INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

THE ESTIMATION OF PRODUCT-DEVELOPMENT PROJECT DELAY CAUSED BY IMPERFECT COMMUNICATION IN OUTSOURCING

Gyesik OH, Yoo S. HONG
Seoul National University, Korea

ABSTRACT

As competition in the market becomes fierce, companies are required to develop a new product within short duration. One of methods to accelerate the product development is concurrent engineering which allows overlapping interdependent development activities. However, the constraint of development capacity limits the number of development activities concurrently executed. In order to overcome the development resource limitation, companies outsource some of development activities. Due to culture and language differences and low media richness of communication channels, communication between the company and the outsourcing provider takes more time than that between in-house development teams. The different combinations of outsourcing activities result in different project duration since overall communication delay varies with the choice of outsourcing activities under complex product development architecture. We estimate the project duration in consideration of communication delay on overall project duration under concurrent engineering by simulating the product development process. The paper may help product development managers to estimate the project prolongation in outsourcing.

Keywords: product development outsourcing, communication delay, product development project duration, product development architecture, concurrent engineering

Contact:
Gyesik Oh
Seoul National University
Department of Industrial Engineering
Seoul
151-744
Korea, Republic of (South Korea)
gushigi4@snu.ac.kr

1 INTRODUCTION

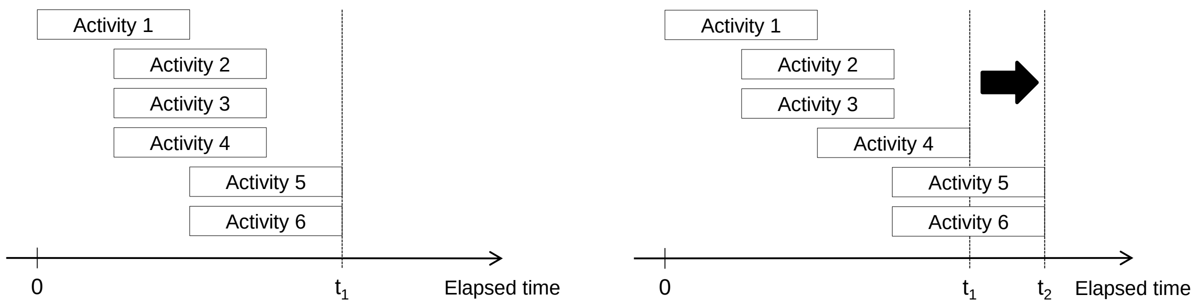
In order to sustain competency and share large portion in the market, companies attempt to reduce time to market. Since the range of the opportunity window is getting smaller, companies need to enter the market earlier than rivals and preoccupy it. Therefore, they adopt several techniques to shorten product development durations. For example, they make cross-functional development groups involved in early stages (Karagozoglu and Brown, 1993), let suppliers participate in product development (Dröge et al., 2000) and develop some activities concurrently (Krishnan, 1996). One of the aforementioned techniques to reduce time to market is concurrent engineering on product development project.

Concurrent engineering is a product development management method which enables firms to accelerate product development process. Under the sequential product development process, successive activities begin after the end of precedent activities. The precedent activity refers to the upstream activity which has influence on the downstream activity, the successive activity. Under concurrent engineering, the overlapping is allowed between the precedent activity and the successive activity. Though the successive activity needs to execute rework due to overlapping, overall process finishes earlier than the sequential process in general.

The company cannot fully take advantage of concurrent engineering because of development capacity limitation. The limitation of resources (e.g., the number of developers) in the organization makes the company impossible to execute many design activities simultaneously (Belhe and Kusiak, 1997). Therefore, project managers schedule the development activities under the consideration of the possible number of activities concurrently executed (Joglekar and Ford, 2005). Figure 1 represents the project prolongation because of the limitation of development capacity. If the company has three development design teams and is able to simultaneously execute three activities at most, activity 4 is not able to be executed concurrently until activity 1 is finished. As a result, process duration prolongs from t_1 to t_2.

In order to overcome the shortage of development capacity under concurrent engineering, companies exploit product development resources of outsourcing providers. It is difficult to expand development capacity in house shortly since a time lag exists between making a decision and realizing the capacity expansion (Parker and Anderson Jr, 2002). Also, companies are not able to acquire main development resources such as skilled technical employees and programmers in short time (Anderson Jr and Joglekar, 2005). Therefore, companies outsource some of product development activities (Swink, 2002), found a development center offshore (Eppinger and Chitkara, 2006), or establish a consortium (Abdalla, 1999). Previous researches indicate that one of the major motivations of product development activity outsourcing is reducing time to market by exploiting the development resources of outsourcing providers (Takeishi, 2001; Anderson Jr and Joglekar, 2005; Narasimhan et al., 2010)

Different from the expectation of project managers who outsource some of development activities in order to shorten project duration, imperfect information exchange between the company and outsourcing providers results in project duration prolongation. Under product development outsourcing, the sender cannot transfer information as much as he/she intends since the information conveyance capacity of communication channel such as video conference, e-mail, and telephone is less than face-to-face meeting. Also, the receiver cannot fully understand the information from the sender because of culture and language differences. We call the phenomenon of imperfect information transfer in product development outsourcing as *communication delay* which leads to project prolongation. In industry, many



(a) Development capacity: Unlimited **(b) Development capacity: 3 activities**

Figure 1. The effect of limited development capacity on concurrent engineering

	i	*j*
i		x
j	x	

Figure 2. Design structure matrix of two interdependent activities

researchers and managers found that global product development including offshoring and outsourcing requires more time to finish the project compared to in-house development(Herbsleb and Mockus, 2003; O'Sullivan, 2003; Gokpinar et al., 2011) .
In this paper, we estimate the delay of product development project with outsourcing by analyzing the structure of product development process. Although many researchers acknowledge the project delay because of communication problem, there is no research to estimate the amount of delay of overall product development. Rework which requires additional development resource exacerbates the problem of capacity limitation. Since product development process is so complex and rework probability has stochastic characteristic, it is difficult to estimate it with simple numeration. Therefore, we provide the simulation method for assessing the project duration under product development outsourcing environments. Our research would help product development project managers to consider both expected effect from exploiting development resources and side effect due to communication loss with numerical assessment.

2 TWO KINDS OF REWORK IN CONCURRENT ENGINEERING

Rework in complex product development process delays overall project duration since it requires development resource and prevents other activities from executing. Rework refers to the design work which modifies its prior work executed under imperfect information of other activities which has influence on it. It is different from nominal work which is performed under the assumption of perfect information for other activities. Figure 2 represents the dependency between two design activities with DSM (design structure matrix). A mark 'x' in cell (*i*, *j*) indicates that activity *i* depends on the design information of activity *j*. We call activity *j* and activity *i* as an affecting activity and an affected activity respectively in the dependent relationship. Since activity *j* relies on activity *i*, activity *i* is also the affecting activity for activity *j* in cell (*j*, *i*).
Rework is categorized into continuous rework and discrete rework as whether the affecting activity has the information of affected activity or not. Figure 3 represents two kinds of rework between coupled activities *i* and *j*. Continuous rework refers to the additional work right after the nominal work under overlapping. Under overlapping environment, the precedent activity *i* performs its work based on the assumption or guess of successive activities which is not executed (Steward, 1981). The precedent activity *i* is not able to set its design variables which are compatible with the variables of the successive activity *j*. Different from the situation of continuous rework, affecting activity *j* has the information of affected activity *i* since affecting activity *j* begins after the end of affected activity *i*. The affecting activity *j* is able to establish its design in consideration of minimizing the change of affected activity *i*. Therefore, affected activity does not necessarily perform discrete rework but continuous rework.

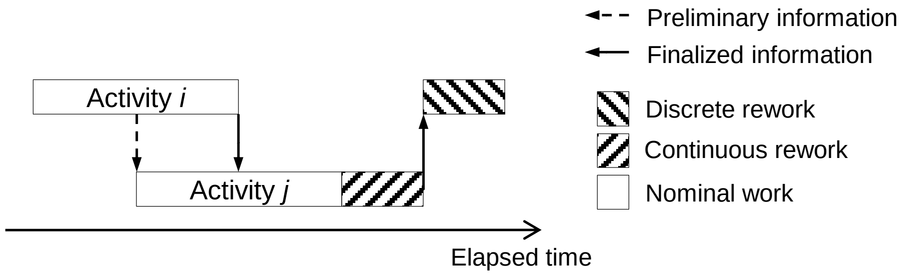


Figure 3. Two kinds of rework in concurrent engineering

The amount of continuous rework is determined by the uncertainty of preliminary information from the precedence activity and the degree of execution of the successive activity at the time of receiving finalized information. The earlier the information is sent from the precedent activity, the more amount of rework the successive activity should executes. Also, the later the successive activity receives information, the more amount of rework the successive activity should executes in order to revise what it has done. Therefore, the amount of continuous rework depends on the progress of both the precedent and the successive activity (Krishnan et al., 1997; Loch and Terwiesch, 1998). The amount of continuous rework, $a(j_{cr})$, is calculated as in Equation (1) in proportion of the degree of progress of the precedent activity i, $p(i,p)$, and that of the successive activity j, $p(j,s)$ to the amount of nominal work, $a(j_n)$.

$$a(j_{cr}) = (1 - p(i,p)) \times p(j,s) \times a(j_n) \tag{1}$$

Depending on the characteristics of the activity, the progress rate of design activity varies (Krishnan et al., 1997). The function of degree of progress can be concave, convex or linear. However, we assume that the rate for the degree of progress is linear to design time in order to simplify the calculation as in Figure 4. The degree of progress at the moment of preliminary information transfer, p_p, is calculated as the portion of design execution duration which is equal to preliminary information transfer time minus activity start time, $t(i,p)$-$t(i,s)$, to the work duration for activity i, $d(i_n)$. In the same manner, the degree of progress for successive activity is represented as the portion of design execution duration before the reception time of finalized information, $t(i,f))$-$t(j,s)$. Under the assumption, the amount of continuous rework is represented as in equation (2).

$$a(j_{cr}) = (1 - \frac{t(i,p) - t(i,s)}{d(i_n)}) \times \frac{t(j,f) - t(j,s)}{d(j_n)} \times a(j_n) \tag{2}$$

The characteristics of discrete rework are represented by two indices, rework impact and rework probability. Rework impact refers to the portion of the amount of rework to that of nominal work. If activity j causes discrete rework to activity i, the amount of rework for activity i is represented as in Equation (3).

$$a(i_{dr}) = RI(i,j) \times a(i_n) \tag{3}$$

The probability of rework occurrence is rework probability. Since the affecting activity modifies its design in consideration of the design information of affected activities, rework probability are less than 1.

Since our goal is to examine the project duration, we need to convert the amount of work into work duration. Work duration for activity i, $d(i)$, is represented as the amount of work, $a(i)$, divided by work rate, w, as in Equation (4).

$$d(i) = \frac{a(i)}{w} \tag{4}$$

For example, if the amount of work for activity i is 150 man-days and the development team k has 5

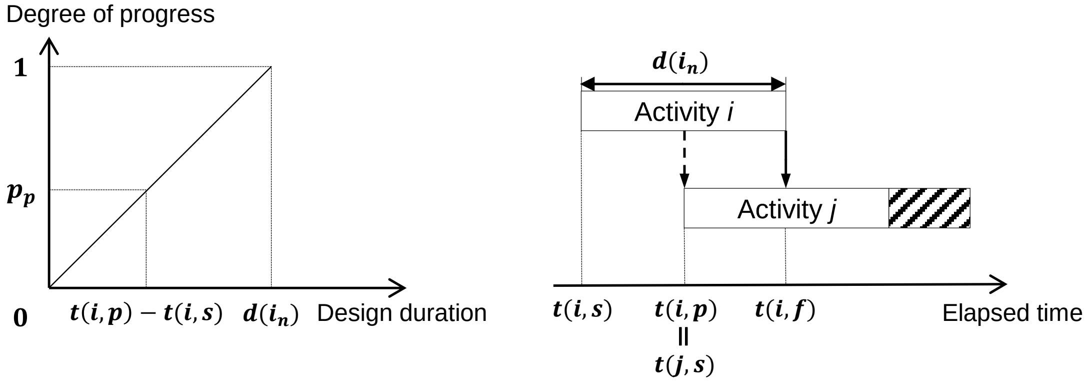


Figure 4. Degree of progress under the assumption of linear development rate function

members, its work duration is 30 days

We consider learning effect in rework rate to reflect the repetitive characteristics of rework. When engineers conduct the same work again, rework rate is higher than nominal work rate. This phenomenon is called learning effect (Ahmadi et al., 2001). Rework rate increases in portion of learning effect index, l, as in Equation (5). i_r refers to the rework for activity i. It is either continuous rework, i_{cr}, or discrete one, i_{dr}.

$$d(i_r) = \frac{a(i_r)}{w(1+l)}, \quad i_r \in \{i_{cr}, i_{dr}\} \tag{5}$$

For instance, when development team k executes activity i again and its learning effect is 50%, its rework duration is 20 days. Therefore, rework duration is shorter than nominal work duration though they execute the same design activity.

3 WORK POLICY OF CONCURRENT ENGINEERING

Product development project managers have to establish work policy in consideration of minimizing project duration. Work policy of concurrent engineering determines when the successive activity begins and whether the successive activity accepts intermediate information from the precedent activity or not. Since work policy affects the overall project duration, it is important to establish proper work policy (Terwiesch et al., 2002; Roemer and Ahmadi, 2004).

We establish that intermediate information between preliminary information and finalized information is disregarded by the successive activity. After preliminary information is sent, the design form of the precedent activity is changing continuously as the design of the precedent activity evolves. Since it takes costs to accept information and coupled activities might lead too many iterations, it is advised for the successive activity to dismiss intermediate information under complex product development project (Terwiesche et al., 2002; Lin et al., 2012). Therefore, the successive activity accepts information from only two time points, critical time and the end of the precedent activity.

The successive activity is able to begin its work when it receives the information above a certain level of certainty from the precedence activity. Since the successive activity relies on the design information of the precedent activity, it needs basic information based on which it executes its design work. Product development activity is defined as to resolve the uncertainty of design form (Loch and Terwiesch, 1998). As time goes on, the precedent activity develops its design and solves the uncertainty. We define *critical time* as the time at which the successive activity is able to start its work. Each pair of the precedent activity and the successive activity has their own critical time as the ratio of precedent activity duration (Yang et al., 2012). We define *CTI* (critical time index) as the proportion of the precedent activity execution duration based on which the successive activity is able to begin to the precedent activity duration. Therefore, critical time between activity i and j, $t(i,c_j)$, is calculated as in Equation (6). The successive activity begins at the critical time as in Figure 5.

$$t(i,c_j) = t(i,s) + d(i_n) \times CTI(i,j) \tag{6}$$

The short successive activity needs to begin its work later than critical time. When the successive activity duration, $d(j_n)$, is less than the precedent activity duration after critical time as in Condition 7, it

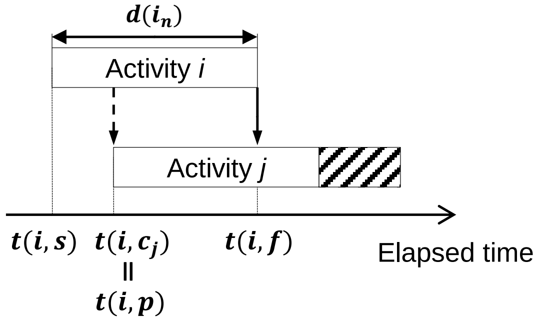


Figure 5. Work policy of concurrent engineering

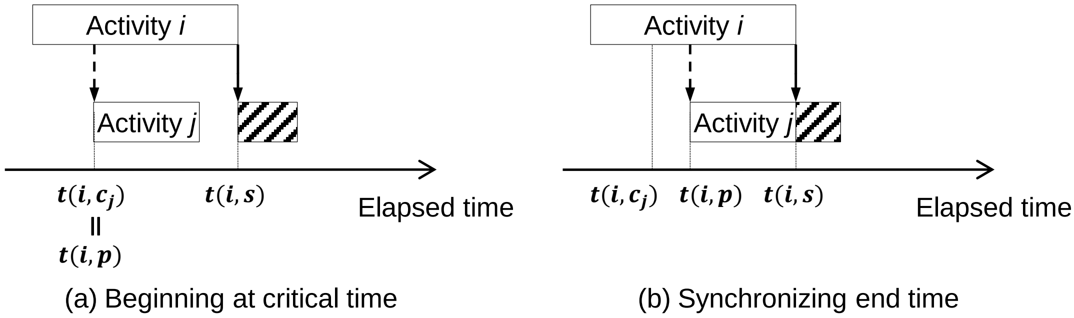


Figure 6. Work policy of concurrent engineering for the short successive activity

has to wait for the finalized information at the end of the precedent activity as in Figure 6(a).

$$d(j_n) < d(i_n) \times (1 - CTI(i, j)) \tag{7}$$

If the successive activity synchronizes its end with If the successive activity synchronizes its end with the precedent activity, it begins its work with more certain information from the precedent activity. Since more certain information causes the less amount of rework, it is preferred to receive information later if the beginning of continuous is not changed. Though continuous rework of activity *j* starts at *t(i,s)* both in Figure 6(a) and 6(b), continuous rework duration is shorter in Figure 6(b).

4 COMMUNICATION DELAY IN CONCURRENT ENGINEERING

Under product development outsourcing, information is not transferred as effectively as in-house development. Effective information transfer is required between design activities in order to enhance the performance of project (Ulrich and Eppinger, 2003). In traditional product development environments, face-to-face meeting is the major communication channel since companies perform all development activities in house. Since outsourcing providers are distributed all over the world, companies which consider outsourcing are looking for other communication channels to solve the problem.

Communication channels which companies adopt to overcome geographical distance take longer time to transfer information than face-to-face meeting. Daft (2009) defines media richness as information carrying capacity and argues that each medium has different media richness. Since the communication channel of high media richness permits multiple communication cues (e.g., tone of voices, facial expression, body language) and allows rapid mutual feedback, it takes shorter time to communicate (Suh, 1999). In respect of media richness, communication channels such as video conference, fax, and email have lower media richness than face-to-face meeting. Therefore, it takes more time to transfer information by using communication channels under outsourcing environments. In fact, Warkentin et al. (1997) find that a video conference team takes longer time to execute the project and underperforms than a face-to-face team in their empirical study.

Culture and language differences and different time zone also delay information transfer between the company and outsourcing providers. In the communication between engineers who have different culture background and use different languages, messages are likely to be understood in various ways (Farmer and Hyatt, 1994). In order to prevent misunderstanding, the receiver needs more detailed information and frequent mutual feedback. Since communication channel with low media richness hinders rapid mutual feedback, it prolongs the duration to receive information. Different time zone exacerbates the communication problem. For example, it requires at least one business day to receive feedback between the company in US and the outsourcing provider in Taiwan since the company is located 14 time zones away from the outsourcing provider (Parker and Anderson Jr, 2002).

In the industrial field, it is possible to assess the effect of communication delay when information is transferred in outsourcing environments. Gokpinar et al. (2011) propose the large archival database of engineering change orders (ECOs) as the source from which managers are able to gauge communication delay effect. ECOs are the record of formal communication and collaboration, which contains the information of location, issue data and other related information. Since managers are able to compare the amount of design work executed in house with that executed with outsourcing providers, they

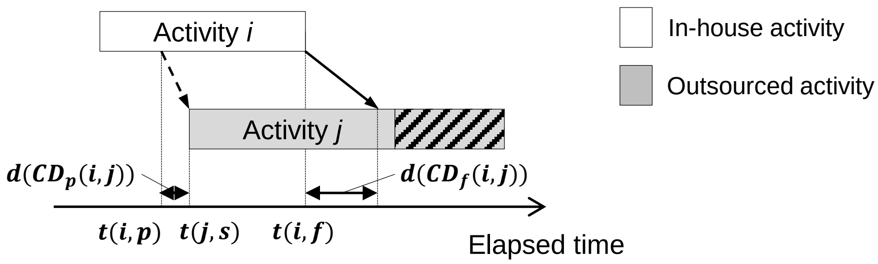


Figure 7. Communication delay in product development outsourcing

gauge the delay due to outsourcing. If the large archival database is absent, managers would rely on survey to engineering designers (Suh, 1999; Gupta and Govindarajan, 2000). From aforementioned methods in the field, the effect of communication delay can be measured.

We define *CDI* (communication delay index) as the ratio of the delay caused by communication problem to the execution duration to be transferred. The communication delay duration for preliminary information transfer between precedent activity i and successive activity j, $d(CD_p(i,j))$, is calculated as the ratio of execution duration until information sending, $t(i,p)-t(i,s)$, in Equation (8). In the same manner, the communication delay duration for finalized information is calculated as in Equation (9).

$$d(CD_p(i,j)) = (t(i,p)-t(i,s)) \times CDI(i,j) \tag{8}$$

$$d(CD_f(i,j)) = (t(i,f)-t(i,p)) \times CDI(i,j) \tag{9}$$

Under concurrent engineering, communication delay affects the product development process as in Figure 7. The time at which the successive activity starts its work, $t(j,s)$, is later than when preliminary information is sent, $t(i,p)$, as preliminary information transfer delay duration, $d(CD_p(i,j))$. Also finalized information from the precedent activity is transferred later as the amount of finalized information transfer delay, $d(CD_f(i,j))$. Since the successive activity receives the finalized information later, the amount of rework is more than in-house development.

5 NUMERICAL EXAMPLE

To illustrate the implementation of project duration with outsourcing estimation model, we present numerical example based on the data of the case study by Browning and Eppinger (2002). In their study, the focal company is in charge of uninhabited combat aerial vehicle (UCAV) preliminary design project. It consists of 14 activities. Off-diagonal cells in Figure 8 represent rework impact and rework probability respectively and diagonal cells represent the amount of nominal design work.

We consider the focal company which has two design teams and are able to execute at most two design activities concurrently. In order to shorten project duration, the company attempts to outsource some of design activities to the outsourcing provider which has one design team. We assumed some of basic parameters before modeling the project execution. We set *CDI* (communication delay index) as 0.1, which means that the receiver delays its beginning at the amount of 10% of precedent activity duration when the sender and the receiver belongs to different companies. We also assume that rework

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2													
2	.5	5							.1					
3		.3	2.8	.5										
4	.4		.8	10										
5	.1		.1		15	.1		.1				.3	.1	
6	.1				.3	10								
7	.5						8							
8								5				.5		
9	.3		.3	.3				.3	20					
10				.1		.5	.4	.3		10	.3			
11						.5	.5	.3		.3	15			
12	.5					.3	.5			.5	.5	15		
13	.9				.9							.3	32	
14	.5	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	5

(a) Rework impact

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2													
2	.4	5							.2					
3		.5	2.8	.4										
4	.3		.5	10										
5	.4		.5		15	.1		.1				.3	.1	
6	.1				.4	10								
7	.4						8							
8								5				.5		
9	.4		.5	.5				.5	20					
10				.1		.5	.2	.1		10	.4			
11						.5	.5	.5		.5	15			
12	.4					.4	.5			.5	.4	15		
13	.5				.5							.4	32	
14	.3	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	5

(b) Rework probability

Figure 8. The rework indices of UCAV preliminary design project (Browning and Eppinger, 2002)

Table 1. Simulation results of product development project duration

(Unit: days)

Development strategy	The number of teams	Outsourced activities	Mean	Standard Deviation
In-house development	2 in-house teams	None	103.88	4.966
	3 in-house teams		93.70	7.170
Outsourcing development	2 in-house teams and 1 outsourcing team	4, 8, 11	98.19	7.215
		4, 6, 13	109.92	5.984
		3, 8, 12	103.63	5.304

rate equals to 1.3. The design team finishes design work 30% faster than when it conducts the same activity again. We set critical point, *Cr*, as 0.3. A successive activity is able to begin its design work when a precedent activity executes 30% of work amount.

We check the effect of outsourcing with different outsourcing activity combinations. Before estimating the project duration of outsourcing, we estimate in-house development project duration. Under current situation with 2 in-house development teams, expected project duration is 104 days as in Figure 9. Since there is no communication delay under in-house development environments, the project duration executed by 3 in-house development teams can be regarded as the ideal project duration for outsourcing. Therefore, managers who outsource some of design activities expect to minimize project duration to 94 days.

Depending on the combination of outsourcing activities, the project duration under product development outsourcing varies as in Table 1. By comparing Figure 10 and 11, the reason of project duration difference might be figured out. In Figure 11, activities from 9 to 11 are developed in house. Because of limitation of development resources, activity 11 in Figure 11 begins later than that in Figure 10. In Figure 11, project duration is longer than that of current development situation with two development teams since the outsourcing company is not able to resolve the development capacity problem. Also communication delay exacerbates the development process delay. In Figure 11, activity 14 starts its work later than that in Figure 10 since it receives information after communication delay. Therefore, careful choice of outsourcing activities is important to achieve the goal of decreasing project duration.

6 CONCLUSION

Companies adopt concurrent engineering method to shorten product development project duration. Since overlapping between a precedence activity and a successive activity is allowed under concurrent engineering, product development project is able to be finished earlier than sequential development process. However, the limitation of development resources prevents companies from fully taking advantages of concurrent engineering. Companies attempt to utilize the development resources from outside by outsourcing some of development activities. Different from the expectation of companies, product development duration prolongs under outsourcing environments. The low media richness of communication channels and culture and language differences prohibit perfect information transfer between the company and outsourcing providers. We define the phenomenon of information transfer delay as communication delay.

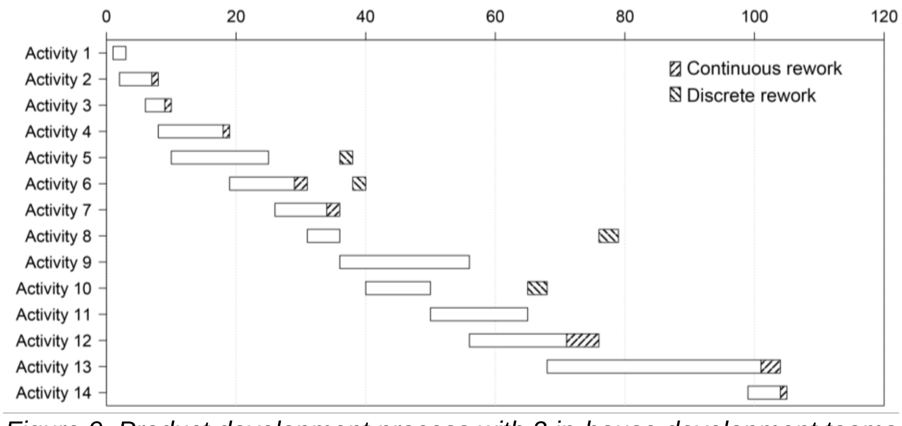


Figure 9. Product development process with 2 in-house development teams

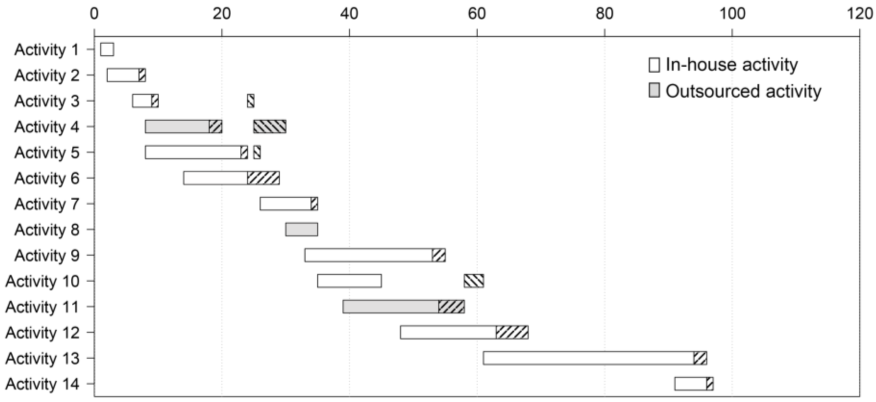


Figure 10. Product development process when activity 4, 8 and 11 are outsourced

We developed the project duration estimation method under outsourcing environments. It includes communication delay between a company and an outsourcing provider, the work policy for concurrent engineering researched by many scholars and effects of reworks initiated by feedback. Since rework initiated by feedback has the stochastic characteristics with rework probability, we simulated the product development process. With the numerical example, we found that the effect of project duration reduction varies depending on the selection of outsourcing activities. The estimation model would help project managers to choose outsourcing development activities in consideration of project prolongation caused by communication delay.

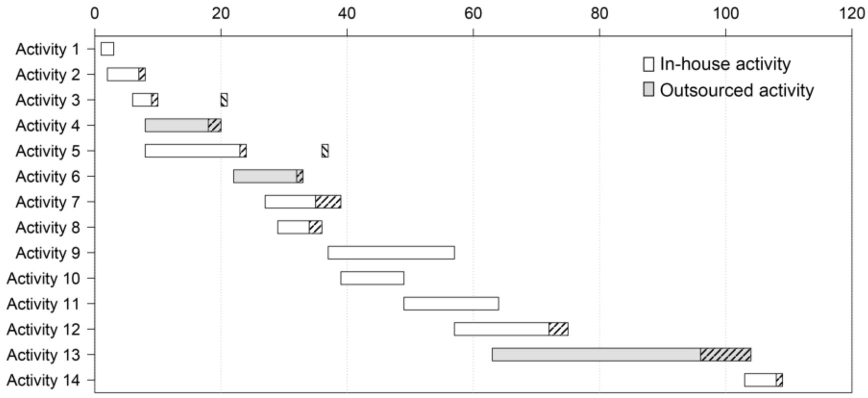


Figure 11. Product development process when activity 4, 6, and 13 are outsourced

REFERENCES

Abdalla, H. S. (1999) 'Concurrent engineering for global manufacturing', *International Journal of Production Economics*, vol.60-61, no.1, pp.251-260.

Ahmadi, R., Roemer, T. A. and Wang, R. H. (2001) 'Structuring product development processes', *European Journal of Operational Research*, vol.130, no.3, pp.539-558.

Anderson Jr, E. G. and Joglekar, N. R. (2005) 'A hierarchical product development planning framework', *Production and Operations Management*, vol.14, no.3, pp.344-361.

Belhe, U. and Kusiak, A. (1997) 'Dynamic scheduling of design activities with resource constraints', *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol.27, no.1, pp.105-111.

Browning, T. R. and Eppinger, S. D. (2002) 'Modeling impacts of process architecture on cost and schedule risk in product development', *IEEE Transactions on Engineering Management*, vol.49, no.4, pp.428-442.

Daft, R. L. (2009) *Organization Theory and Design*, Connecticut, South-Western Cengage Learning.

Dröge, C., Jayaram, J. and Vickery, S. K. (2000) 'The ability to minimize the timing of new product development and introduction: an examination of antecedent factors in the North American automobile supplier industry', *Journal of Product Innovation Management*, vol.17, no.1, pp.24-40.

Eppinger, S. D. and Chitkara, A. R. (2006) 'The new practice of global product development', *MIT Sloan Management Review*, vol.47, no.4, pp.22-30.

Farmer, S. M. and Hyatt, C. W. (1994) 'Effects of task language demands and task complexity on computer-mediated work groups', *Small Group Research*, vol.25, no.3, pp.331-366.
Gokpinar, B., Hopp, W. J. and Iravani, S. M. R. (2011) 'In-house globalization: The role of globally distributed design and product architecture on product development performance (Working Paper No.1163)'. Retrieved from University of Michigan website: http://deepblue.lib.umich.edu/bitstream/2027.42/85793/1/1163_Hopp.pdf.
Gupta, A. K. and Govindarajan, V. (2000) 'Knowledge flows within multinational corporations', *Strategic Management Journal*, vol.21, no.4, pp.473-496.
Herbsleb, J. D. and Mockus, A. (2003) 'An empirical study of speed and communication in globally distributed software development', *Software Engineering, IEEE Transactions on*, vol.29, no.6, pp.481-494.
Joglekar, N. R. and Ford, D. N. (2005) 'Product development resource allocation with foresight', *European Journal of Operational Research*, vol.160, no.1, pp.72-87.
Karagozoglu, N. and Brown, W. B. (1993) 'Time-based management of the new product development process', *Journal of Product Innovation Management*, vol.10, no.3, pp.204-215.
Krishnan, V. (1996) 'Managing the simultaneous execution of coupled phases in concurrent product development', *Engineering Management, IEEE Transactions on*, vol.43, no.2, pp.210-217.
Krishnan, V., Eppinger, S. D. and Whitney, D. E. (1997) 'A model-based framework to overlap product development activities', *Management science*, vol.43, no.4, pp.437-451.
Lin, J., Qian, Y. and Cui, W. (2012) 'Managing the Concurrent Execution of Dependent Product Development Stages', *Engineering Management, IEEE Transactions on*, vol.59, no.1, pp.104-114.
Loch, C. H. and Terwiesch, C. (1998) 'Communication and uncertainty in concurrent engineering', *Management science*, vol.44, no.8, pp.1032-1048.
Narasimhan, R., Narayanan, S. and Srinivasan, R. (2010) 'Explicating the mediating role of integrative supply management practices in strategic outsourcing: a case study analysis', *International Journal of Production Research*, vol.48, no.2, pp.379-404.
O'Sullivan, A. (2003) 'Dispersed collaboration in a multi-firm, multi-team product-development project', *Journal of Engineering and Technology Management*, vol.20, no.1, pp.93-116.
Parker, G. G. and Anderson Jr, E. G. (2002) 'From buyer to integrator: the transformation of the supply-chain manager in the vertically instegrating firm', *Production and Operations Management*, vol.11, no.1, pp.75-91.
Roemer, T. A. and Ahmadi, R. (2004) 'Concurrent crashing and overlapping in product development', *Operations Research*, vol.52, no.4, pp.606-622.
Steward, D. V. (1981) 'The design structure system: A method for managing the design of complex systems', *IEEE Transactions on Engineering Management*, vol.28, no.3, pp.71-74.
Suh, K. S. (1999) 'Impact of communication medium on task performance and satisfaction: an examination of media-richness theory', *Information & Management*, vol.35, no.5, pp.295-312.
Swink, M. (2002) 'Product development-faster, on-time', *Research-Technology Management*, vol.45, no.4, pp.50-58.
Takeishi, A. (2001) 'Bridging inter-and intra-firm boundaries: management of supplier involvement in automobile product development', *Strategic Management Journal*, vol.22, no.5, pp.403-433.
Terwiesch, C., Loch, C. H. and De Meyer, A. (2002) 'Exchanging preliminary information in concurrent engineering: Alternative coordination strategies', *Organization Science*, vol.13, no.4, pp.402-419.
Ulrich, K. T. and Eppinger, S. D. (2003) *Product design and development*, New York, McGraw-Hill.
Warkentin, M. E., Sayeed, L. and Hightower, R. (1997) 'Virtual Teams versus Face-to-Face Teams: An Exploratory Study of a Web-based Conference System', *Decision Sciences*, vol.28, no.4, pp.975-996.
Yang, Q., Zhang, X. and Yao, T. (2012) 'An overlapping-based process model for managing schedule and cost risk in product development', *Concurrent Engineering*, vol.20, no.1, pp.3-17.