activity duration, ii) medium–level *solution uncertainty* that dynamically calculates rework probability, and iii) high–level *environmental uncertainty* captured by the arrival frequency and magnitude of IECs.

2.3. Rework and change propagation

Evidences show clearly that excessive project budget and schedule overruns typically involve significant effort on rework [14, 15, 16, 26, 29, 30, 32]. Moreover, it is claimed by Reichelt and Lyneis [32] that "these phenomena are not caused by late scope growth or a sudden drop in productivity, but rather by the late discovery and correction of rework created earlier in the project." In this study, primary characteristics of NPD projects will be transformed into a DES model to study their relative impacts on the stochastic arrivals of *rework* (i.e., iterations or EECs).

Rework probability, if included in previous PD process models, is typically assigned a fixed number and remains statically along the process [4, 8, 9, 15, 26]. In reality, however, it is not always the case. Rework probability will be calculated in the proposed DES model by dynamic, evolving solution uncertainty influenced by important feedback effects from other interrelated system variables such as design solutions scope, resource availability, etc. And also, any type of rework is usually discussed on an aggregate level, instead of being categorized into iterations, EECs, and IECs as discussed in this study.

A change rarely occurs alone and multiple changes can have interacting effects on the complex change networks [13]. *Change propagation* is included by considering both of dependent product items and interrelated NPD activities. A complex product usually consists of several interrelated major systems, and each further contains interconnected subsystems, components, and elements. The interactions, in terms of spatial, energy, information, and material [31], that occur between the functional and physical items will cause EC of one product item propagate to the others. Besides highly dependent product configuration, product development activities are also coupled. An EC may propagate to its later activities within the current phase or after. For example, an EC that solves a design fault may trigger further changes to downstream activities in design or production phase.

3. Causal framework

Before the actural construction of a computer simulation model that is quantitatively augmented by algebraic relationships among interrelated variables, causal loop diagrams are first constructed to study how external factors and internal system structure (the interacting variables comprising the system and the cause-and-effect relationships among them) contribute qualitatively to specific behavioral patterns.
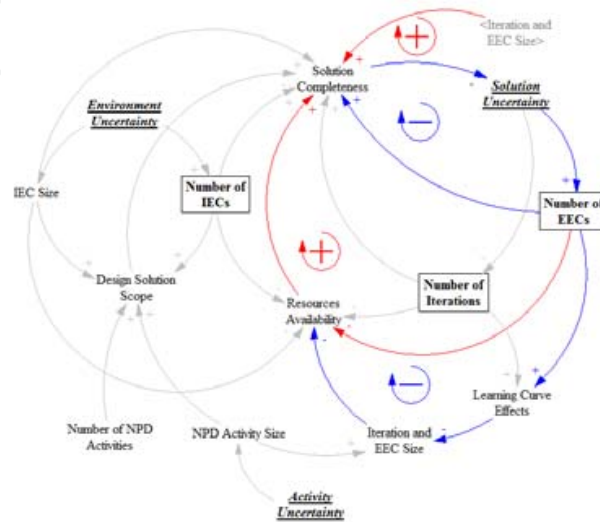
Figure 1. Feedback Loops of EEC Occurrence

Four feedback loops of various lengths (i.e., the number of variables contained within the loop) that drive EEC occurrence are illustrated in *Fig. 1* for purpose of demonstration. Five interdependent variables[1] are considered to form these loops: i) *EEC size*, ii) *solution completeness*, iii) *solution uncertainty*, iv) *Learning Curve Effects (LCE)*, and v) *resource availability*.

3.1. Balancing (negative) loops

Balancing Loop 1 (# of EECs → Solution Completeness → Solution Uncertainty → # of EECs) depicts the reduction in the number of incoming EECs as a result of handling EECs. This phenomenon is due to the fact that processing of more EECs leads to an increase in solution completeness of the NPD project; and thus solution uncertainty decreases. Given the assuption that EEC probability is exponentially decreasing as the project's solution uncertainty decreases, the influence is along the same direction, and therefore number of EEC occurrence decreases.

The reasoning behind *Balancing Loop 2* (# of EECs → Learning Curve Effects → EEC Size → Resource availability → Solution Completeness → Solution Uncertainty → # of EECs) is that an increase in occurrence of EECs leads to a reduction in later EEC durations compared with the original level (i.e., the basework duration of that particular activity) because of increasing LCE. As a result, resource availability increases since less time is taken

---

[1] See *Subsection 4.3.3* for detailed mathematical definition of variables ii) and iii).

for completing EECs, which in turn accelerates the rate of solution completeness and thus leads to the decreasing occurrence of EECs.

3.2. Reinforcing (positive) loops

While the explanation of Balancing Loop 2 is based upon the indirect positive impact of EEC occurrence on resource availability through the reduction of later EEC durations owing to learning curve effects, *Reinforcing Loop 3* (# of EECs → Resource availability → Solution Completeness → Solution Uncertainty → # of EECs) can be interpreted by the direct negative influence of EEC occurrence on resource availability: the more EECs occur, the more resource will be allocated to process them. As opposed to Loop 2, a decrease in resource availability decelerates the rate of solution completeness, and thus causes an increasing occurrence of EECs.

Despite the indirect effects of EEC size reduction on an accelerating solution completeness rate that results in an increase in the resource availability, a decrease in EEC size also has a direct negative impact on solution completeness because of less contribution to close the information deficiency towards the final design solution, which is shown in *Reinforcing Loop 4* (# of EECs → Learning Curve Effects → EEC Size → Solution Completeness → Solution Uncertainty → # of EECs).

3.3. Summary

The above four closed feedback loops depict how the initial occurrence of EECs will lead to the subsequent modification of occurrence frequency by taking into account other interrelated variables and presenting simple cause–and–effect relationships between them. A combination of both positive and negative feedback loops indicates that the complex and dynamic interrelationships among variables make the prediction of occurring patterns of iterations/EECs not so straightforward. This phenomenon points out the necessity of constructing a simulation model that can help further quantitative analyses.

4. Model description

4.1. General assumptions

This model has two constituent sections: *NPD Section with Reworks* and *IEC Section*. Primary model assumptions underlying are listed below.

1. The overall structure of NPD process can be systematically planned beforehand in an activity–based representation according to historical data from previously accomplished projects of similar products and teams' expertise as well. All NPD phases and activities, their expected durations and units of resource required, and interdependencies relationships among them are obtainable and remain stable as the NPD project evolves. Therefore, optimization of process sequencing and scheduling is not pursued by this study.

2. There is no overlapping between activities within a same phase. An NPD activity only receives finalized information from its upstream activity within one phase, but downstream action can start with information in a preliminary form before all activities in upstream phase are completed. In addition, there is no information exchange in the middle of an activity.

3. Demand on resource for NPD activity is assumed to be deterministic fixed. However, the activity duration varies stochastically subject to activity uncertainty and LCE which vary depending on the number of attempts to that particular activity.

4. The dynamic progress of an NPD entity is reflected in the work flow within and among NPD phases. Workflow routing is probabilistically altered by either intra–phase iterations or inter–phase EECs according to the dynamically updated rework probabilities, which are calculated based on the current value of solution uncertainty.

5. Each IEC is initially associated with a directly affected NPD activity (and a directly affected product item when product structure is modeled), and may further propagate to any downstream activities based on randomly assigned probabilities. IECs are modeled within a parallel co–flow structure similar to its NPD counterpart. The IEC work flow is restricted by the precedence constraints.

4.2. Notations

Based upon these general assumptions made, notations of important model parameters and variables which will be later used in mathematical formulation are introduced.

4.2.1. Model parameters

-   I: number of NPD phases
-   J_i: number of NPD activities within phase i (for $i = 1, 2, \ldots, I$)
-   M: number of participating departments
-   R_m: total number of resources available from department m (for $m = 1, 2, \ldots, M$)
-   r_{ijm}: units of resource required from department m to complete activity j (for $j = 1, 2, \ldots, J_i$) in phase i
-   d_{ij}:[2] time expected to complete activity j in phase i when resource requirement is met
-   D_{ij}: mean value of d_{ij}, $D_{ij} = \beta k$

4.2.2. Model variables

-   i_t / j_t: the latest–finished activity basework j_t in phase i_t at time t
-   n_t: number of reworks finished at time t
-   x_1 / y_1: the first rework for activity y_1 in phase x_1
-   x_{n_t} / y_{n_t}: the latest–finished rework for activity y_{n_t} in phase x_{n_t} at time t
-   $(R_m)_t$:[3] the cumulative functional effort of the ongoing rework(s) at time t

---

[2] The Erlang distribution $ERLANG\ (\beta, k)$ is used as a description of NPD activity duration

[3] An aggregate term consists of ongoing rework(s)/rework propagations each one corresponding to its current stochastic functional effort value

- L_t: number of IECs finished at time t
- g_{l_1}: the activity in which IEC l is initiated
- $g_{l_{G_l}}$: number of activities IEC l propagates to ($G_l \leq I \times J_i$)
- s_{lgm}: resources required from department m to complete IEC l (for $l = 1, 2, \ldots, L_t$) to activity g (for $g = g_{l_1}, g_{l_2}, \ldots, g_{l_{G_l}}$)
- w_{lg}:[4] time expected to complete IEC l to activity g
- $(I_m)_t$:[5] the cumulative functional effort of the ongoing IEC(s) at time t

4.3. Design solution scope

Design Solution Scope (DSS) is defined as the overall extent of an NPD project in terms of total effort required (person–days), by completing of which the entire set of product goals will be met. It depends not only on the number of constituent activities, but also the expected duration and units of resources needed to produce the desired outputs of each activity. In a sense, design solution scope indicates one facet of the NPD project complexity with regards to its content (as a function of activity duration d_{ij} and demand for resource r_{ijm}). Of course, project complexity can also be reflected from the perspective of its architecture (i.e., the coupling among product components or the process precedence constraints), which will be discussed more in later sections on the topics of overlapping and rework probabilities.

The estimated functional effort to complete the whole NPD project is obtained as follows:

$$EN_m = \sum_{i=1}^{I} \sum_{j=1}^{J} e_{ijm} = \sum_{i=1}^{I} \sum_{j=1}^{J} (r_{ijm} \times d_{ij}) \tag{1}$$

Let's assume that L_t is the total number of incoming IECs that are finished at time t, g_{l_1} is the activity to which a randomly occurring IEC l (for $l = 1, 2, \ldots, L_t$) is directly related, and $g_{l_{G_l}}$ is the last activity along the IEC propagation loop. Through the estimation of IEC duration w_{lg} and s_{lgm} number of resource required from department m, the functional effort needed to process IEC l to activity g (for $g = g_{l_1}, g_{l_2}, \ldots, g_{l_{G_l}}$) is $e_{lgm} = s_{lgm} \times w_{lg}$. By a double summation over both l (of the entire set of completed IECs) and g (including the original incoming IEC and a sequence of its propagations), the cumulative functional IEC effort at time t can be represented as

$$(EI_m)_t = \sum_{l=1}^{L_t} \sum_{g=g_{l_1}}^{g_{l_{G_l}}} e_{lgm} + (I_m)_t = \sum_{l=1}^{L_t} \sum_{g=g_{l_1}}^{g_{l_{G_l}}} (s_{lgm} \times w_{lg}) + (I_m)_t \tag{2}$$

Note that besides the first term $\sum_{l=1}^{L_t} \sum_{g=g_{l_1}}^{g_{l_{G_l}}} e_{lgm}$ which describes the total functional effort spent on those already completed IECs, another aggregate term $(I_m)_t$, which represents the cumulative functional effort of the ongoing IEC(s) at time t, is used to avoid the inherently tedious expression of such stochastic, probabilistic, and discrete events in a mathematical formula. The difficulties encountered here in translating such occurrences into a precise math equation, again, confirm the advantages of using computer simulation as the research methodology in studying the interrelated and dynamic ECM problems.

---

[4] The Triangular distribution $TRIANGULAR\ (Min, Mode, Max)$ is used as a description of IEC duration

[5] An aggregate term consists of ongoing probabilistically dependent IEC(s)/IEC propagations each one corresponding to its current stochastic functional effort value.

Based on EN_m and $(EI_m)_t$, a dynamic NPD property, functional design solution scope $(S_m)_t$, can be obtained as appeared in Eq. (3) by making the following assumptions:

$$(S_m)_t = EN_m + (EI_m)_t \qquad (3)$$

1. DSS of an NPD project reflects the amount of effort needed to meet the entire set of product goals, including both original pre–defined goals when the project is initiated and those additional ones determined as the project evolves.
2. Both iterations and EECs are mandatory error–correction oriented to achieve the same pre–defined goals, and thus there is no overall resultant increase in DSS. However, they will be taken into account when calculating the actual cumulative functional effort.
3. IECs are carried out to accomplish additional product goals in response to outside requirements such as altering market demands, growing customer needs, new legislations, or rapid advances in technology. IEC arrivals cause increase in DSS.

4.4. NPD framework with iterations and EECs

From an "*information processing*" view, the generic activity network proposed in [3, 4] is adopted as the fundamental modeling structure. By doing so, the NPD process can be decomposed into I numbers of *Phase* P_i ($i = 1, 2, \dots, I$) with certain degrees of overlapping. Each phase is further made up of J_i sequentially numbered *Activities* P_iA_j ($j = 1, 2, \dots, J_i$) to represent several chronological stages in design and development process. The present study assumes that there is no overlapping among activities within each phase. That is, within a single phase an NPD activity begins only after the completion of its predecessor. However, NPD phases can be overlapped by letting the successor phase begin with only preliminary information before activities in the upstream phase are all finished.

The completion of an NPD activity for the first time is called **NPD basework**. Any later attempt, no matter in the form of *intra–phase iteration* or *inter–phase EEC*, is referred as **rework**. When work flow is routed back by probability, it is assumed that some of the previously completed activities have encountered errors and the farthest upstream one will be identified as the "starting point" of rework loop. All downstream activities are supposed to be "corrupted" and have to be reattempted before moving on. *Fig. 2* illustrates this *I–*phase and *J_i–*activity NPD framework.
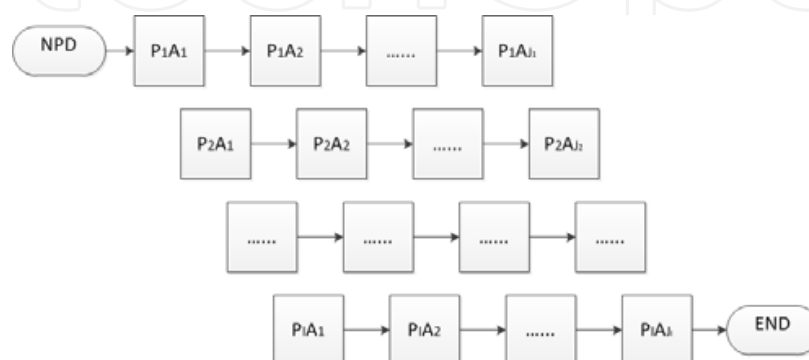
Figure 2. *I*–phase & *J_i*–activity NPD framework

4.4.1. NPD activity duration and learning curve effects

Low–level activity uncertainty is represented by random variation of activity duration around its estimate. For each NPD activity, its duration d_{ij} is sampled from a pre–determined probability distribution. The Erlang distribution $ERLANG\ (\beta, k)$ is used as a description of the activity duration. Employment of the Erlang distribution to represent activity interval is based on the hypothesis that each NPD activity consists of k number of random tasks, everyone individually having an identical exponentially distributed processing time with mean β. These mutually independent tasks can be considered as the lowest un–decomposable unit of the NPD process. Number of tasks k comprising each activity and the anticipated task duration β should be estimated by process participants and provided as model inputs.

According to the learning curve theory, the more often an activity is performed, the less time it requires to complete it and thus the lower will be the cost. This well recognized phenomenon is included as a process characteristic to improve the comprehensiveness of this DES model. Following the assumptions made in [9], LCE is modeled in the form of a linearly diminishing fraction ($0 < L_f < 1$) of the original duration whenever an activity is repeated until the minimum fraction ($0 < L_{min} < L_f < 1$) is hit and the rework processing time remains unchanged afterward. That is to say, learning curve improves through each round of rework until it reaches the minimum fraction of basework duration which is indispensible for activity execution. Let n be the number of times an activity is attempted, LCE can be expressed as:

$$LCE = \max\left(\left(L_f\right)^{N_{ij}-1}, L_{min}\right) \tag{4}$$

Therefore, the processing time of a rework to an NPD activity depends on two variables: the stochastic basework duration d_{ij} of the activity and the number of times N_{ij} it is attempted. Any types of NPD rework, no matter intra–phase iterations or inter–phase EECs, are assumed to be subject to the same LEC.

4.4.2. Overlapping and cross–functional interactions

Overlapping is defined as the partial or full parallel execution of nominally sequential development activities [25]. The underlying risk of overlapping raised by Krishnan that "the duration of the downstream activity may be altered in converting the sequential process into an overlapped process" [24] is addressed here in a slightly different way from directly increasing downstream duration and effort by a certain calculated value (e.g., [33]). The more number of activities start with information in a preliminary form or even missing information, the less is the design solution completeness, which will in turn affect rework probabilities as discussed in detail in the next section.

The concept of cross–functional integration among different functional areas during an NPD process is defined as **Departmental Interaction** (DI). One of the m departments takes major responsibility for the phase in its own area with specialized knowledge, and is called *major*

department during that phase. However, the other $m - 1$ departments, defined as *minor departments*, also need to participate but with less resource requirements. Cross–functional integration enables a decentralized NPD process by facilitated communications among involving departments. Recourse consumption in the form of departmental interaction is, again, an estimate from process participants. Resources can represent staffs, computers /machines, documentations, or any other individual servers. It's assumed that each resource is qualified to handle all the NPD activities within all phases.

4.4.3. Solution uncertainty

In the process modeling literature, NPD is often considered as a system of interrelated activities that aims to increase knowledge or reduce uncertainty of the final design solution [7, 24, 37]. This DES model assumes that any knowledge or experience accumulation through an NPD activity, no matter accepted to be transferred to the next activity/activities or rejected and requested for a rework, will contribute to the common knowledge base of the NPD project towards its final design solution. No development effort is ever wasted. In this context, knowledge/experience accumulation is simply measured by the cumulative effort that has been committed to the project in terms of person–days.

Functional solution completeness is defined as a criterion to reflect the effort gap between the actual cumulative functional effort accomplished to date and the evolving functional design solution scope $(S_m)_t$. The exact expression for $(C_{ijm})_t$ is determined by the amount of overlap between NPD activities. The more concurrency a process holds, the more complicated the expression will be. Eq. (5) is an illustration of solution completeness at time t for the easiest case: a sequential process. $(C_{ijm})_t$ is improved by knowledge or experience accumulation through performing NPD basework (indicated by the first term in Eq. (5)) and rework (the second term), and handling IECs (the third term). Again, a generalized abstract term $(R_m)_t$ is used here to represent the cumulative functional effort of the ongoing rework(s) at time t.

$$\left(C_{ijm}\right)_t = \frac{\left(\sum_{i=1}^{i_t-1}\sum_{j=1}^{J}e_{ijm} + \sum_{j=1}^{j_t}e_{i_tjm}\right) + \left(\sum_{x=x_1,y=y_1}^{x=x_{n_t},y=y_{n_t}}\left(\sum_{i=x+1}^{i_t}\sum_{j=1}^{j_t}e_{ijm} + \sum_{i=x}^{I}\sum_{j=y}^{j_t}e_{ijm}\right) + (R_m)_t\right) + (EI_m)_t}{(S_m)_t} \quad (5)$$

On the contrary, *functional solution uncertainty* $(U_{ijm})_t$ reflects the degree of functional effort absence towards the design solution scope. Therefore, the solution uncertainty of activity j in phase i at time t is $(U_{ijm})_t = 100\% - (C_{ijm})_t$.

4.4.4. Rework probability

After each activity, there is a rework review decision point that decides whether the activity output is acceptable and the NPD project entity gets through or it needs to flow back for a rework according to a weighted rework probability determined by the latest levels of functional solution uncertainty. A critical assumption we made is that the *iteration probability* of an activity is negatively proportional to the NPD project's latest level of solution uncertainty. That is, chance of an activity gets to iterate before it is released to the next phase

will increase as the project unfolds with more information available and its solution uncertainty decreases. Two arguments are presented to backup this assumption:

1. As the project unfolds, more information will be available to justify further iteratively refinement of the design solution for each component [37].
2. Since a product architecture often consists of multiple conflicting targets that may be difficult to meet simultaneously and thus requires further trade–offs, "design oscillations" on a system level may occur due to the interdependencies among local components and subsystems even after the achievement of individual optimum [10, 28].

The functional iteration probability is formulated by a negative exponential function of uncertainty as appeared in Eq. (6), where $0 < \alpha < 1$ is a process–specific *Iteration Probability Constant* (IPC) that should be determined beforehand as a model input.

$$(PI_{ijm})_t = \alpha^{(U_{ijm})_t + 1} \tag{6}$$

Since NPD activities are decentralized through cross–functional integration among participating departments, so is the decision making process of carrying out rework. The overall iteration probability of activity j in phase i is the weighted mean by the number of resources each department commits to the activity.

$$(PI_{ij})_t = \frac{\sum_{m=1}^{M}(r_{ijm} \times (PI_{ijm})_t)}{\sum_{m=1}^{M} r_{ijm}} \tag{7}$$

Similarly, *EEC probability* is characterized by an **EEC Probability Constant** (EPC) $0 < \gamma < 1$. However, as opposed to iteration probability, it is assumed to be exponentially decreasing as the project's solution uncertainty decreases. That is to say, the chance of revisiting NPD activities, whose outputs have already been frozen and released to their successor phase, is the highest after the first activity of the second phase and continuously reduces according to the continually increasing design solution completeness.

$$(PE_{ijm})_t = \gamma^{(C_{ijm})_t + 1} = \gamma^{2 - (U_{ijm})_t} \tag{8}$$

$$(PE_{ij})_t = \frac{\sum_{k=1}^{n}(r_{ijm} \times (PE_{ijm})_t)}{\sum_{k=1}^{n} r_{ijm}} \tag{9}$$

Given the overall rework probability, the next step is to identify which upstream activity generates the design error disclosed by rework review and therefore becomes the starting point of rework loop. For simplicity, it is assumed that each upstream activity gets an equal chance of initiating an intra–phase iteration loop or an inter–phase EEC loop.

4.4.5. Rework criteria and rigidity of rework review

According to the rationale explained in previous subsections and causal loop diagrams created, the occurrences of both iterations and EECs are governed by a combination of balancing and reinforcing loops. Take *Loop 3* as an example, less resource availability resulted from increasing EEC arrivals will decelerates the rate of solution completeness, and further increase the occurrence of EECs.

To avoid the dominance of such reinforcing loops which will eventually lead to a net effect of overall divergence with no termination condition, *rework criteria* are established as the first step of rework review after the completion of an activity to check whether the cumulative functional effort committed to the deliverable is high enough to provide a satisfying outcome, and therefore let the NPD entity pass rework evaluation. If the cumulative devoted effort fails to meet the pre–determined criteria (i.e., the cumulative effort is less than the expected amount), the entity will be evaluated at the rework decision–point and go for iteration or EEC according to the rework probability calculated by solution completeness. If the committed effort is higher than the pre–set amount, the NPD entity will conditionally pass rework evaluation and continue executing next activity/activities.

Unger and Eppinger [36] define *rigidity* by the degree to which deliverables are held to previously–established criteria as metrics to characterize design reviews. By putting it in a slightly different way, rigidity of rework review is considered in this DES model as the strictness of pre–defined rework criteria with respect to the amount of cumulative effort committed to a particular NPD activity.

4.5. IEC framework

Unlike iterations and EECs, IECs are studied through a different DES model section other than the NPD framework. The IEC framework explores how IECs emerging from outside sources after the NPD process begins are handled and how an initiating IEC to a specific activity of a product item will cause further change propagation in its downstream activities and other dependent product items.

4.5.1. IEC processing rules

IECs affecting activities in different NPD phases are assumed to arrive in randomly after the NPD project starts. A checkpoint is inserted before the processing of an IEC to verify whether the directly affected NPD activity has started yet. The incoming IEC will be hold until the beginning of processing of that particular activity.

During NPD rework reviews, the upcoming NPD activity will also be hold from getting processed if there are IECs currently being handled with respect to any of its upstream activities until new information from these IECs becomes available (i.e., the completion of IECs). Purpose of such an inspection is to avoid unnecessary rework as a result of expected new information and updates. However, the NPD activity will not pause in middle of its process due to the occurrence of IECs to any of its upstream activities.

Fig. 3 summaries the entire rework review process after the completion of each activity that includes three major steps as discussed before:

1. Check if there are currently any IECs being handled with regards to any of its upstream activities. If the condition is true, wait until new information from all of these IECs becomes available; if condition is false, go to the next step;

2. Compare the cumulative devoted functional effort so far to the pre–determined rework criteria. If the condition is true, the work flow conditionally pass the rework review and directly proceeds to next activity/activities; if the condition is false, go to the next step;

3. As a result of cross–functional negotiation and integration, calculate rework probability according to the current levels of functional solution uncertainty. NPD project entity either flows back to the identified activity that contains design errors for rework or proceeds to the activity/activities by probability.
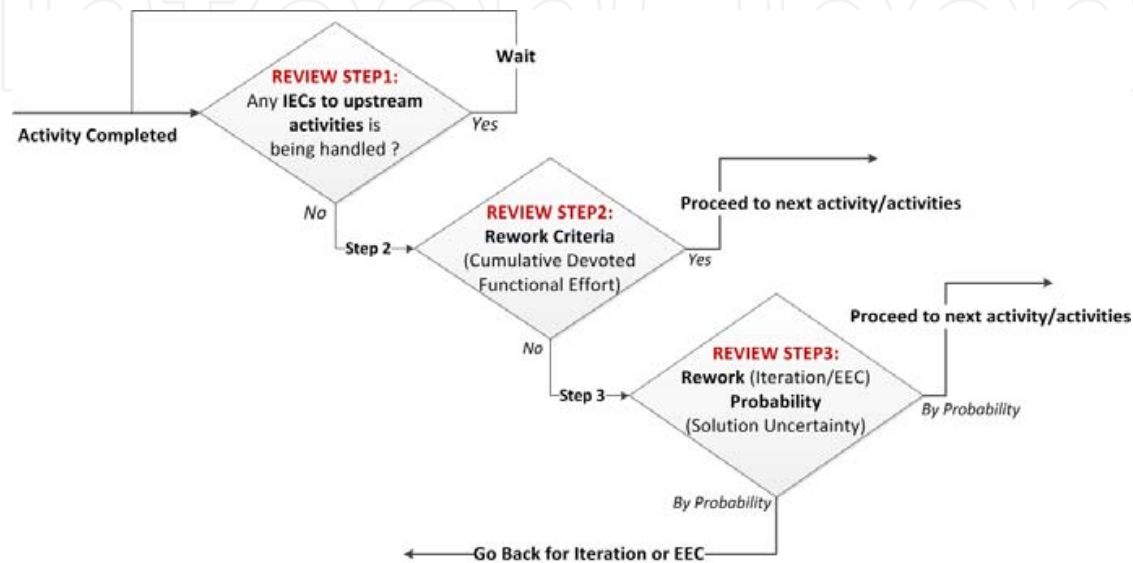
Figure 3. 3-Step NPD Rework Review Process

4.5.2. Frequency and resource consumption of IEC

Compared with NPDs that are much more likely to adhere to a planned schedule, IECs can occur without any plans. Therefore, the Exponential distribution is used to represent IECs' arrival interval. IEC's processing time is assumed to follow the Triangular distribution, where there is a most–likely time with some variation on two sides, represented by the most likely (Mode), minimum (Min), and maximum (Max) values respectively. The Triangular distribution is widely used in project management tools to estimate activity duration (e.g., Project Evaluation and Review Technique, Critical Path Method, etc.). The amount of resources required for an IEC to be processed is *IEC effort*. When there are not enough resources available for both processes, resource using priority needs to be assigned to either NPD or ECM to seize necessary resource first.

4.5.3. IEC propagation

Change Propagation (CP) is assumed to be rooted in both interrelated *activities* of a PD process and closely dependent constituent product *items*. That is, modifications to an initiating activity of one product item are highly likely to propagate to other activities within the same or different stages along the PD process, and may require further changes across to other items that are interconnected through design features and product attributes [23].

This phenomenon is simulated by two layers of *IEC propagation loop*. Firstly, CP review decisions are performed after the completion of an IEC and then propagate to one of its downstream activities by predefined probabilities. We assume uni–directional change propagation based on process structure. That is, an IEC to one NPD activity will propagate only to its successor activities within the current or next phase. For example, an IEC to enhance a particular design feature may result in substantial alterations in prototyping and manufacturing. On the other hand, innovations in manufacturing process will only cause modifications within production phase but not changes in design.

Secondly, the first–level activity IEC propagation loop is then nested within an outer loop determined by particular dependency properties of the product configuration. Once an IEC to one product item and its CPs to affected downstream activities are completed, it will further propagate to item(s) that is/are directly linked to it.

5. Numerical application

A numerical example is presented in this section to illustrate how this DES model can actually be applied to facilitate ECM policy analysis. A combination of different process, product, team, and environment characteristics are tested through design of experiments. NPD project lead time, cost (or engineering effort in some cases), and quality are generated by the model as the three key performance measurements of the project under study to evaluate overall product development efforts.

5.1. NPD section

The NPD section is demonstrated by a simple application of three representational phases of an NPD process: i) concept design and development (*Concept*), ii) detailed product design (*Design*), and iii) production ramp up (*Production*). Each phase consists of three sequentially numbered and chronologically related activities. The information flow between every two activities is indicated by solid arrows as shown in *Fig. 4*.
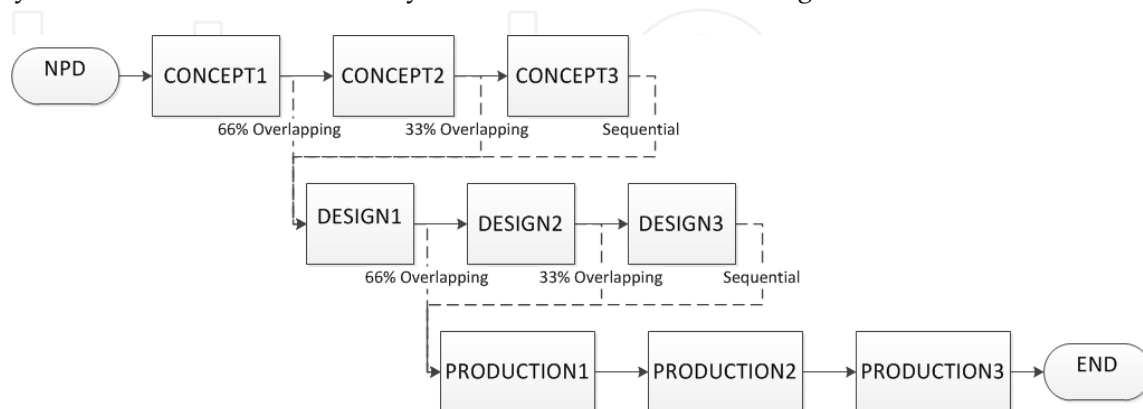
Figure 4. NPD Framework with Iterations and EECs

Through this 3–phase and 3–activity framework, various overlapping ratios of an NPD process: 0%, 33%, 66%, or mixed (e.g., 0% overlap between Concept and Design and 33%

overlap between Design and Production), can be constructed by connecting intra–phase activities via different combinations of dashed arrows.

5.2. Overlapping strategy

An NPD process with 0% overlapping is also called a *sequential* process, in which the downstream phase is allowed to start only after receiving the output information from the upstream phase in its finalized form. That is, different phases comprising an NPD process are connected in a completely linear fashion.

Besides its capability of representing a sequential process, this framework can also be assembled into *concurrent* processes by allowing the parallelization of upstream and downstream activities. For a 33% overlapped process, the first activity of downstream phase begins simultaneously with the last activity of upstream phase. For a 66% overlapped NPD process, the first activity of the following phase starts simultaneously with the second activity of the preceding phase.
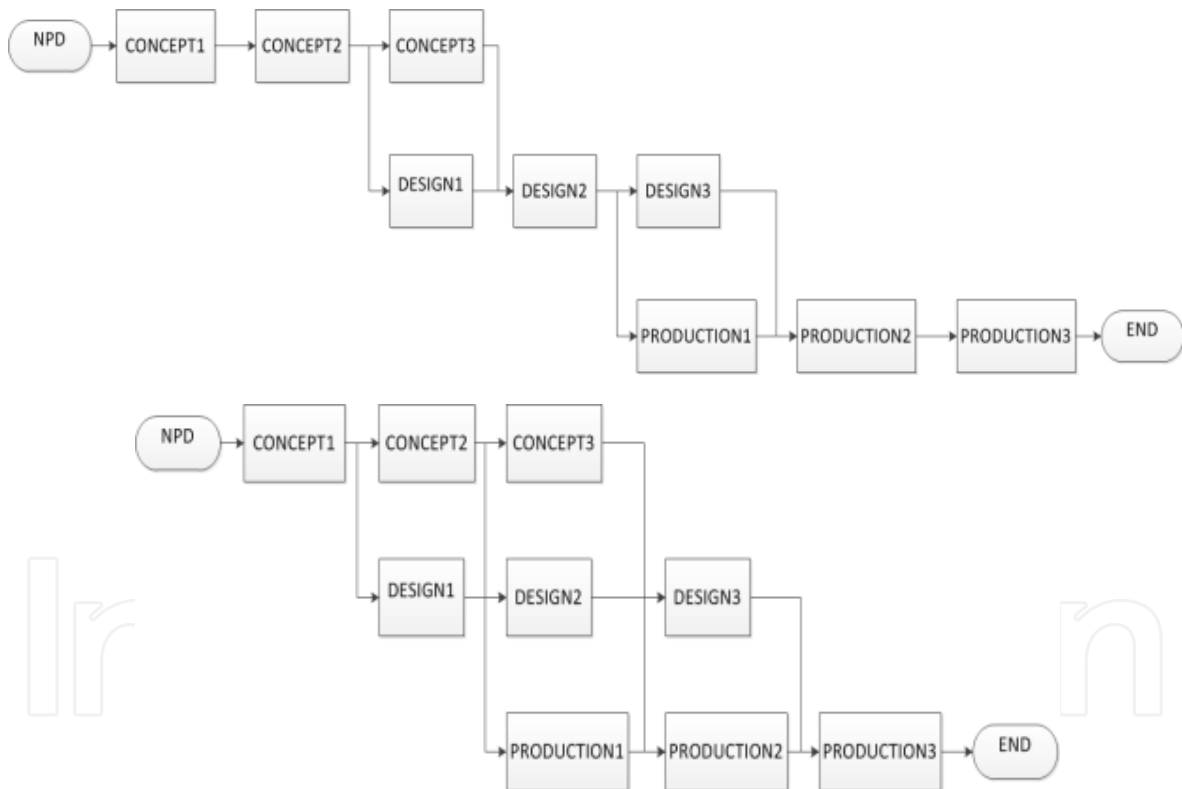
Figure 5. NPD Framework with Iterations and EECs

Obviously, as compared to its counterpart in a sequential process, the solution uncertainty of downstream activity increases due to the fact that it begins before the completion of all upstream activities using only preliminary output information, while the solution uncertainty of the upstream activity remains unchanged. That is, only the solution uncertainty of overlapped activities in succeeding phases will be affected under the current model assumptions.

5.3. NPD process parameter

When considering the activity duration estimates, it is further assumed that the mutually independent and exponentially distributed duration has a mean of $\beta = 2$ days for activities in all three phases. Furthermore, the number of tasks that compose activities within one phase remains the same, but increases from phase to phase to represent the increasing content and complexity of design and development activities as the NPD project unfolds: $k = 4$ for activities in Concept phase; $k = 6$ for Design phase; and $k = 10$ for Production phase. Note that when LCE are taken into account, random variables described by the Erlang distribution $ERLANG(\beta, k)$ only represent processing intervals of NPD basework. Rework duration is also subject to N_{ij}, the number of times that an activity is attempted, in the form of

$$LCE = max\left(\left(\frac{1}{2}\right)^{N_{ij}}, 0.1\right)$$

To match the three major phases of the illustrated NPD process, it is assumed that there exist three different functional areas: *marketing*, *engineering*, and *manufacturing*, that participate in the overall NPD process through integrated DI. Based on the model assumption that each activity consumes a total number of 100 resources units to complete, DI is defined as follows: 60 units requested from major department and 20 units requested from each of the other two minor departments. To estimate the final project cost, the busy usage cost rates are set as *$25/hour* and idle cost as *$10/hour* for all resources.

Different rigidities of rework review, which are represented by various rework criteria ratios (i.e., relationships between rework criteria and the evolving functional design solution scope $(S_m)_t$) will be explored more in depth through "what–if" analysis.

5.4. IEC section

Fig. 5 gives an overview of the IEC model section applying 33% overlapping strategy. It is assumed that an IEC will propagate to one of its downstream activities in the current or next phase with equal chances, and this propagation will continue in the same manner until the end of IEC propagation loop when no more change is identified. For the purpose of demonstration, a full list of potential downstream change propagations of each IEC is provided on the right side of the *IEC Propagation* decision point. In the actual simulation model, verbal description is replaced by connectors between the IEC propagation decision point and the corresponding IEC process modules.

Take the IEC to activity Concept1 as an example, change propagation will result in a maximum of six follow–up IECs (i.e., IECs to C2, C3/D1, D2, D3/P1, P2, and P3) and a minimum of two (i.e., IECs to C3/D1/D2 and D3/P1/P2/P3). For simplicity, it is also assumed that each IEC, no matter in which activity it is occurred, equally consumes 10 resource units from each of the three departments to get processed.
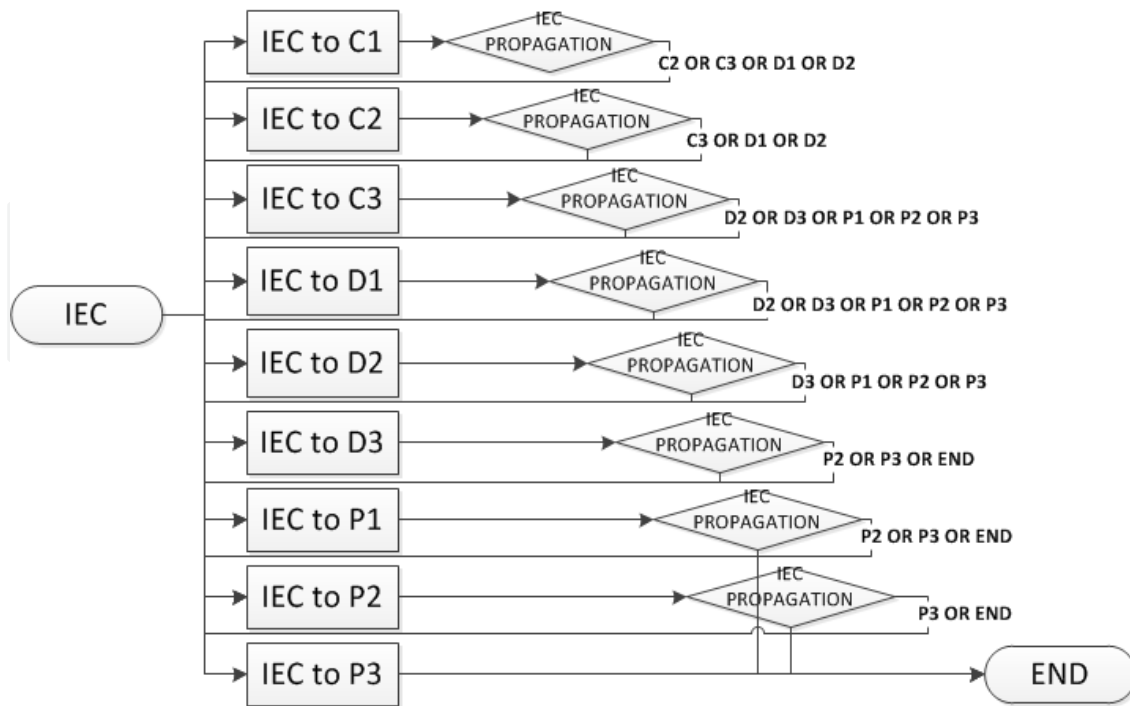
Figure 6. NPD Framework with Iterations and EECs

5.6. Summary of model inputs and outputs

Table 1 summarizes the complete list of model input parameters/variables and their corresponding values. There are altogether 14 model inputs that represent key NPD and ECM decision parameters, among which 7 are chosen as *design factors* or *constraints* (highlighted rows in gray) and their effects on the project performance measures (i.e., model outputs) will be tested at specific *levels* (highlighted text in bold), while others will be held constant when the design of experiment is conducted.[6]

It is important to know that all these values are set in a way to facilitate relative comparison of project performance among various scenarios using "what–if" analysis instead of aiming to reproduce the real behavior patterns of an NPD project of any kind. To successfully implementation of the proposed simulation model for a specific use or situation, these inputs should be appropriately calibrated depending on different circumstances.

At the end of each simulation run, Arena automatically generates a variety of both default and user specified model output statistics, which include time, cost, Work in Process (WIP), count, etc. Information is displayed under different categories (e.g., Entity, Process, Queue, Resource, and User Specified). Some of the key model *responses* are listed in the table below.

---

[6]These held-constant factors, such as number of phases and activities comprising the process, number of involving departments, duration estimates of NPD activities and IECs, etc., are peculiar to specific development project as . For purposes of the present experiment these factors are not of interest.

Input Data	Value
List of phases and activities comprising process	$I = 3\ (Concept - Design - Production);$ $J_i = 3\ (e.g.\ Concept1 - Concept2 - Concept3)$
List of involving departments	$M = 3\ (Marketing; Engineering; Manufacturing)$
Overlapping Strategy **(OS)**	**Low**: 0%; **Medium**: 33%; **High**: 66%
NPD Activity Duration *(days)*	$d_{1j} = ERLANG(2,4), D_{1j} = 8;\ d_{2j} = ERLANG(2,6), D_{2j} = 12;\ d_{3j} = ERLANG(2,10), D_{3j} = 20;\ j = 1,2,3$
Learning Curve Effects **(LCE)**	**no LCE**; **LCE** $= max\left(\left(\frac{1}{2}\right)^{N_{ij}}, 0.1\right)$
NPD Activity Functional Resource Consumption	$r_{1j1} = 60, r_{1j2} = r_{1j3} = 20;\ r_{2j1} = 20, r_{2j2} = 60, r_{2j3} = 20;\ r_{3j1} = r_{3j2} = 20, r_{3j3} = 60;\ j = 1,2,3$
Functional Resources Constraints **(FRC)**	$R_m = 70, 80, \ldots, 190, 200;\ m = 1,2,3$
Cost of Resource	$Busy/Hour = \$25;\ Idle/Hour = \$10$
Rework Likelihood **(RL)**	**Low**: $\alpha = \gamma = 0.3$; **High**: $\alpha = \gamma = 0.45$
Rework Criteria **(RC)**	**Stepped Linear; Linear; Convex–Up; Concave–Up**
IEC Arrival Frequency (Inter–arrival Times) *(days)*	**Low**: $Random\ (Expo)20$; **Medium**: $Random\ (Expo)10$; **High**: $Random\ (Expo)5$
IEC Duration Estimates *(days)*	$w_{lg} = TRIA(1.6, 2, 3.2), g = 1,2,3;$ $w_{lg} = TRIA(2.4, 3, 4.8), g = 4,5,6;$ $w_{lg} = TRIA(4,5,8), g = 7,8,9;\ l = 1,2,\ldots,L_t$
IEC Functional Resource Consumption	$s_{lgm} = 10\ and\ 20,\ l = 1,2,\ldots,L_t;\ g = 1,2,3;\ m = 1,2,3$

Table 1. Model Inputs

Output Data	Definition
NPD Project Lead Time	The total time of an NPD entity accumulated in process activities and delays (time elapsed between start of Concept phase and end of Production phase).
Project Cost	The total of busy costs (i.e., costs while seize) for all staffing and resources for both NPD and IEC entities.
Total Cost	The total expenditure on both busy and idle (i.e., costs while scheduled, but not busy) resources for NPD and IEC entities.
Cumulative Functional Effort	The accumulated departmental workload (in units of person–days) accounted for both NPD and IEC entities.
Cumulative Total Effort	The accumulated total effort accounted for both NPD and IEC entities (i.e., the sum of all the cumulative functional efforts).
Quality	Ratio of the final design solution scope over the original design solution scope.

Table 2. Model Outputs

6. Results

Impacts of the following managerial strategies and coordination policies on the responses of interest are investigated, and the root causes behind the performance of measurement system are explored:

a.   Impact of NPD process characteristics such as LCE, Rework Likelihood (RL) and *Overlapping Strategy* (OS);
b.   Impact of rework review rigidity – *Rework Review Strategy (RRS)*[7];
c.   Impact of IEC arrival frequency;
d.   Combined impact of IEC arrival frequency and size – *IEC batching policy*;
e.   Impact of functional resource constraints – *resource assignment Strategy;*
f.   Impact of change propagation due to interconnected product configuration.

Due to space limit, only partial results of policy analysis a are presented to demonstrate how the proposed DES model can be used as a valuable tool for evaluating ECM decisions. 200 *replicates* are generated under each combination of LCE, RL, and OS, and thus result in altogether 2400 simulation runs, each using separate input random numbers. Performance data generated by the model are then exported to a Excel worksheet, in which individual project performance measures are recorded and various data graphs are generated.

Mean values of the experiment outcomes are displayed in Table 3. Columns *(i)* and *(ii)* record in an absolute sense the mean values of the observed lead time and project cost from 200 replications of each scenario, while columns *(I)* and *(II)* show the percentage change of (i) and (ii) relative to the baseline case results *(BL1)*, respectively.

It is important to note that managerial suggestions are not made merely based on the final output performance measures obtained for each scenario. Rather, attention is focused on the comparison of these numbers to their corresponding baseline results, which helps to provide us intuitive understanding of the impacts of reworks on project performance under different process features and parameter settings. Through the interpretation of results presented in *Table 3*, several concluding observations can be issued:

1.   When rework is not involved, the project performance stays consistent: the higher the activity overlapping ratio, the less the lead time. It can be obtained by summing up the durations of activities along the critical path. At the same time, since total person–days effort required for completing the project remains unchanged no matter which OS is applied, final project cost for all levels of *OS* (i.e., *(a)*, *(b)*, and *(c)*) in the baseline case should be very much similar, which is confirmed by the running results. This can be considered as a simple model verification check[8].
2.   Effects of LCE: by comparing the mean values of lead time and project cost of scenarios *(A)* with scenarios *(B)* under different combinations of RL and OS levels, it can be

---

[7] The first two strategies are analyzed with only the NPD section of the model.
[8] Model is continuously verified by the reading through and examining the outputs for reasonableness and justification under a variety of scenarios and settings of parameters.

concluded that the evaluation of learning curve effects unambiguously results in a remarkable decrease in both NPD lead time and cost.

3. Effects of RL: by comparing *(i)* and *(ii)* of scenarios *(1)* with scenarios *(2)* under different combinations of LCE and OS levels, it can be concluded that a higher likelihood of rework in NPD activity undoubtedly causes an increase in both lead time and cost.

4. Effects of OS w/o LCE: by comparing lead time and project cost of scenarios *(A)* in a relative sense, we find that an increasing overlapping ratio aggravates the impact of NPD rework on both responses. That is, when NPD rework is included in the model but no LCE is considered, the greater the overlapping ratio, the higher the percentages of increase in both lead time and project cost as compared to baseline case. In addition, we notice the time–cost tradeoffs between a sequential process and a 66% overlapped process from columns *(i)* and *(ii)*. This observation agrees to the general acknowledgement that overlapping may save time but is more costly.

5. Effects of OS w/ LCE: Situation is not that predictable when LCE is taken into account and formulated as $LCE = \max\left(\left(\frac{1}{2}\right)^{N_{ij}-1}, 0.1\right)$ in the model. Significant increase of both time and cost due to rework is alleviated by the evaluation of LCE. Under low RL circumstances ($\alpha = \gamma = 0.3$), a highly overlapped process excels in both response variables in an absolute sense. However, there is not clear trend shown in the comparative values. Particularly, at high level of RL ($\alpha = \gamma = 0.45$), we observe that a 33% overlapped process leads to both absolute (compared with the results of 0% and 66% in scenario *(B)–(2)*) and relative (compared with the 33% baseline results *(BL1)–(b)*) maximum values for lead time and project cost.

6. By comparing columns *(I)* and *(II)*, we observe a project behavioral pattern that the percentage increase of project cost is always higher than that of lead time at the occurrence of rework. That is to say, compared with lead time, project cost is more sensitive to rework. And the difference between the two percentages of increase is largest when a sequential NPD process is adopted. The only exception is scenario *(B)–(1)–(c)* with the percentage increase of project cost 0.9% lower than that of lead time.

After investigating project cost performance that reflects the overall effort devoted to the NPD project, how the amount of functional effort contributed by each participating department is affected by different LCE, RL, and OS levels is further examined.Three major conclusions can be drawn by breaking down the overall committed effort into functional effort contributed by each department:

1. From *Fig. 7-a*, we observe that differences between the committed effort from the major department (i.e., Mfg Effort) of downstream phase (i.e., Production phase), and the efforts devoted by the other two departments (i.e., Mkt Effort & Eng Effort) drop dramatically from a sequential process *(a)* to concurrent processes *(b)* and *(c)* regardless of LCE or RL levels.

2. Moreover, from a relative perspective (*Fig. 7-b*), the percentage increase of Mfg Effort versus baseline is higher than those of Mkt and Eng Efforts in all sequential processes but *(A)–(2)–(a)*, in which Mfg Effort %Change = *72.5%* and is slightly lower than Mkt

Effort %Change = *75.6%*. However, in concurrent processes, an inverse relationship but of a much greater magnitude (especially at high RL level) is observed. That is, by starting downstream activities early with only preliminary information, concurrent engineering tends to alleviate the impacts of rework on activities in Production phase while intensifying those on activities in the two upstream phases. Although the concept of *cross–functional integration* has already been applied to the sequential process that allows engineers from Mfg Dept to be engaged early in both Concept and Design phases, which differentiates it from a traditional waterfall process, the impact of rework mostly occur in Mfg Dept. A concurrent process tends to shift rework risks and even out committed efforts among various functional areas owing to another critical characterization of concurrent engineering: *parallelization of activities*.

3. Mkt Effort undergoes the highest percentage of increase from when RL changes from low to high regardless of LCE or OS levels. Then is the Eng Effort. Mft Effort has the least amount of fluctuation across different scenarios.

LCE	RL (α, γ)	OS	(i) Lead Time (Days)	(I) Time %Change c/w BL1	(ii) Project Cost ($\$ \times 1000$)	(II) PC %Change c/w BL1
(BL1) Baseline	No Rework	**(a)** 0%	119		7,168	
		(b) 33%	101		7,168	
		(c) 66%	81		7,169	
(A) *No LCE*	**(1)** *Low* $\alpha = \gamma = 0.3$	**(a)** 0%	158	32.0%	**10,781**	48.2%
		(b) 33%	160	58.9%	11,778	61.9%
		(c) 66%	**131**	62.6%	12,107	66.6%
	(2) *High* $\alpha = \gamma = 0.45$	**(a)** 0%	176	47.2%	**11,948**	64.2%
		(b) 33%	192	90.4%	14,542	99.8%
		(c) 66%	**162**	100.1%	14,927	105.4%
(B) $LCE = max\left(\left(\frac{1}{2}\right)^{N_{ij}-1}, 0.1\right)$	**(1)** *Low* $\alpha = \gamma = 0.3$	**(a)** 0%	141	17.6%	9,542	33.1%
		(b) 33%	129	28.1%	9,436	31.6%
		(c) 66%	**106**	31.0%	**9,185**	28.1%
	(2) *High* $\alpha = \gamma = 0.45$	**(a)** 0%	152	27.2%	**10,370**	44.7%
		(b) 33%	158	56.6%	12,044	68.0%
		(c) 66%	**121**	49.2%	11,037	54.0%

Table 3. Project Performance under the Impact of OS, RL and LCE

To better visualize the correlations between lead time and effort, scatter plots of 200 model replicates' lead time and total effort outcomes under different levels of OS and RL are demonstrated in *Fig. 8*. Red lines in the plots indicate the lead time and total effort required for BL1 baseline cases (an "ideally executed" project without accounting for rework).
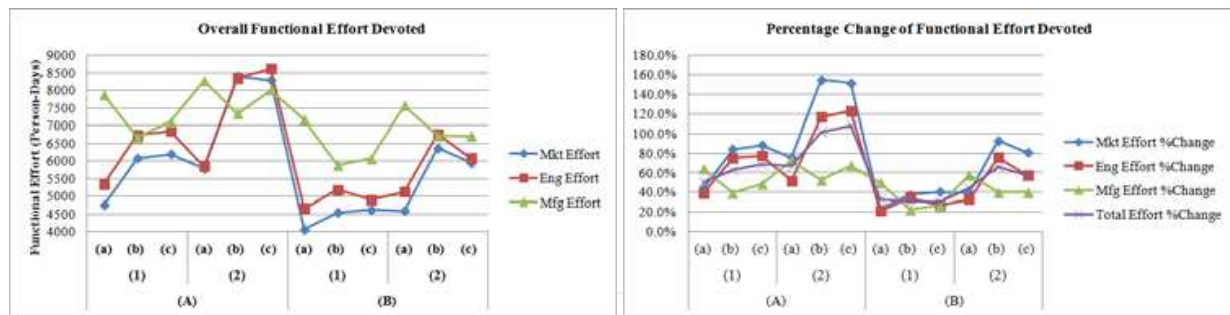
Figure 7. (a/b). Overall/ Percentage Change of Functional Effort Devoted
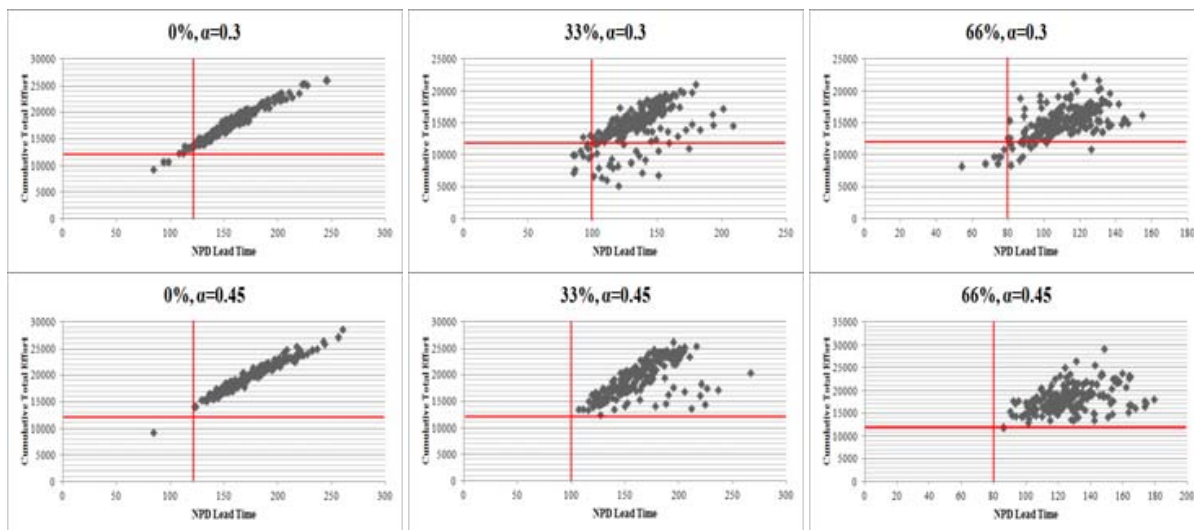

Figure 8. (a/b). Scatter Plots of the RL Impact on Different OS

We can clearly observe that a majority of replications exceed the lead time and effort of **BL1** by a considerable amount because of rework. Furthermore, as overlapping ratio and rework probability constants (α for IPC and γ for EPC) increase, there is also a notable increase in the number of replicates that are off the trend line. This phenomenon reveals that a high overlap ratio of upstream and downstream activities, combined with a high likelihood of unanticipated activity rework that requires additional resources will result in a strong tendency for NPD projects to behave in an unstable and unpredictable manner and lead to unforeseen departures from the predetermined baseline plan. Also note that, there exist possibilities where total effort or lead time or both are smaller than those required for the respective baseline cases, which is due to the stochastic nature of the model inputs (i.e., random inputs of activity duration, rework probabilities, etc.).

7. Conclusion

This research proposes a comprehensive discrete event simulation model that captures different aspects of PD project–related (i.e., product, process, team, and environment) complexity to investigate their resultant impacts on the occurrence and magnitude of

iterations and ECs that stochastically arise during the course of an NPD project, and how the multiple dimensions of project performance, including lead time, cost, and quality, are consequently affected. In addition to the integration of several critical characteristics of PD projects that have been previously developed and tested, (e.g., concurrent and collaborative development process, learning curve effects, resources constraints), this research introduces the following new features and dynamic structures that are explicitly modeled, verified, and validated for the first time:

1. This DES model *explicitly distinguishes between two different types of rework by the time of occurrence*: intra–phase iterations and inter–phase EECs. Moreover, *engineering changes are further categorized into two groups by their causes of occurrence*, emergent ECs "that are necessary to reach an initially defined standard in the product" [13], and initiated ECs in response to new customer requirements or technology advances.

2. *Uncertainty is differentiated and conceptualized into three categories*. Activity uncertainty is reflected in the stochastic activity duration using probability distributions, and environmental uncertainty is primarily modeled by the arrival frequency and magnitude of IECs. In particular, solution uncertainty is an important model variable that dynamically determines the rework probability which will be discussed next.

3. This study provides presumably the first attempt to integrate cause–and–effect relationships among project variables into a DES model of PD projects. Traditional DES model deals with only static project features in "open–loop, single–link" causal relationship format [14] that remain constant as the model evolves. *Rework probability is no longer pre–determined* and remains fixed over the entire time frame of the NPD process as appeared in most of previous studies. Instead, it is calculated in real time by the model itself. That is to say, rework probability is now included in a *feedback structure* that changes over time in response to the project's evolving uncertainty levels.

4. The specific three–step *rework review process structure*, together with the *rigidity of rework reviews*, allows more explicit and detailed modeling of this critical aspect of ECM, which is not attempted by previous studies. Decision points are used with rules to conditionally process ECs. They also give the users flexibility to define one or more rules in priority evaluation order.

5. *The traditional restrictive assumption of a stable development process with no environmental disturbance is also relaxed* by introducing the random occurrence of IECs, which will lead to an enlarged design solution scope of the final product and thus affecting the project solution uncertainty.

Results show under different conditions of uncertainty, how we should apply various kinds of strategies and policies, including process overlapping, rework review, IEC batching, resource allocation, to not only achieve benefits but also recognize potential tradeoffs among lead time, cost and quality. The study concludes with the following observations or understandings that either have been identified previously in the existing literature or disclosed for the first time with the help of newly added and verified model features:

1. Significant increase of both time and cost due to rework is alleviated by the evaluation of LCE.

2. The percentage increase of project cost is always higher than that of lead time at the occurrence of rework and IECs. That is, compared with lead time, project cost is more sensitive to rework/IECs.

3. By starting downstream activities early with only preliminary information, concurrent engineering tends to alleviate the impacts of rework on activities in downstream phases while intensifying those on activities in the upstream phases. It also tends to shift rework risks and even out committed efforts among various functional areas. In addition, departments that are majorly involved in upstream phases undergo higher fluctuation in effort.

4. A high overlap ratio of upstream and downstream activities, combined with a high likelihood of unanticipated activity rework that requires additional resources will result in a strong tendency for NPD projects to behave in an unstable and unpredictable manner and lead to unforeseen departures from the predetermined baseline plan.

5. Adopting a more restrictive *RRS* (Convex–Up) leads to a longer NPD lead time and higher project cost. There is no obvious distinction between Stepped Linear and Linear *RRSs*. Also, the evaluation of *LCE* reduces the impacts of *RRS*.

6. When only the IEC process propagation among development activities is examined, high correlations between lead time, cost, and quality are observed. However, when the effects of IEC product propagation among dependent product components/systems, the correlation between lead time and project cost, and the one between lead time and quality drop significantly.

7. Batching of IECs possesses a competitive advantage in lead time over handling IECs individually. This superiority is the greatest when a sequential PD process is adopted, and reduces as overlapping ratio increases. However, there is neither IEC policy shows "dominant" advantage in project cost or quality.

8. Potential tradeoffs among NPD lead time and total cost are clearly identified when resource assignment decision is to be made. A higher level of OS leads to a shorter NPD lead time and less total cost given the same amount of functional resource allocation. However, the benefits of lead time reduction by assigning more resources are the most obvious in a sequential process, and activity overlap reduces the degree of obviousness the benefits have. The higher the OS, the less the benefits.

9. Linearity between lead time and quality is observed in all three OS levels: the higher the functional resource availability, the shorter the lead time, and the lower the quality. The linearity slope increases as the *OS* increases. The percentage of decrease in quality versus baseline case is the largest in a sequential process and decreases as *OS* increases.

10. The evaluation of IEC product propagation leads to a general increase of the multiple dimensions of NPD project performance from baseline case, except a counterintuitive decrease in NPD project lead time for a less coupled product configuration under a high environmental uncertainty and a high *RL*.

Three possible main directions of future studies beyond the work presented here are summarized as follows:

1. Model features including: i) different relationships between solution uncertainty and rework probability, ii) more detailed modeling of dynamic rework review criteria (in replace of the current static one), and iii) parallel rework policy need to be tested to assess their impacts on project performance measures.
2. The review of literature has indicated a lack of development process models that are capable to be extended and implemented into a multi–project environment while still keeping detailed aspects of project complexity. Building blocks of the model framework can be reconfigured and applied at various detail levels. From a single project level to the entire organizational level, it opens possibilities for further analyses of multi–project management, such as work force planning strategies, coordination policies of interdependent parallel projects, etc.
3. This DES model can also be further extended across organizations. By relaxing the single organization restriction of the current model and including inter–organizational influences, how engineering changes propagate along supply chain and affect NPD project performance can be explored.

Author details

Weilin Li
Syracuse University, The United States

8. References

[1] Balakrishnan, N., A. K. Chakravarty. (1996). Managing Engineering Change: Market Opportunities and Manufacturing Costs. *Production and Operations Management*. 5 (4):335–356.

[2] Barzizza, R., M. Caridi, R. Cigolini. (2001). Engineering Change: A Theoretical Assessment and A Case Study. *Production Planning & Control*. 12 (7):717–726.

[3] Bhuiyan, N., D. Gerwin, V. Thomson. (2004). Simulation of the New Product Development Process for Performance Improvement. *Management Science*. 50(12):1690–1703.

[4] Bhuiyan, N., G. Gregory, V. Thomson. (2006). Engineering Change Request Management in a New Product Development Process. *European Journal of Innovation Management*. 9(1):5–19.

[5] Black, L. J., N. P. Repenning. (2001). Why Firefighting Is Never Enough: Preserving High–Quality Product Development. *System Dynamic Review*. 17(1):33–62.

[6] Bouikni, N., A. Desrochers. (2006). A Product Feature Evolution Validation Model for Engineering Change Management. *Journal of Computing and Information Science in Engineering*. 6(2):188–195.

[7] Browning, T. R. (1998). Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development. Doctoral thesis. Massachusetts Institute of Technology. Cambridge, MA.

[8] Browning, T. R., S. D. Eppinger. (2002). Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development. *IEEE Transactions on Engineering Management*. 49(4): 428–442.

[9] Cho, S. H., S. D. Eppinger. (2005). A Simulation–Based Process Model for Managing Complex Design Projects. *IEEE Transactions on Engineering Management*. 52(3):316–328.

[10] Clark, K. B., T. Fujimoto. (1991). Product Development Perfornmance: Strategy, Organization, and Management in the World Auto Industry. Boston, Mass.: Harvard Business School Press.

[11] Clarkson P. J. and C. Eckert. (2004). Design and Process Improvement: A Review of Current Practice. Springer. 1st edition.

[12] Earl C., J. H. Johnson and C. Eckert. (2005). "Complexity" in *Design Process Improvement – A Review of Current Practice*. 174-196, Springer, ISBN 1-85233-701-X, 2005

[13] Eckert, C., P. J. Clarkson, W. Zanker. (2004). Change and Customisation in Complex Engineering Domains. *Research in Engineering Design*. 15(1):1–21.

[14] Ford, D. N. (1995). The Dynamics of Project Management: An Investigation of the Impacts of Project Process and Coordination on Performance. Doctoral thesis. Sloan School of Management. Massachusetts Institute of Technology. Cambridge, MA.

[15] Ford, D. N., J. D. Sterman. (1998). Dynamic Modeling of Product Development Processes. *System Dynamics Review*. 14(1): 31–68.

[16] Ford, D. N., J. D. Sterman. 2003. Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects. *Concurrent Engineering: Research and Applications*. 11(3):177–186

[17] Ha, A. Y., E. L. Porteus. (1995). Optimal Timing of Reviews in Concurrent Design for Manufacturability. *Management Science*. 41(9):1431–1447.

[18] Hegde, G. G., Sham Kekre, Sunder Kekre. 1992. Engineering Changes and Time Delays: A Field Investigation. *International Journal of Production Economics*. 28(3):341–352.

[19] Ho, C. J. 1994. Evaluating the Impact of Frequent Engineering Changes on MRP System Performance. International Journal of Production Research. 32(3):619–641.

[20] Ho, C. J., J. Li. 1997. Progressive Engineering Changes in Multi–level Product Structures. Omega: International Journal for Management Science. 25(5):585–594.

[21] Hobday, M. (1998). Product Complexity, Innovation, and Industrial Organization. *Research Policy*. 26(6):689–710.

[22] Huang, G. Q., K. L. Mak. (1999). Current Practices of Engineering Change Management in Hong Kong Manufacturing Industries. *Journal of Materials Processing Technology*. 19(1):21–37.

[23] Koh, E. CY., P. J. Clarkson. (2009). A Modelling Method to Manage Change Propagation. In *Proceedings of the 18th International Conference on Engineering Design*. Stanford, California.

[24] Krishnan, V., S. D. Eppinger, D. E. Whitney. (1997). A Model–Based Framework to Overlap Product Development Activities. *Management Science*. 43(4):437–451.

[25] Krishnan, V., K. T. Ulrich. (2001). Product Development Decisions : A Review of the Literature. *Management Science*. 47(1):1-21.

[26] Lin, J., K. H. Chai, Y. S. Wong, A. C. Brombacher. (2007). A Dynamic Model for Managing Overlapped Iterative Product Development. *European Journal of Operational Research*. 185:378-392

[27] Loch, C. H., C. Terwiesch. (1999). Accelerating the Process of Engineering Change Orders: Capacity and Congestion Effects. *Journal of Product Innovation Management*. 16(2):145–159.

[28] Loch, C. H., J. Mihm, A. Huchzermeier. (2003). Concurrent Engineering and Design Oscillations in Complex Engineering Projects. *Concurrent Engineering: Research and Applications*. 11(3):187–199.

[29] Lyneis, J. M., D. N. Ford. (2007). System Dynamics Applied to Project Management: a Survey, Assessment, and Directions for Future Research. *System Dynamics Review*. 23(2/3): 157–189.

[30] Park, M., F. Peña–Mora. (2003). Dynamic Change Management for Construction: Introducing the Change Cycle into Model-Based Project Management. *System Dynamics Review*. 19(3):213-242.

[31] Pimmler, T. U., and S. D. Eppinger. (1994). Integration Analysis of Product Decompositions. In *Proceedings of the ASME Design Theory and Methodology Conference*, 343–351. Minneapolis, Minnesota: American Society of Mechanical Engineers.

[32] Reichelt, K., J. Lyneis. (1999). The Dynamics of Project Performance: Benchmarking the Drivers of Cost and Schedule Overrun. *European Management Journal*. 17(2):135-150.

[33] Roemer, T. A., R. Ahmadi. (2004). Concurrent Crashing and Overlapping in Product Development. *Operations Research*. 52(4): 606–622.

[34] Smith, R. P., S. D. Eppinger. (1997). A Predictive Model of Sequential Iteration in Engineering Design. *Management Science*. 43(8):1104–1120.

[35] Terwiesch, C., C. H. Loch. 1999. Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development. Journal of Product Innovation Management. 16(2):160–172.

[36] Unger, D. W., S. D. Eppinger. 2009. Comparing Product Development Processes and Managing Risk. International Journal of Product Development. 8(4):382–401.

[37] Wynn, D. C., K. Grebici, P. J. Clarkson. 2011. Modelling the Evolution of Uncertainty Levels during Design. International Journal on Interactive Design and Manufacturing. 5:187–202.