Computers & Industrial Engineering 119 (2018) 321–337

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Product development network modelling extensions to the cycle elimination method

Omar Abou Kasm*, Ali Yassine

Department of Industrial Engineering and Management, Faculty of Engineering and Architecture, American University of Beirut, P.O. Box 11-0236, Beirut 1107 2020, Lebanon

ARTICLE INFO

Keywords:
Project networks
Product development
Activity rework
Iteration
Design structure matrix (DSM)
Cycle elimination method
Product development case studies

ABSTRACT

This paper considers Product Development (PD) project networks, which are characterized by stochastic activity durations and activity rework or iteration (i.e., potential to repeat some activities several times during PD execution). The Cycle Elimination (CE) approach presented in Nasr et al. (2016) reduces the computational complexity of analyzing iterative PD project networks by considering an approximate network with no iteration. We build on the CE approach to investigate practical scenarios which arise in real world PD projects which are not accounted for by the CE approach. These scenarios include: (i) forward probabilities, (ii) dynamic rework probabilities and proportions, (iii) multiple dependency relationships between activities, and (iv) different rework through indirect connections. We demonstrate these extensions using two case studies. The first case study considers a software development process, where we collected the data by interviewing the managers of the company. The second case study involves a hardware development process (adapted from Pinkett (1998)), where the results show that the proposed method outperformed three existing techniques from the literature. Both cases were solved using the proposed modification to the CE approach, and then simulated to gauge the accuracy of the proposed method showing very promising results.

1. Introduction

Product development (PD) projects are notorious for their iterative nature, where ignoring rework potential results in inaccurate estimates of project duration (and cost) and can lead to misleading analysis and managerial decisions (Browning & Yassine, 2016; Meier, Browning, Yassine, & Walter, 2015). Iterative rework is denoted by a feedback loop in an Activity on Node (AON) representation of the PD project network, where the completion of a downstream activity may cause one or more upstream activity to be reworked (Yassine & Braha, 2003). The stochastic nature of the activity duration along with the probabilistic occurrence of feedback loops, significantly increases the complexity of estimating the duration of the PD project (Browning & Ramasesh, 2007; Unger & Eppinger, 2009). Feedbacks are a typical characteristic of any complex design and development project and a potential source of design iterations, which can account for one-third to two-thirds of the project duration and cost (Meier, Yassine, & Browning, 2007). This fact makes the study of project management in the presence of iteration, as suggested in this paper, a central issue for the PD community.

In the absence of stochastic feedback, the PD network reduces to a classical project network where traditional and well-established techniques can be utilized such as the critical path method (CPM) and program evaluation and review technique (PERT) (Mantel, Meredith, Shafer, & Sutton, 2007; Pinto, 2012). When considering project networks which exhibit feedback, the majority of the literature utilizes simulation techniques (e.g., Abdelsalam & Bao, 2006; Browning & Eppinger, 2002; Cho & Eppinger, 2005) or heuristic algorithms (e.g., Browning & Yassine, 2016; Jun, Park, & Suh, 2006) to estimate the duration of the project. Analytical approaches to approximate the expected duration of PD projects exist but not without limitations; for example, the Reward Markov Chain (RMC) approach (Smith & Eppinger, 1997) and the Signal Flow Graph (SFG) approach (Eppinger, Nukala, & Whitney, 1997) are both used for sequential PD networks. More recently, the Cycle Elimination (CE) approach (Nasr, Yassine, and Abou Kasm (2016)) investigated the duration of a PD network for sequential and parallel networks. The CE method uses the RMC approach as a starting point and is extended to include finding the expected duration and variance of sequential, parallel, and mixed (i.e., combination of sequential and parallel activities) activity networks. The CE method mainly works by transforming the PD network into a traditional network (i.e. eliminating feedback) and then traditional project management techniques such as CPM and PERT can be used to calculate the

* Corresponding author.
E-mail address: oia03@mail.aub.edu (O. Abou Kasm).

https://doi.org/10.1016/j.cie.2018.04.002
Received 28 June 2017; Received in revised form 14 February 2018; Accepted 2 April 2018
Available online 05 April 2018
0360-8352/

expected duration and variance of the network.

Pinkett (1998) also implemented modifications to previous analytical methods, namely the signal flow graph (SFG) and the RMC. The modifications include rework proportions (i.e. repeating a fraction of the original activity duration when feedback is triggered). Another modification is accounting for "terminal probabilities". The terminal probability as explained by Pinkett (1998) is the probability that a certain activity will have to be reworked after the downstream activity, responsible for triggering the rework, is worked a second time (or more). Also, a modification to account for forward probabilities (p_f), the probability to skip an activity in the first iteration, is discussed. The terminal probability (identified as dynamic probability in our work) and the forward probability modifications in the case study presented by Pinkett (1998) inspired us to investigate further real case scenarios through a real world case study of our own accompanied by discussions with managers of a product development company. Pinkett's modifications along with different real case complications that were discovered are presented and discussed in this paper.

After reflecting on the different real case PD scenarios, we noted that additional complications can create limitations or inaccuracies in the original CE approach, if the method is used without some modification. Thus, this paper is dedicated to extending the CE method in order to account for these different complications or special cases. The issues that are addressed in this paper are:

(i) *Forward probabilities* that exist when there is a chance to skip a certain activity from the first iteration. For example, consider an employee receiving outsourced material which needs to be inspected for paint scratches before being sent to the assembly department to become part of a final product. Thus, the employee, in this case, can either send the material to the painting department or skip this activity and send it directly to the assembly department if found acceptable during inspection.

(ii) *Dynamic rework probabilities and proportions* that exist when the rework probabilities and proportions change with successive iterations (generally decrease) and this can be justifiable due to learning. For example, consider an engineer submitting a design for her manager's review. Assume that the first review, having an occurrence probability of 70%, requires the engineer to fix certain aspects of the design requiring 60% of the time spent on the original design. After the latter fixes as requested by the manager, the probability to ask for successive modifications decreases along with the duration to fix them due to a better understanding of what is required.

(iii) *Multiple dependency relationships between activities* that exist when a certain activity triggers more than one type of rework from another activity. This is best explained by a manager asking a subordinate to repeat a certain design where the amount of rework depends on amount of errors the manager detected; that is, the rework can target minor adjustments or detailed adjustments which require much more time.

(iv) *Different rework through indirect connections* that exist when two or more activities have the potential to cause rework for the same activity but each requesting a different kind of rework, then when the reworked activity triggers rework for another activity, the latter rework will depend on the kind of rework initially requested. For example, consider two engineers that design a product sequentially. The first engineer is in charge of preliminary design (A), while the second engineer performs detailed design (B). Now, consider two quality assurance employees, where one is responsible for technical inspections (C) and the other for visual ones (D). Each can provide feedback for activity (A) and then (A) feeds information to (B). The rework required from (B) differs depending on the initial feedback; that is, whether it is initiated from (C) or from (D) to (A). Note that there is no direct connection between (C) or (D) and (B).

This paper is divided into five sections. Following this section, a literature review is provided in Section 2. Then, the extensions and modifications of the CE method are discussed in Section 3 with illustrative examples. In Section 4, two real case studies are presented. The first, from a software development company, is presented highlighting the different complications. The second, adopted from Pinkett (1998), is presented where we compare our approximations with Monte Carlo simulation results as well as the results obtained in Pinkett (1998). Finally, summary, discussion and conclusion are presented in Section 5.

2. Literature review

Different literature streams such as, Browning and Eppinger (2002), Cho and Eppinger (2005), and Abdelsalam and Bao (2006) discussed simulation techniques to find the expected duration of a PD project network. However, due to the time-consuming nature of simulation techniques, our interest in this paper is developing or extending existing analytical techniques to solve a wider range of project networks. Specifically, the paper aims for extending the Cycle Elimination (CE) method developed by Nasr et al. (2016). Thus, the literature of the CE method fundamentals is first presented before moving to the details of the proposed extensions.

The Signal Flow Graph (Eppinger et al., 1997) and the Reward Markov Chain (Smith & Eppinger, 1997) are two analytical techniques used to calculate the mean and variance of the PD network durations. However, they suffer from limitations such as tackling only sequential PD projects and not including rework proportions (Nasr et al., 2016). A signal flow network represents the activities by nodes, while the arcs leaving the nodes represent the different mutually exclusive choices after an activity is worked, meaning that each activity can have at most one predecessor and thus parallel work is not allowed. This assumption can be relaxed by adding additional states that represent activities in parallel. Finally, a sequence of activities, called a path is defined for the network and rework is considered by allowing an activity to appear more than once. On the other hand, the RMC approach uses a modified form of Gaussian elimination to calculate the expected duration of deterministic activity sequential networks with feedback (Nasr et al., 2016). A stage in the method is defined by the completion of an activity along with all feedback generated by the same activity. The RMC works in a regressive manner, it starts with the duration calculation of the final stage and works itself backwards until the first stage duration is calculated and then sums all durations. The two methods, signal flow graph and RMC converge to give the same expected duration (Pinkett, 1998), where the signal flow graph calculates the expected duration to pass through the network while the RMC calculates the expected duration spent in the network.

Nasr et al. (2016) extended the RMC to account for rework proportions in the expected duration calculations as well as finding the variance in sequential activity networks. Moreover, they extended the method to account for parallel and coupled activities. As such, their method, called the cycle elimination (CE) method, requires the following as inputs: Rework probabilities & proportions, distributions of the activity durations, and the sequence of working the activities (with identifying sequential, parallel, or coupled activities). When it comes to sequential networks, the CE method's algorithms and formulations are used to find the expected durations at every stage and then they are simply summed. However, a bigger role is played in mixed networks. The cycle elimination approach starts by modifying the project network to allow for removing the feedback and then analyzing the network using traditional project management techniques. Specifically, consider the probability DSM (Design Structure Matrix) in Fig. 1 and its network representation in Fig. 2. The probability DSM in Fig. 1 shows the activity durations (in the diagonal entries) and the activity connections by the presence of any value greater than zero in the off-diagonal entries, where these values are the associated probabilities. For example, there is a 31% chance that Activity 4 will cause rework to Activity 2. Note

	1	2	3	4	5	6
1	**74**					0.40
2	0.27	**20**		0.31		
3	0.25		**72**		0.50	
4		0.55		**41**		
5			0.29	0.17	**36**	
6					0.77	**59**

Fig. 1. 6 × 6 probability DSM.
Adapted from Nasr et al., 2016.

that the numbers in the lower triangular part of the matrix are only applicable when rework is triggered. For instance, working Activity 1 for the first time, will always result in tackling activities 2 and 3, however if Activity 6 triggers rework for Activity 1, then there is a 27% chance to rework Activity 2 and a 25% chance to rework Activity 3. The connections in the DSM are translated to a network without feedback in Fig. 2 allowing it to be solved by any traditional project management technique after the calculation of all node durations. Three types of nodes can be observed in this figure, the circles representing the activity's duration, the boxes representing the rework caused by an activity, and the doted boxes representing the stages which are the sum of an activity duration and the rework it generated. Since Activities 4, 5 and 6 have the potential to create rework as seen in the DSM, their stages are represented to contain rework nodes. Finally, the arcs represent the connections between activities in the first iteration, i.e. when the activities are worked for the first time, and they can be inferred from the lower triangle values in the DSM.

Using the CE algorithm, the stage expected durations and variances are calculated, and then the problem is tackled as a network without feedback. The user can then use any preferable traditional technique to solve such networks. Specifically, the simplified Eqs. (1) and (2) represent the set of equations required to find the expected duration $\mathrm{E}[T_k]$ at stage k.

$$\mathrm{E}[R_{ij;k}] = \mathrm{E}[W_{ij}t_j] + \sum_{u=1}^{k} \mathrm{E}[R_{ju;k}]P_{ju} \tag{1}$$

$$\mathrm{E}[T_k] = \mathrm{E}[t_k] + \sum_{u=1}^{k} \mathrm{E}[R_{ku;k}]P_{ku} \tag{2}$$

where the $R_{ij;k}$ is the duration required to complete stage k when activity j is requested for rework after activity i; P_{ij} is the associated probability. W_{ij} is the rework proportion associated with reworking activity j after activity i and t_j is the duration of the single activity k.

These equations result in a total of $k^2 + 1$ equations where the first k^2 are generated by Eq. (1) and the final equation is generated by Eq. (2). Similar equations are established to generate the second moment used to calculate the variance of each stage. The variance equations along with efficient algorithms to generate the matrices are shown in Appendix A.

The CE method is then extended to solve two special cases: coupled activities, and accounting for parallel rework. The math and algorithms are not changed in these extensions, however modifications in the input DSMs are made to capture the specific complication. These modifications along with the right choice of the stage duration generate more accurate results. Specifically, the modification for coupled activities involved the inclusion of a dummy activity in the DSM having the right connections and then including its stage duration in the final calculations. While the modification of accounting for parallel rework only involved changes in the rework probabilities input. The extensions provided in this paper use similar techniques, but applied in different ways, to capture other complications.

3. Cycle elimination method extensions

After applying the CE method on real case scenarios and examples, it was found that modifications for the input DSMs are required to achieve accurate results under certain complications. The CE method imposes a set of assumptions that when a certain case deviates from, the accuracy of the result decreases. In this section, we state such cases and discuss how to manipulate the input matrices in order to better capture the desired reality and restore the method's accuracy. We begin by explaining the different cases and providing examples in this section while tackling real cases relating to the discussed modifications in Section 4. Specifically, four issues are discussed in this section: (i) forward probabilities, (ii) dynamic rework probabilities and proportions, (iii) multiple dependency relationships between activities, and (iv) different rework through indirect connections. Moreover, since the required modifications are only in the inputs and not in the calculations/algorithms, the discussions and examples in this section focus on the DSM modifications required for each special case. To carry the calculations that follows, the equations introduced by Nasr et al. (2016) must be followed; specifically, the expected duration calculations are based on the set of Eqs. (1) and (2) discussed in the literature. For interested readers, we placed the required detailed calculations for each example in Appendix B. Additionally, Appendix A summarizes the necessary theory from Nasr et al. (2016) to carry the expected duration and variance calculations.

3.1. Forward probabilities

Consider three sequential activities: i, j and k. The forward probability "p_f" denotes the probability to work activities j and k sequentially after Activity i and "1-p_f" is the probability to skip Activity j and directly work on Activity k (see Fig. 3(a)).

Pinkett (1998) discussed a modification for the RMC approach to account for the forward probabilities. The same modification can be used in the CE method: Activity j is given the expected duration "p_f times the original duration". Fig. 3(b) demonstrates this adjustment. If

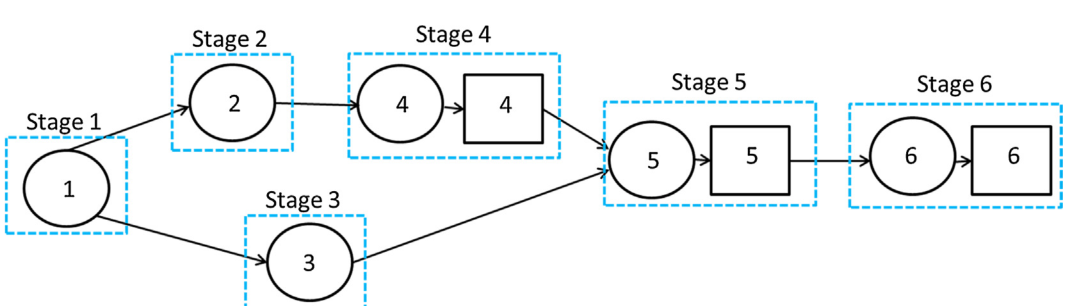

Fig. 2. Network with rework nodes.
Adapted from Nasr et al., 2016.

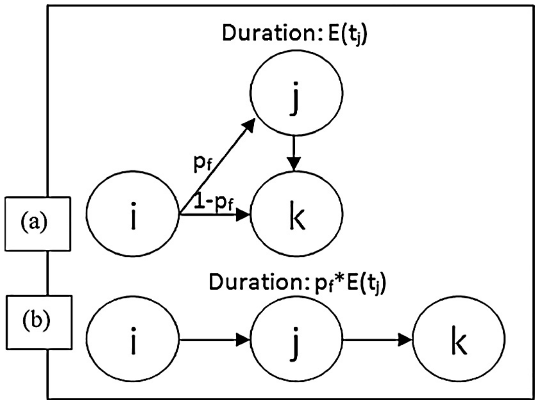

Fig. 3. Forward probability adjustment.

this case is represented without the adjustment, then the forward probability will be in the lower triangular part of the DSM. As such, it will not be considered in the first iteration calculations as discussed in the literature review. This idea makes the adjustment necessary for the RMC and CE approaches.

As an example, consider network (a) shown in Fig. 4 with the given information (durations, probabilities, and proportions) in Table 1. A forward probability is shown between activities 1 and 2. Then, to use the method, Activity 2 must be given the expected duration value of p_f multiplied by the original expected duration as shown in Table 1. The proposed method assumes that the network is as shown in Fig. 4(b), but with the duration adjustment, it gives the accurate expected duration of the initial network in Fig. 4(a). Fig. 5 shows the DSMs, with entries extracted from the given information in Table 1, to be used in order to proceed with the calculations. For example, the probability of reworking activity 3 after activity 4, p_{43}, in Table 1 is translated to the rework probability matrix in Fig. 5. The expected duration calculated is obtained to be 188.45 with a standard deviation of 30.49. Note that at each stage, the required matrices are generated by the algorithms in Appendix A, and solved to find the stage expected parameters and finally summed. The calculation details are shown in Appendix B.

3.2. Dynamically changing rework probabilities and proportions

This special case is also mentioned in Pinkett (1998), where the "terminal probability" was defined as the probability that a certain activity will have to be reworked after the downstream activity (responsible for the rework) is worked a second time (or more). However, Pinkett only considered rework probabilities. But this idea can also be applied to rework proportions, and thus both are implemented in our proposed modifications to the CE method as discussed here.

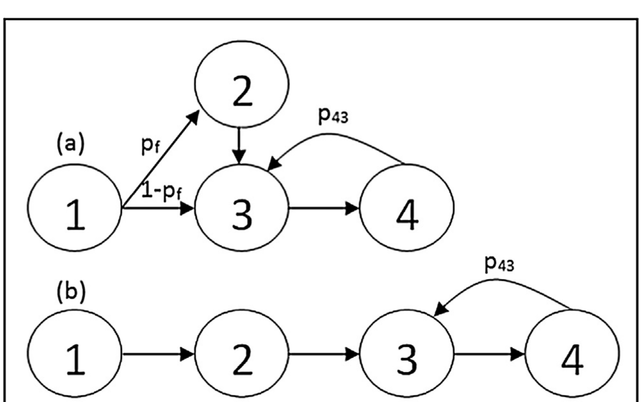

Fig. 4. Forward probability network (Example 1).

Table 1
Given information and adjusted duration (Example 1).

Activity	Duration	Forward and rework probabilities		Rework proportions	
1	56	p_f	0.6	w_{43}	0.4
2	70	p_{43}	0.45	w_{34}	0.3
3	40	Activity 2 Adjusted Duration (p_f * Duration$_2$)			
4	30	0.6 * 70 = 42			

To account for these probabilities and proportions, the user must add dummy activities representing these states. Assume that Activity j causes rework for Activity i with a probability "p_1" and rework proportion "w_1" when triggered the first time, but a probability and proportion "p_2" and "w_2" when triggered a second time. Then the user must create a dummy Activity j' located before the original Activity j, and all the preceding activities which are supposed to be connected to j will be connected to the dummy Activity j' (Activity i in our example) and j' will be connected to Activity i with the new rework probability and proportion "p_2" and "w_2" (see Fig. 6). This way, the stage duration of Activity j will trigger rework with probability p_1 and proportion w_1 but when j must be tackled again as a second order rework, the connections will lead to j' rather than j (as seen in Fig. 6, Activity i is only connected to Activity j') and thus the new probabilities and proportions are considered. All dummy activity stage durations must be ignored; because when created, they will automatically be accounted for in the rework calculations for the original activity stage duration.

As an example, consider the network in Fig. 7. It is given that when Activity 2 is worked for the first time, there is an 80% chance to repeat 40% of Activity 1; But when tackled a second time or more, there is a 30% chance to repeat 20% of Activity 1. 50% of Activity 2 is always repeated after Activity 1. Due to the dynamic change of probabilities, a dummy activity must be added for Activity 2 having the same duration as Activity 2. To find the expected duration, it is required to use the DSMs in Fig. 8 which represent the discussed information. The activity stage durations are summed (The dummy activity stage duration is neglected), and the obtained results show the expected duration as 207.3 with a standard deviation of 34.33. Note that at each stage, the required matrices are generated, by the algorithms in Appendix A, and solved to find the expected duration. Finally, the expected durations of all non-dummy activity stages are summed. The calculation details are presented in Appendix B.

3.3. Multiple dependency relationships between activities

Consider two activities: i and j. Activity j can generate more than one kind of rework for Activity i. For example; Activity i could be reworked with a probability "p" after Activity j but with a proportion "w_1" or "w_2" under the probabilities "p_1" and "p_2" respectively. As an example, a manager asking simple modifications as compared to asking extensive modifications from a design engineer. The expected time required to complete the extensive modifications will require a higher rework proportion of the original duration than the rework proportion required to complete the simple modifications.

Similar to Section 3.2, this idea can be accounted for by using dummy activities. Specifically, in the given example, dummy Activity i would be created and linked to Activity j with the required probabilities (i.e. $p * p_1$ for the rework proportion "w_1"). However, when the given probabilities and proportions are the same in every run (i.e. not dynamic, unlike the discussion in Section 3.2), then the use of the dummy activities is not necessary. The user can simply modify the inputted probabilities and proportions in an expected value manner. Specifically, the rework probability would stay "p" but the rework proportion would be "$p_1 * w_1 + p_2 * w_2$".

The same idea can be extended to the rework probability change depending on the kind of rework triggered. Specifically, when Activity j

Rework Probabilities (P)

	1	2	3	4
1				
2	1			
3		1		0.45
4			1	

Rework Proportions (W)

	1	2	3	4
1				
2	1			
3		1		0.4
4			0.3	

Durations
56
42
40
30

Fig. 5. Input DSMs (Example 1).

	i	j'	J
i		p_2	p_1
j'	p_{ij}		
j			

Fig. 6. Dynamic rework modification DSM.

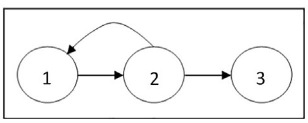

Fig. 7. Dynamic rework network (Example 2).

triggers rework for Activity *i*, Activity *j* is then reworked with a probability depending on the type of rework asked in the first place. The use of dummy activities can also be used. But, similar to the above, adjustments of the rework probabilities and proportions are a better method when they are invariable with number of times an activity is reworked.

Fig. 9 demonstrates the rework probability and proportions adjustments. The figure shows two rework types; but more can be available, and the rework probabilities and proportions can be found similarly (by finding the expected value).

As an example, consider the network shown in Fig. 10; when rework for Activity 1 is triggered after Activity 3 (65% of the time), it can be of two types. Each type has different rework probabilities and proportions (Table 2). Activity 3 is always reworked after Activity 2, but Activity 2's rework probability (after rework of Activity 1) depends on the type of rework triggered by Activity 3 on Activity 1. Fig. 11 shows the adjusted DSMs and the given durations to be used in the calculations. The obtained results show the expected duration as 280.63 with a standard deviation of 96. Note that at each stage, the required matrices are generated, by the algorithms in Appendix A, solved to find the stage expected duration and then summed. The necessary calculations are shown in Appendix B.

Matrices sample calculations:

$P_{12} = 0.6{*}0.65 + 0.4{*}0.45 = 0.57 \quad w_{23} = 0.6{*}0.4 + 0.4{*}0.5 = 0.44$

3.4. Different rework through indirect connections

The dynamic rework case (discussed in Section 3.2) can be available through indirect connections. Suppose that Activity *i* gives input to Activity *j*. Also, suppose that Activity *i* can have rework from two downstream activities *m* and *k*; then Activity *j* is connected indirectly

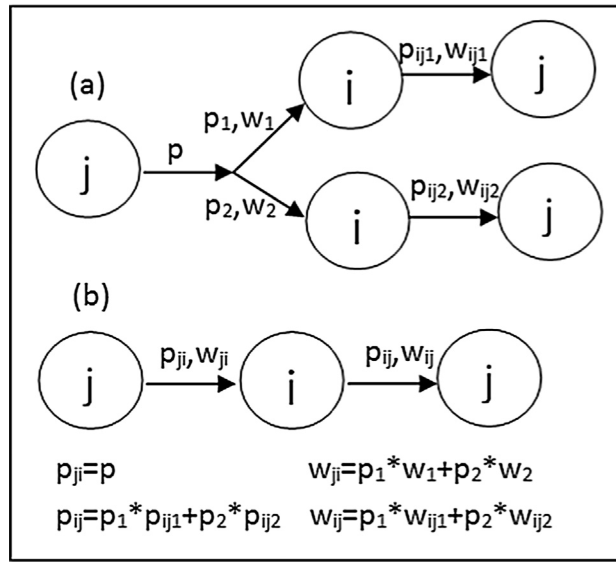

Fig. 9. Multiple dependency relationships adjustments.

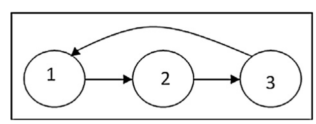

Fig. 10. Multiple dependency relationships network (Example 3).

Table 2
Rework types (Example 3).

	Percentage	p_{12}	w_{31}	w_{12}	w_{23}
Type 1	60	65	50	50	40
Type 2	40	45	30	70	50

with activities *m* and *k*. Activity *j* is then reworked in a probability and proportion "p_m" and "w_m" if Activity *i* is reworked after Activity *m* but reworked with "p_k" and "w_k" when Activity *i* is reworked after Activity *k*. Then the user must create two dummy activities for Activity *i*, connecting them to activities *m* and *k*, and each dummy activity is connected to Activity *j* with its respective probability and proportion. Figs. 12 and 13 show the network and the adjustment required.

As an example, consider the network in Fig. 14. When rework for Activity 1 is triggered, Activity 2 can be reworked as a second order rework. But, the rework probability and proportion of Activity 2

P	1	2'	2	3
1		0.3	0.8	
2'	1			
2				
3			0.2	

W	1	2'	2	3
1		0.2	0.4	
2'	0.5			
2				
3			1	

Durations
31
60
60
70

Fig. 8. DSMs (Example 2).

Rework Probabilities (P)

	1	2	3
1			0.65
2	0.57		
3		1	

Rework Proportions (W)

	1	2	3
1			0.42
2	0.58		
3		0.44	

Durations
81
56
71

Fig. 11. DSMs (Example 3).

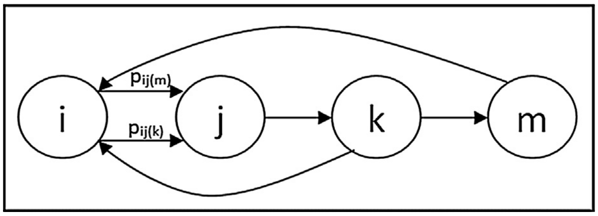

Fig. 12. Different rework through indirect connections network.

	$i^{(k)}$	$i^{(m)}$	j	k	m
$i^{(k)}$				p_{ki}	
$i^{(m)}$					p_{mi}
j	$p_{ij(k)}$	$p_{ij(m)}$			
k			p_{jk}		
m				p_{km}	

Fig. 13. Different rework through indirect connections DSM adjustment.

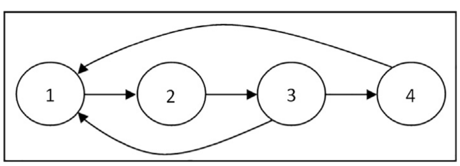

Fig. 14. Different rework through indirect connections network (Example 4).

depends on whether the rework of Activity 1 is generated by Activity 3 or Activity 4. Specifically there is an 80% chance of reworking 70% of Activity 2 or a 65% chance of reworking 50% of Activity 2 as a second order rework if Activity 1 is reworked after Activity 3 or Activity 4 respectively. Fig. 15 shows the adjusted DSMs; Two dummy activities (given the duration of Activity 1) are created, but their stage durations and variances are neglected in the network expected duration calculations (they are accounted for in the rework duration calculations of activities 3 and 4). The obtained results show the expected duration as 358.49 with a standard deviation of 178.41. Note that at each non-dummy activity stage, the required matrices are generated, by the algorithms in Appendix A, and solved to find the stage expected duration and then summed. Additional notes on the calculations are discussed in Appendix B.

4. Case studies

The method modifications are designed to tackle different complications that may rise in real PD environments and thus the method is best demonstrated by applying it to practical case studies. In this section, we tackle a case from a Lebanese company (Softex) in the software development field where we collected the data from the managers of the company. We then provide another case study from the literature (Pinkett, 1998) with the results compared with Pinkett's findings and a simulation for the network.

4.1. Softex case study

Each time Softex receives a project, the same development process (i.e. network) is followed; however the expected durations of the activities depend on the type of project received. The network has activities which can result in feedback to upstream activities (the probabilities and rework proportions also depend on the type of the project); thus making the Softex a suitable case to apply our methods on. For confidentiality reasons, the company name and activity names are changed. But all other data collected (durations, probabilities, rework proportions) are real and based on a specific type of project. Fig. 16 shows the network of the work process, with all the data given in Tables 3 and 4.

As shown in Table 4, rework probabilities depend on whether the activity is being done for the first time or not; and there are rework proportions between activities with indirect connections. For example, activity 18 can cause rework to activity 11 while activity 13 can cause rework to activities 11 and 12 with direct connections. Activities 11 and 12 can both generate rework to activity 13 and thus this activity (13) is claimed to have an indirect connection with activities 18 and 13 (here activity 13 is having an indirect connection with itself), but it will be worked with a proportion of 10% when the original feedback is from activity 18 and a proportion of 25% if the original feedback is from activity 13. Thus adjustments are required as discussed in Sections 3.2 and 3.4. Note that the indices of activities and dummy activities are carefully chosen to include all rework that maybe caused; for example Activity 15 is placed after Activity 18 to include the latter in the rework

Rework Probabilities (P)

	1	1'	1''	2	3	4
1						
1'	1				0.45	
1''						0.4
2		0.8	0.65			
3				1		
4					0.55	

Rework Proportions (W)

	1	1'	1''	2	3	4
1						
1'	1				0.9	
1''						0.8
2		0.7	0.5			
3				0.4		
4					0.6	

Duration
60
60
60
53
66
33

Fig. 15. DSMs (Example 4).

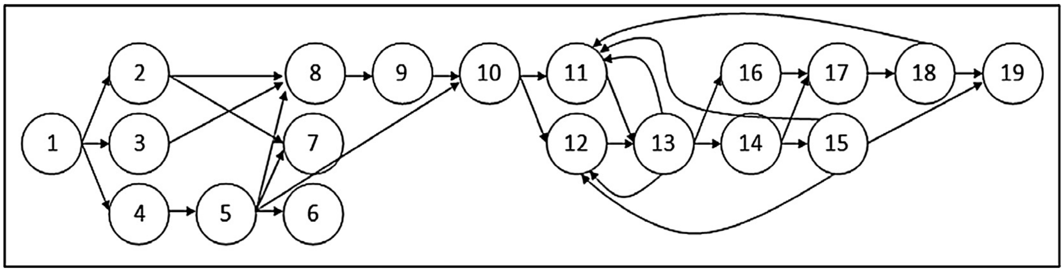

Fig. 16. Softex process network.

Table 3
Activities, durations, and inputs.

Task	Name	Type	Distribution (Hours)	Input From
1	Analysis of Clients Requirements	Analysis	UNIF(6,8)	–
2	Analysis of Technical Requirements	Development	UNIF(6,8)	1
3	Preliminary Visual Designing	Visual	12	1
4	Wire framing the Requirements	Analysis	UNIF(5,7)	1
5	Writing the Functional Document	Analysis	UNIF(7,9)	4
6	Reading the Functional Document and Writing the Task cases	Quality Control	UNIF(7,9)	5
7	Preliminary Development of Application (Basic Functionality)	Development	20	2,5
8	Complete Visual Development of Application	Visual	16	2,3,5
9	Supplementary Development of Application	Development	8	8
10	Executional Testing	Quality Control	UNIF(12,20)	5,9
11	Defect Fixing	Development	12	10,13,15,18
12	Visual Defect Fixing	Visual	8	10,13,15
13	Verification of Global Application	Quality Control	UNIF(8,16)	11,12
14	Deployment to Client-Based Environment	Development	4	13
15	Testing of Client-Based Environment	Quality Control	UNIF(3,5)	14
16	Automation Scripting	Development + QC	UNIF(8,12)	13
17	Automation Execution	Quality Control	8	14,16
18	Automation Results and Fixes	Development	UNIF(2,8)	17
19	Ensuring the Client-Based Environment	Analysis	UNIF(1,2)	15,18

caused by Activity 15.

Also there are activities that can be reworked in parallel, and thus priorities must be assigned as discussed by Nasr et al. (2016). The idea is that the activity with highest priority is the one responsible in deciding any additional rework, when called in parallel with other activities. Thus when there is no company policy in this decision, we need to give priority for the activity that generates the maximum amount of rework to avoid underestimating the network's expected duration. For two activities (i and j) being reworked in parallel after activity k (the only applicable case in this case study) and activity i is given priority over activity j, then $P_{k,i}$ remains the same and Eq. (3) defines the adjusted probability $P^{a}_{k,j}$.

$$P^{a}_{k,j} = P_{k,j}(1-P_{k,i}) \tag{3}$$

The below clarify the priority choices for the activities that can be reworked in parallel,

- Activities 11 and 12: Since both cause the same rework, but Activity 11 has a higher deterministic duration, Activity 11 is given priority.
- Activities 14 and 16: Since the rework after Activity 16 has a higher expected durations than the rework after Activity 14, Activity 16 is given priority.
- Activities 15 and 17: Since Activity 16 is given priority to Activity 14, and both activities are worked together with a 100% chance when Activity 13 is triggered, then the priority adjustment leaves Activity 14 with 0% chance of occurring in rework after Activity 13 in the adjusted DSM. This implies Activity 15 is not triggered in rework, then priority will not affect the solution and can be chosen at random.

The rework probability and proportion matrices are adjusted (they are posted in Appendix C for interested readers) and used with the first and second moments[1] (Table 5) to find the rework nodes expected durations and variances for the transformed network by the cycle elimination procedure (Fig. 17). Then, after simulating the no-feedback network, the expected duration is obtained to be 127.82 Hours with a standard deviation of 11.4. Finally, for comparison sake, the problem is simulated using ARENA which resulted in a small mean error (0.16%) as shown in Table 6.

4.2. Pinkett's case study

Pinkett (1998) tackled a case study for a hardware development process (designing and testing analog and digital PCBs) by the use of three methods. Fig. 18 shows the network required for the hardware development. Different prototypes undergo the same process, but the activity durations and rework probabilities differ for each prototype. Table 7 shows all the given activity durations and rework probabilities; moreover, it is given that all rework proportions are 50% of the original duration. Also, the terminal probabilities are set to 1%; which is the probability that a certain activity will have to be reworked after the downstream activity (responsible for the rework) is worked a second time (or more).

The first step is to write the matrices to be inputted in the CE method. Since all rework proportions are given to be 50%, then the rework proportion matrix (W) will have all elements (resembling the connections) equal to 0.5. For the rework probability matrix (P), other than the given input, adjustments are required to account for the

[1] The means and second moments are calculated using the following formulas (Casella & Berger, 2002): $E(X) = \frac{a+b}{2}$ & $E(X^2) = \frac{a^2 + ab + b^2}{3}$ Where a and b are the minimum and maximum values of the uniform distribution

Table 4
Probabilities and rework proportions.

Direct connections		Probabilities (%)		Rework proportions (%)
From	To	First Run	Other Runs	
13	11	75	25	50
	12	25	15	40
15	11	25	15	25
	12	10	–	15
18	11	5	–	10

Indirect connections		Rework proportions (%)
From	To	
13	13	25
15	13	10
	14	25
18	13	10
	14	25

Table 5
Activity durations.

Task	Mean	Second moment	Task	Mean	Second moment	Task	Mean	Second moment
1	7	49.33	10	16	261.33	14	4	16
2	7	49.33	11	12	144	16	10	101.33
3	12	144	12	8	64	17	8	64
4	6	36.33	11′	12	144	15′	4	16.33
5	8	64.33	12′	8	64	18′	5	28
6	8	64.33	13′	12	149.33	18	5	28
7	20	400	13	12	149.33	15	4	16.33
8	16	256	13″	12	149.33	19	1.5	2.33
9	8	64						

terminal probabilities (as discussed in Section 3.2): four dummy tasks are added. And finally, the duration matrix is adjusted to account for the forward probabilities (as discussed in Section 3.1). The acquired rework probability matrix and activity durations for prototype 1 are shown in Fig. 19 and Table 8 respectively. The matrix is populated by inputting the rework probability values where applicable. For example, activity 5 provides rework for activity 4 with a probability $p_3 = 0.3$; as such a value of 0.3 is placed at the intersection of the column of activity 5 with the row of activity 4.

Now, the CE method can be used to find the expected duration and variance. The dummy activity stage durations are neglected since they are added just to be considered in the rework calculations. The network is transformed using the Cycle Elimination procedure (as shown in Fig. 20) and the critical path of the parallel activities is chosen. Since all given durations are deterministic, there is no need for simulation to find the expected maximum of the parallel activities.

Table 9 shows the stage durations and variances of prototype 1. At each stage, the algorithms in Appendix A are used to generate the necessary matrices and carry the calculations. Since the durations are deterministic, then any stage without feedback is expected to have the activities duration with a variance of zero while activities with feedback will get a variance due to the associated probabilities such as the stage 5

Table 6
Case study results.

Method	Expected duration (Hours)	%Mean Error
CE	127.82	−0.16%
ARENA Simulation	128.02 (H-W < 0.98)	

duration. Similarly, prototypes 2 and 3 are solved, and all final results are shown in Table 10. To know where the proposed method stands, a simulation was done using ARENA to set a benchmark for the comparison with the results of the signal flow method (SFM) (which generates exact results as the RMC) and the modifications discussed by Pinkett (Dynamic SFM and Adjusted Markov). The comparison is shown in Table 10 and infers the accuracy of the proposed method. For example, the expected duration using the proposed method is equal to 279.9 days for prototype 1 which is 0.32% away from the simulation result (280.8), while the other methods are 50%, 7%, and 47% away from the simulation result. Prototypes 2 and 3 also accurate results with 0.04% and 0.18% mean errors for the proposed method while the others have larger deviations.

5 Summary, discussion and conclusion

Finding a good estimate of the expected duration (and cost) to develop complex products can be crucial for project managers as they plan for resources and provide clients with estimated completion times. Poor resource planning, staffing decisions, and commitments can result from inaccurate estimates of project duration. Additionally, contractual penalties may arise for noncompliance with promised delivery times. Simulation models to find the expected project duration can be time consuming to design, build, and run. Thus, our focus in this paper is on analytical techniques which can generate the results more promptly.

The cycle elimination (CE) method (Nasr et al., 2016) is an analytical technique which can be used to solve sequential and mixed networks (combination of sequential and parallel activities). The method

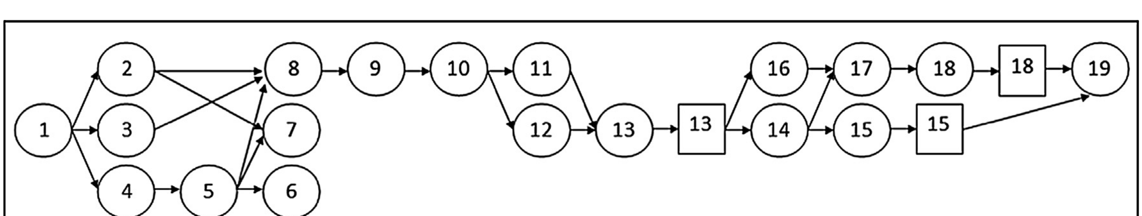

Fig. 17. Softex network without feedback.

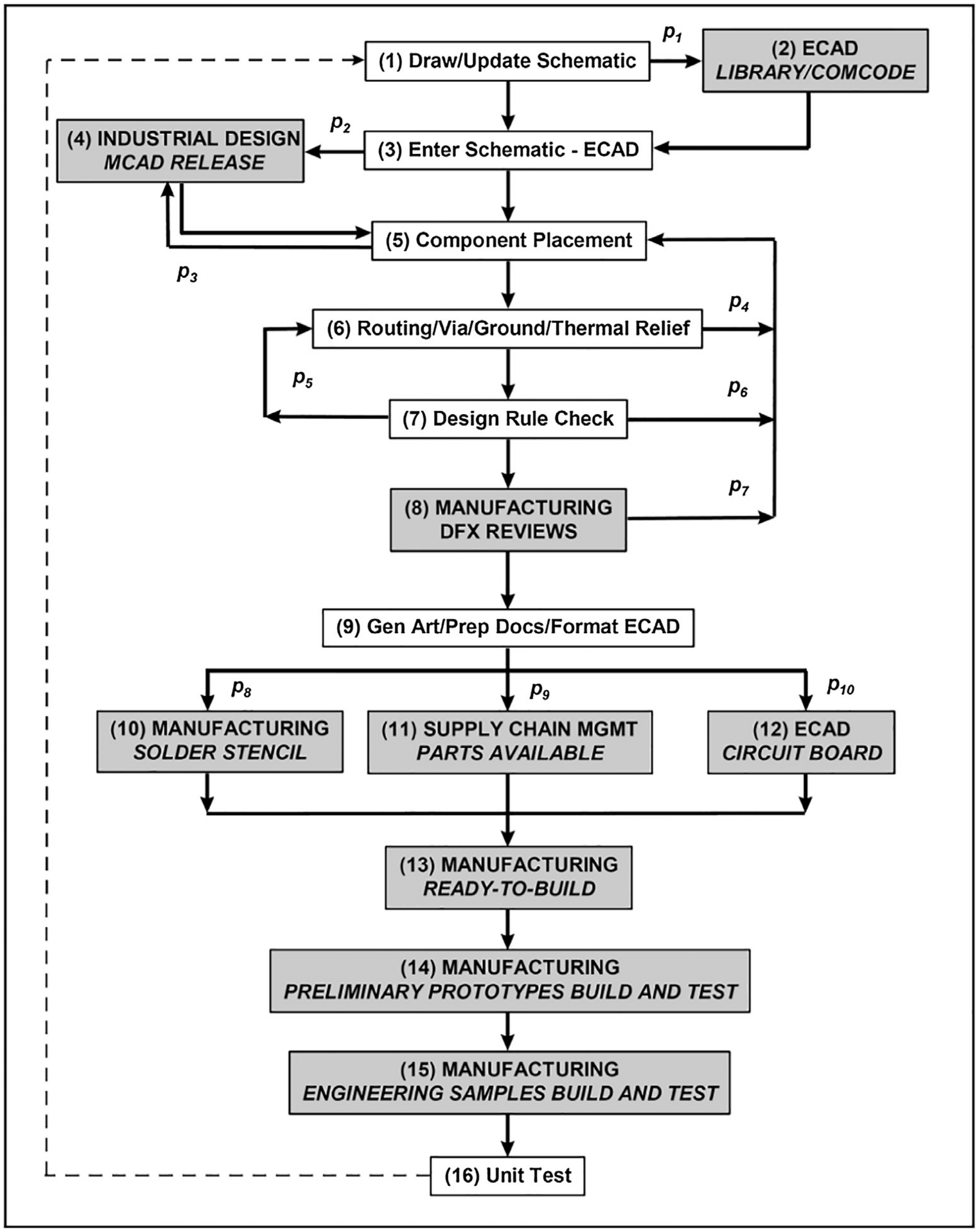

Fig. 18. Process network.
Adapted from Pinkett (1998).

tackles PD networks having feedback possibilities by transforming the network to a non-cyclic network in order to proceed with traditional project management techniques to find the expected duration. The method uses DSM to represent the network and finds the stage expected durations based on the DSM inputs, where the stages represent the activity's expected duration and their possible rework. Modifications to DSM are necessary for the method capture different complications. Two special cases, coupled activities and allowing parallel rework, were discussed by Nasr et al. (2016) and in this paper we extended the method to account for four additional complications (i) forward probabilities, (ii) dynamic rework probabilities and proportions, (iii) multiple dependency relationships between activities, and (iv) different rework through indirect connections.

The CE method can be used to solve any kind of complication with the right DSM modifications. The modifications can be in the rework probability and proportion inputs, adding dummy activities, or reordering the activities to capture the complication properly and reach the desired results. The reordering of activities is seen in the Softex case study; it is used when an activity may generate indirect rework for another activity (the two activities being in parallel tracks in the first iteration), then the latter's DSM index must be lower. The two other modification techniques are well described in Sections 3.1–3.4 of this paper. When more complications arise, more adjustments are required to achieve accurate results. For that, it is recommended that a software tool is created as a future work for using the method efficiently. The program would take the rework probabilities, rework proportions, durations (first and second moments), and any special feature as input from the user, creates the matrices automatically with the right adjustments, and generates the expected duration results of the whole network.

Generally, as seen in the case studies discussed in this paper, product development companies have the same network with only the durations, rework probabilities and proportions changing with the change of the product type. Thus, programming the method for such

Table 7
Data for Pinkett's case study.

	Prototype		
Activity	**1**	**2**	**3**
1	30	22	4
2	40	0	0
3	4	0	0
4	26	0	0
5	34	32	44
6	8	2	2
7	0	0	0
8	4	2	2
9	0	0	0
10	6	80	8
11	60	0	154
12	6	6	8
13	2	16	4
14	4	12	4
15	14	16	38
16	46	22	24

(a) Activity Durations

	Prototype		
Probability (p)	**1**	**2**	**3**
1	0.5	0.2	0.05
2	0.8	0.4	0.05
3	0.3	0.2	0.05
4	0.4	0.1	0.1
5	0.2	0.2	0.1
6	0.2	0.2	0.1
7	0.3	0.1	0.1
8	0.05	0.05	0.08
9	0.9	0.9	0.84
10	0.05	0.05	0.08

(b) Rework Probabilities

	1	2	3	4	5'	5	6'	6	7'	7	8'	8	9	10	11	12	13	14	15	16
1																				
2	0.5																			
3	0.5	1																		
4			0.8		0.01	0.3														
5'			0.2	1																
5							0.01	0.4	0.01	0.2	0.01	0.3								
6'					0.99	0.7														
6									0.01	0.2										
7'							0.99	0.6												
7																				
8'									0.98	0.6										
8																				
9											0.99	0.7								
10													0.05							
11													0.9							
12													0.05							
13														1	1	1				
14																	1			
15																		1		
16																			1	

Fig. 19. Prototype 1 rework probability matrix (P).

Table 8
Prototype 1 activity duration matrix.

Activity	Adjusted Durations	Activity	Adjusted Durations	Activity	Adjusted Durations	Activity	Adjusted Durations
1	30	5	34	8′	4	12	0.3
2	20	6′	8	8	4	13	2
3	4	6	8	9	0	14	4
4	20.8	7′	0	10	0.3	15	14
5′	34	7	0	11	54	16	46

cases can be very easy by creating a specific spread sheet (and VBA coding) where the user only inputs the durations and probabilities/proportions, and the results are automatically generated. Then each time a new product is being considered, the spread sheets can be used to find the expected duration and variance, and the results will be obtained in no time.

Finally, as discussed earlier, the expected duration is essential to find when planning a certain product development project, or when

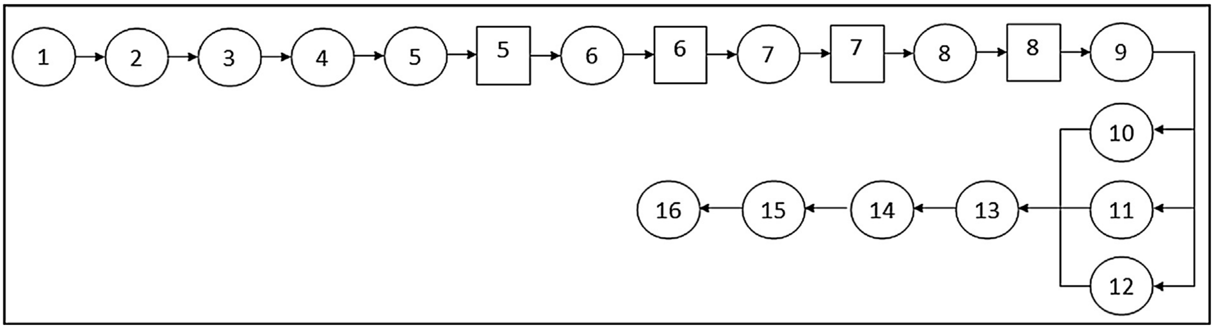

Fig. 20. Network with rework nodes.

Table 9
Prototype 1 stage durations and variances.

						Dummy activities[a]		
Stage	Stage Duration	Variance	Stage	Stage Duration	Variance	Stage	Stage Duration	Variance
1	30	0	9	0	0	5′	34.28	7.6
2	20	0	10	0.3	0	6′	8.296	10.41
3	4	0	11	54	0	7′	0.464	16.34
4	20.8	0	12	0.3	0	8′	4.324	12.41
5	42.3	163.2	13	2	0			
6	19.84	279.7	14	4	0			
7	9.273	245.1	15	14	0			
8	13.72	281	16	46	0			

[a] Dummy Activities' stages are neglected in the critical path calculations.

Table 10
Results and method comparisons.

		Expected duration [Stdev] (Days)	Simulation duration [Half-Width, Stdev] (Days)	% Mean Error
Prototype 1	Our Method	279.9 [31.13]	280.81 [< 1.13, 31.58]	−0.32%
	Static SFM	421.4 [14.99]		50.07%
	Dynamic SFM	261 [5.74]		−7.05%
	Adjusted Markov	413.6 [–]		47.29%
Prototype 2	Our Method	140.912 [14.77]	140.86 [< 0.64, 14.6]	0.04%
	Static SFM	167.4 [7.38]		18.71%
	Dynamic SFM	146.2 [7.33]		3.76%
	Adjusted Markov	166.2 [–]		17.86%
Prototype 3	Our Method	261.22 [14.45]	261.69 [< 1.56, 25.17]	−0.18%
	Static SFM	216 [7.33]		−17.46%
	Dynamic SFM	224.2 [5.25]		−14.33%
	Adjusted Markov	215.8 [–]		−17.54%

giving a client the lead time to finalize a requested product. But another important issue is optimizing the product development process by minimizing the expected duration. Prasad (1999), qualitatively described several collaboration techniques that can result in transferring sequential activities to parallel ones, and thus reducing the production's expected duration. Also, Hu, Liu, and Prasad (2003), discuss an execution plan to maximize the concurrency in a network in the effort of minimizing the expected duration. As future work, the cycle elimination (CE) method can be used to provide quantitative measures on the expected duration improvements that the collaborations and maximized concurrency can result in. As some execution methods may require additional costs to achieve a lower duration, and since concurrency may create rework risks (Yang, Zhang, & Yao, 2012), and since under high uncertainty, sequential networks may perform better than networks with parallel activities (Liu, Hisarciklilar, Thomson, & Bhuiyan, 2015), the CE method can be used to find the expected duration for each arrangement of activities and then analyze the results to identify the best collaboration and concurrency combinations that improve the duration with minimal cost.

Appendix A

Expected value simplified equations:

$$\mathrm{E}[R_{ij;k}] = \mathrm{E}[W_{ij}t_j] + \sum_{u=1}^{k} \mathrm{E}[R_{ju;k}]P_{ju} \tag{A.1}$$

Step 1: Set $\mathbf{A} = \text{zero}\left((k^2+1)\text{x}(k^2+1)\right)$

Step 2: For $i = 1, \ldots, k$
  For $j = 1, \ldots, k$
    For $u = 1, \ldots, k$
      Set $row = (i-1)k + j$
      Set $col = (j-1)k + u$
      Set $\mathbf{A}(row, col) = \mathrm{P}_{ju}$
  Set $row = k^2 + 1$
  For $u = 1, \ldots, k$
    Set $col = (k-1)k + u$
    Set $\mathbf{A}(row, col) = \mathrm{P}_{ku}$

Step 3: Set $\mathbf{A} = \mathbf{I} - \mathbf{A}$

Fig. A1. Algorithm to populate the entries of A.
Adapted from Nasr et al. (2016)).

For $i = 1, \ldots, k$
  For $j = 1, \ldots, k$
    Set $v = (i-1)k + j$
    Set $\mathbf{b_1}(v) = \mathrm{W}_{ij}\,\mathrm{E}[t_j]$
Set $\mathbf{b_1}(k^2+1) = \mathrm{E}[t_k]$

Fig. A2. Algorithm to populate the entries of b1.
Adapted from Nasr et al. (2016)).

For $i = 1, \ldots, k$
  For $j = 1, \ldots, k$
    Set $v = (i-1)k + j$
    Set $\mathbf{b_2}(v) = \mathrm{W}_{ij}^2\,\mathrm{E}\left[t_j^2\right] + 2\mathrm{W}_{ij}\,\mathrm{E}[t_j]\left(\sum_{u=1}^{k}\mathrm{E}[R_{ju;k}]P_{ju}\right)$
Set $\mathbf{b_2}(k^2+1) = \mathrm{E}\left[t_k^2\right] + 2\mathrm{E}[t_k]\left(\sum_{u=1}^{k}\mathrm{E}[R_{ku;k}]P_{ku}\right)$

Fig. A3. Algorithm to calculate the elements of b2.
Adapted from Nasr et al. (2016)).

$$\mathrm{E}[T_k] = \mathrm{E}[t_k] + \sum_{u=1}^{k} \mathrm{E}[R_{ku;k}]P_{ku} \tag{A.2}$$

Second Moment Simplified Equations:

$$\mathrm{E}[R_{ij;k}^2] = \mathrm{E}[(W_{ij}t_j)^2] + \sum_{u=1}^{k} \mathrm{E}[2W_{ij}t_jR_{ju;k} + R_{ju;k}^2]P_{ju} \tag{A.3}$$

$$\mathrm{E}[T_k^2] = E[t_k^2] + \sum_{u=1}^{k} \mathrm{E}[2t_kR_{ku,k} + R_{ku,k}^2]P_{ku} \tag{A.4}$$

where $\mathrm{E}[T_k]$ and $\mathrm{E}[T_k^2]$ is the expected duration and second moment respectively for stage. $R_{ij;k}$ and $R_{ij,k}^2$ are the duration and second moment required to complete stage k when activity j is requested for rework after activity i; P_{ij} is the associated probability. W_{ij} is the rework proportion associated with reworking activity j after activity i and t_j is the duration of the single activity k. $\mathrm{E}[t_k^2]$ is the second moment of the single activity k.

$\mathbf{AX_1} = \mathbf{b_1}$ and $\mathbf{AX_2} = \mathbf{b_2}$ are system of linear equations to be solved to obtain the expected duration and the second moment at each stage k. Where $\mathbf{A}$, $\mathbf{b_1}$, and $\mathbf{b_2}$ (algorithms in Figs. A1–A3) are the required matrices to carry the calculations. $\mathbf{X_1}$ and $\mathbf{X_2}$ represent the arrays of expected durations and second moments.

Appendix B

B.1. Example 1 calculations

The matrices in Fig. A4 are the result of the CE method extensions discussed in example 1. Now, we need to calculate the expected duration and variance at each stage using the CE method's equations, and then sum them to get the parameters of the whole network. We can easily determine the durations and variances of the first three stages as they generate no feedback and they take the values of their respective activity parameters. For this

Rework Probabilities (P)

	1	2	3	4
1				
2	1			
3		1		0.45
4			1	

Rework Proportions (W)

	1	2	3	4
1				
2	1			
3		1		0.4
4			0.3	

Durations
56
42
40
30

Fig. A4. Example 1 input matrices.

example, stages 1, 2, and 3 take the expected duration of 56, 42 and 40 respectively with a variance of 0 as they are deterministic. For stage 4, the CE method's equations are necessary and they are represented in the below matrices (populated by the algorithms in Figs. A1 and A2) where $\mathrm{E}[T_4]$ is the required value representing the expected duration of stage 4. Solving the set of equations results in $\mathrm{E}[T_4] = 50.45$ and thus the total expected duration of the network is 188.45.

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -0.45 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{E}[R_{11;4}] \\ \mathrm{E}[R_{12;4}] \\ \mathrm{E}[R_{13;4}] \\ \mathrm{E}[R_{14;4}] \\ \mathrm{E}[R_{21;4}] \\ \mathrm{E}[R_{22;4}] \\ \mathrm{E}[R_{23;4}] \\ \mathrm{E}[R_{24;4}] \\ \mathrm{E}[R_{31;4}] \\ \mathrm{E}[R_{32;4}] \\ \mathrm{E}[R_{33;4}] \\ \mathrm{E}[R_{34;4}] \\ \mathrm{E}[R_{41;4}] \\ \mathrm{E}[R_{42;4}] \\ \mathrm{E}[R_{43;4}] \\ \mathrm{E}[R_{44;4}] \\ \mathrm{E}[T_4] \end{pmatrix} \begin{pmatrix} 0 \\ 42 \\ 0 \\ 0 \\ 0 \\ 0 \\ 40 \\ 0 \\ 0 \\ 0 \\ 0 \\ 9 \\ 0 \\ 0 \\ 16 \\ 0 \\ 30 \end{pmatrix}$$

For the variance of stage 4, we use the CE method's equations to calculate the second moment and accordingly finding the variance. The below matrices (populated by the algorithms in Figs. A1 and A3) represent the required equations where $\mathrm{E}[T_4^2]$ is the second moment value of our interest. Solving the equations results in $\mathrm{E}[T_4^2] = 34754.4$, and thus $\mathrm{Var}[T_4] = \mathrm{E}[T_4^2]{-}\mathrm{E}^2[T_4] = 929.75$ and this is the variance of the whole network as the other variances are equal to zero. Thus the standard deviation is 30.49.

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -0.45 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.45 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{E}[R_{11;4}^2] \\ \mathrm{E}[R_{12;4}^2] \\ \mathrm{E}[R_{13;4}^2] \\ \mathrm{E}[R_{14;4}^2] \\ \mathrm{E}[R_{21;4}^2] \\ \mathrm{E}[R_{22;4}^2] \\ \mathrm{E}[R_{23;4}^2] \\ \mathrm{E}[R_{24;4}^2] \\ \mathrm{E}[R_{31;4}^2] \\ \mathrm{E}[R_{32;4}^2] \\ \mathrm{E}[R_{33;4}^2] \\ \mathrm{E}[R_{34;4}^2] \\ \mathrm{E}[R_{41;4}^2] \\ \mathrm{E}[R_{42;4}^2] \\ \mathrm{E}[R_{43;4}^2] \\ \mathrm{E}[R_{44;4}^2] \\ \mathrm{E}[T_4^2] \end{pmatrix} \begin{pmatrix} 0 \\ 7598.2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 3956.4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 449.2 \\ 0 \\ 0 \\ 1198.5 \\ 0 \\ 2127.3 \end{pmatrix}$$

B.2. Example 2 calculations

The matrices in Fig. A5 are the result of the CE method extensions discussed in example 2. Looking at the DSMs, 4 stages are represented (1, 2′, 2, and 3). Since stage 2′ is a dummy activity, it is not considered in the network calculations as a stage but it is necessary for the rework calculations in stage 2. Stages 1 and 3 do not generate any feedback, as such their expected durations are the given activity durations 31 and 70 respectively with a

P	1	2'	2	3
1		0.3	0.8	
2'	1			
2				
3			0.2	

W	1	2'	2	3
1		0.2	0.4	
2'	0.5			
2				
3			1	

Durations
31
60
60
70

Fig. A5. Example 2 input matrices.

P	1	2'	2
1		0.3	0.8
2'	1		
2			

W	1	2'	2
1		0.2	0.4
2'	0.5		
2			

Durations
31
60
60

Fig. A6. Stage 2 input matrices (Example 2).

variance of 0. Stage 2 requires the CE method's equations for its calculations. Since it is not the last stage in the DSM, the input matrices are adjusted by deleting all proceeding stages and represented in Fig. A6.

Now, the CE method equations are developed and represented by the matrices below (populated by the algorithms in Figs. A1 and A2) to calculate the expected duration of stage 2 ($\mathrm{E}[T_2]$). Solving the set of equations results in $\mathrm{E}[T_2] = 106.3$, and thus the expected duration of the network is 207.3 which is the sum of stage 1, 2, and 3 expected durations.

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -0.8 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.8 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.3 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.8 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.8 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{E}[R_{11;2}] \\ \mathrm{E}[R_{12';2}] \\ \mathrm{E}[R_{12;2}] \\ \mathrm{E}[R_{2'1;2}] \\ \mathrm{E}[R_{2'2';2}] \\ \mathrm{E}[R_{2'2;2}] \\ \mathrm{E}[R_{21;2}] \\ \mathrm{E}[R_{22';2}] \\ \mathrm{E}[R_{22;2}] \\ \mathrm{E}[T_2] \end{pmatrix} = \begin{pmatrix} 0 \\ 30 \\ 0 \\ 6.2 \\ 0 \\ 0 \\ 12.4 \\ 0 \\ 0 \\ 60 \end{pmatrix}$$

Then, we use the CE method's equations to calculate the second moment and find the variance of stage 2. The below matrices (populated by the algorithms in Figs. A1 and A3) represent the required equations to find the second moment of stage 2 ($\mathrm{E}[T_2^2]$). Solving the equations results in $\mathrm{E}[T_2^2] = 12485$, and thus $\mathrm{Var}[T_2] = \mathrm{E}[T_2^2]-\mathrm{E}^2[T_2] = 1178.5$ and this is the variance of the whole network as the other variances are equal to zero. Thus the standard deviation is 34.33.

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -0.8 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.8 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.3 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.8 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.8 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{E}[R^2_{11;2}] \\ \mathrm{E}[R^2_{12';2}] \\ \mathrm{E}[R^2_{12;2}] \\ \mathrm{E}[R^2_{2'1;2}] \\ \mathrm{E}[R^2_{2'2';2}] \\ \mathrm{E}[R^2_{2'2;2}] \\ \mathrm{E}[R^2_{21;2}] \\ \mathrm{E}[R^2_{22';2}] \\ \mathrm{E}[R^2_{22;2}] \\ \mathrm{E}[T_2^2] \end{pmatrix} = \begin{pmatrix} 0 \\ 1830.9 \\ 0 \\ 602.8 \\ 0 \\ 0 \\ 1282.5 \\ 0 \\ 0 \\ 9159.8 \end{pmatrix}$$

B.3. Example 3 calculations

The matrices in Fig. A7 are the result of the CE method extensions discussed in example 3. Since stages 1 and 2 do not generate any feedback, their expected durations are the given activity durations 81 and 56 respectively with a variance of 0. Stage 3 requires the CE method's equations for its calculations. The equations are developed and represented by the matrices below (populated by the algorithms in Figs. A1 and A2). Our interest is to calculate the expected duration of stage 3 ($E[T_3]$) and it is found, by solving the set of equations represented in matrix format below, to be $E[T_3] = 143.63$. And thus the expected duration of the network is obtained to be 280.63 which is the sum of all stage expected durations, where the

P	1	2	3
1			0.65
2	0.57		
3		1	

W	1	2	3
1			0.42
2	0.58		
3		0.44	

Durations
81
56
71

Fig. A7. Example 3 input matrices.

other durations are noted earlier (81 and 56 for stages 1 and 2 respectively).

$$\begin{pmatrix} 1 & -0.57 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -0.65 & 0 & 0 & 0 \\ 0 & -0.57 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.65 & 0 & 0 & 0 \\ 0 & -0.57 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.65 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.65 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{E}[R_{11;3}] \\ \mathrm{E}[R_{13;3}] \\ \mathrm{E}[R_{13;3}] \\ \mathrm{E}[R_{23;3}] \\ \mathrm{E}[R_{23;3}] \\ \mathrm{E}[R_{23;3}] \\ \mathrm{E}[R_{31;3}] \\ \mathrm{E}[R_{32;3}] \\ \mathrm{E}[R_{33;3}] \\ \mathrm{E}[T_3] \end{pmatrix} = \begin{pmatrix} 0 \\ 32.48 \\ 0 \\ 0 \\ 0 \\ 31.24 \\ 34.02 \\ 0 \\ 0 \\ 71 \end{pmatrix}$$

To find the variance, we start by calculating the second moment using the CE equations. The below matrices (populated by the algorithms in Figs. A1 and A3) represent the required equations to find the second moment of stage 3 ($\mathrm{E}[T_3^2]$). Solving the equations results in $\mathrm{E}[T_3^2] = 29847$, and thus $\mathrm{Var}[T_3] = \mathrm{E}[T_3^2] - \mathrm{E}^2[T_3] = 9217.6$ and this is the variance of the whole network as the other variances are equal to zero. Thus the standard deviation is 96.

$$\begin{pmatrix} 1 & -0.57 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -0.65 & 0 & 0 & 0 \\ 0 & -0.57 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.65 & 0 & 0 & 0 \\ 0 & -0.57 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.65 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.65 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{E}[R_{11;3}^2] \\ \mathrm{E}[R_{12;3}^2] \\ \mathrm{E}[R_{13;3}^2] \\ \mathrm{E}[R_{21;3}^2] \\ \mathrm{E}[R_{22;3}^2] \\ \mathrm{E}[R_{23;3}^2] \\ \mathrm{E}[R_{31;3}^2] \\ \mathrm{E}[R_{32;3}^2] \\ \mathrm{E}[R_{33;3}^2] \\ \mathrm{E}[T_3^2] \end{pmatrix} = \begin{pmatrix} 0 \\ 7802 \\ 0 \\ 0 \\ 0 \\ 5514 \\ 6445 \\ 0 \\ 0 \\ 15355 \end{pmatrix}$$

B.4. Example 4 calculations

The above matrices are the result of the CE method extensions discussed in example 4. Six activities exist where two of them are dummy and thus their stage calculations are neglected. They are included in the matrices to be considered in the rework calculations of other stages. Activities 1 and 2 do not generate any feedback and thus their stage durations are simply the activity durations 60 and 53 respectively with variance 0. However, stages 3 and 4 require the CE method equations to find the expected durations and variances. The required equations are generated in similarly to the previous examples, however since the input DSMs for stages 3 and 4 contain 5 and 6 activities, they require 26 and 37 equations respectively, and thus they are not presented. Stage 4 takes the input matrices shown in Fig. A8. For stage 3, since it is not the final stage, the input matrices are adjusted by deleting the proceeding activities and presented in Fig. A9. The results of these stage calculations are presented in Table A1 giving the network an expected duration and variance of 358.49 and 31829.1 (standard deviation of 178.41) when summed with the stages 1 and 2.

P	1	1'	1''	2	3	4
1						
1'	1				0.45	
1''						0.4
2		0.8	0.65			
3				1		
4					0.55	

W	1	1'	1''	2	3	4
1						
1′	1				0.9	
1′′						0.8
2		0.7	0.5			
3				0.4		
4					0.6	

Durations
60
60
60
53
66
33

Fig. A8. Example 4 input matrices.

P	1	1'	1''	2	3
1					
1'	1				0.45
1''					
2		0.8	0.65		
3				1	

W	1	1'	1''	2	3
1					
1′	1				0.9
1′′					
2		0.7	0.5		
3				0.4	

Durations
60
60
60
53
66

Fig. A9. Stage 3 input matrices (Example 4).

Table A1
Stage calculation results (Example 4).

Stage	Expected duration	Second moment	Variance
3	139.687	31,999	12486.54
4	105.8	30,537	19343.36

Appendix C

Section 4 Softex Case Study's Input Matrices (see Figs. A10 and A11).

	1	2	3	4	5	6	7	8	9	10	11	12	11'	12'	13'	13	13''	14	16	17	15'	18'	18	15	19
1																									
2	1																								
3	1																								
4	1																								
5				1																					
6					1																				
7		1			1																				
8		1	1		1																				
9								1																	
10					1				1																
11										1					0.25	0.75	0.25								
12										1					0.11	0.06	0.11								
11'																					0.15	0	0.05	0.25	
12'																					0			0.08	
13'											1	1													
13																									
13''													1	1											
14															0	0	0								
16															0.64	0.19	0.64								
17																			1						
15'																		1							
18'																				1					
18																									
15																									
19																					1	1	1	1	

Fig. A10. Rework probabilities DSM.

	1	2	3	4	5	6	7	8	9	10	11	12	11'	12'	13'	13	13''	14	16	17	15'	18'	18	15	19
1																									
2	1																								
3	1																								
4	1																								
5				1																					
6					1																				
7		1			1																				
8		1	1		1																				
9								1																	
10					1				1																
11										1					0.5	0.5	0.5								
12										1					0.4	0.4	0.4								
11'																					0.25	0.1	0.1	0.25	
12'																					0.15			0.15	
13'											0.25	0.25													
13																									
13''													0.1	0.1											
14															1	1	0.25								
16															1	1	1								
17																		1	1						
15'																		1							
18'																				1					
18'																									
15																									
19																					1	1	1	1	

Fig. A11. Rework proportions DSM.

References

Abdelsalam, H., & Bao, H. (2006). A simulation-based optimization framework for product development cycle time reduction. *IEEE Transactions on Engineering Management, 53*(1), 69–85.

Browning, T. R., & Eppinger, S. D. (2002). Modeling impacts of process architecture on cost and schedule risk in product development. *IEEE Transactions on Engineering Management, 49*(4), 428–442.

Browning, T. R., & Ramasesh, R. V. (2007). A survey of activity network-based process models for managing product development projects. *Production and Operations Management Society, 16*(2), 217–240.

Browning, T. R., & Yassine, A. (2016). Managing a portfolio of product development projects under resource constraints. *Decision Sciences, 47*(2), 333–372.

Casella, G., & Berger, R. L. (2002). *Statistical inference* (2nd ed). Australia: Thomson Learning.

Cho, S., & Eppinger, S. (2005). A simulation-based process model for managing complex design projects. *IEEE Transactions on Engineering Management, 52*(3), 316–328.

Eppinger, S., Nukala, M., & Whitney, D. (1997). Generalized models of design iteration using signal flow graphs. *Research in Engineering Design, 9*(2), 112–123.

Hu, J., Liu, J., & Prasad, B. (2003). A constraint-driven execution plan for maximizing concurrency in product development. *Concurrent Engineering Research and Applications, 11*(4), 301–312.

Jun, H. B., Park, J. Y., & Suh, H. W. (2006). Lead time estimation method for complex product development process. *Concurrent Engineering Research and Applications, 14*(4), 313–328.

Liu, Y., Hisarciklilar, O., Thomson, V., & Bhuiyan, N. (2015). A study of overlapping and functional interaction mechanisms for concurrent engineering processes. *Concurrent Engineering Research and Applications, 23*(2), 83–97.

Mantel, S., Meredith, J., Shafer, S., & Sutton, M. (2007). *Project management in practice.* John Wiley & Sons.

Meier, C., Browning, T., Yassine, A., & 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.

Meier, C., Yassine, A. A., & Browning, T. R. (2007). Design process sequencing with competent genetic algorithms. *ASME Journal of Mechanical Design, 129*, 566–585.

Nasr, W., Yassine, A., & Abou Kasm, O. (2016). An analytical approach to estimate the expected duration and variance for iterative product development projects. *Research in Engineering Design, 27*(1), 55–71.

Pinkett, R. (1998). Product development process modeling and analysis of digital wireless telephones, *Master Thesis, MIT*.

Pinto, J. K. (2012). *Project management: Achieving competitive advantage* (3rd ed.). Prentice Hall.

Prasad, B. (1999). System integration techniques of sharing and collaboration among work-groups, computers, and processes. *Journal of Systems Integration, 9*, 115–139.

Smith, R., & Eppinger, S. (1997). A predictive model of sequential iteration in engineering design. *Management Science, 43*(8), 1104–1120.

Unger, D. W., & Eppinger, S. D. (2009). Comparing product development processes and managing risk. *International Journal of Product Development, 8*(4), 382–402.

Yang, Q., Zhang, X., & Yao, T. (2012). An overlapping-based process model for managing schedule and cost risk in product development. *Concurrent Engineering Research and Applications, 20*(1), 3–17.

Yassine, A., & Braha, D. (2003). Complex concurrent engineering and the design structure matrix method. *Concurrent Engineering Research and Applications, 11*(3), 165–176.