Accelerating the Process of Engineering Change Orders: Capacity and Congestion Effects

Christoph H. Loch
INSEAD, Fontainebleau, France

Christian Terwiesch
The Wharton School, University of Pennsylvania, Philadelphia, USA

June 1998

Abstract

Engineering change orders (ECOs) are important drivers of development costs and lead time. This article analyzes the process of administering engineering change orders in the case of the climate control system development within a large vehicle development project. This administrative process encompasses the emergence of a change (e.g., a problem or a market-driven feature change), its management approval and final implementation. Despite strong time pressure, this process can take several weeks, several months, and, in extreme cases, even over a year. Such a long lead time is especially remarkable as the actual processing time for the change typically does not exceed two weeks. Based on our case study, we develop an analytical framework that explains how such an extreme ratio between theoretical processing time and actual lead time is possible. The framework identifies congestion, stemming from scarce capacity coupled with processing variability, as a major lead time contributor. We outline five improvement strategies which an organization can use in order to reduce ECO lead time, namely flexible capacity, balanced workloads, merged tasks, pooling, and reduced set-ups and batching.

Introduction

Engineering change orders (ECOs) – changes to parts, drawings or software that have already been released – are important drivers of development costs and lead time. Given this importance of ECOs, most large organizations use a formal support process administering these changes. This ECO process is present at the back end of almost all complex new product development (NPD) projects. It has been identified as one of the root causes of high ECO costs [5], which in many projects can account for one third to one half of engineering capacity [21] and 20-50% of tool costs [17].

In an in-depth field study of the climate control system (CCS) development in a new vehicle (reported in [22]), we have identified congestion, stemming from scarce engineering capacity, as one of the main drivers of long ECO lead times. Specifically, we have observed numerous cases where the overall lead time exceeded the pure problem-solving time by a factor 10 and more. This observation is consistent with Blackburn [4] who reports that the value-added time for ECOs in airframe manufacturing is as low as 8.5%. Thus, for each day of actual processing time, there are two weeks of non-value-added time. Most of this non-value-added time is waiting time. But how is it possible that a process that has a net task time (value-added time) of less than a week, takes a full month? What happens in the residual time? And finally, how can we improve the ratio between value-added time and non-value-added time?

The present article provides a detailed analysis of the ECO support process of complex NPD projects. The analysis is based on the theory of queueing and congestion applied to the ECO process. Queueing theory has found successful applications in manufacturing [15] and services [11]. We use queueing models to describe the detailed flows of documents and information in the ECO process. This approach allows us to explain the disproportionately long waiting times and to identify five improvement strategies, namely flexible capacity, balanced workloads, merged tasks, pooling, and reduced set-ups and batching, and apply them to the example of the ECO process.

Background

The effects of congestion on throughput times are well-known in manufacturing and service contexts (e.g. [11,15]), but have rarely been identified in the context of NPD in general and not at all in the ECO process. Among the few exceptions are Wheelwright and Clark [27], who observe that there are a number of manufacturing-like activities and even true manufacturing activities (such as prototype-building) within product development. This suggests that some process analysis approaches from manufacturing may be applied.

Blackburn [4,5] points concretely to the problem of long lead times in NPD processes. He observes that batching and delayed information transfer contribute to this problem. Adler *et al.* [1,2] go one step further and quantitatively demonstrate congestion effects

based on projects competing for engineering capacity. The authors use simulation analysis to understand the reasons for long lead times and recommend four improvement strategies: cross-training technicians to offload engineers (who were process bottlenecks), limiting the total number of projects under way at any point in time (that is, limit the backlog), avoiding expedited projects (projects that get high priority at the expense of others), and tracking project throughput times.

In Terwiesch and Loch 1998 [22], we present an in-depth case study of the CCS development in a vehicle. In this case study, we find that the time it took the organization to move an ECO from its creation to its successful implementation was surprisingly long. Despite the tremendous time pressure in development projects in general and in the ECO process in particular, process lead times were in the order of several weeks, several months, and, in extreme cases, even over a year. Looking at the causes for these long lead times in more detail, we were surprised by the low proportion of value-added time in the ECO support process. An ECO spent most of its lifetime "sitting on someone's desk", waiting for further processing. This observation is consistent with Blackburn [4] who reports similarly low proportion of value-added times for ECOs in airframe manufacturing.

The long lead times and the disproportionate amount of non-value-added time motivated us to take a more detailed look at the dynamics of the ECO process. In particular, we were interested in answering the following research questions:

- What causes the low ratio of value-added to total ECO lead time? What happens in the residual time?
- How can one improve the ratio between value-added time and non-value-added time?

The remainder of this article is organized according to the established logic of process analysis and redesign [12,18]. We first describe the existing ECO support process in the form of a process map. We then calculate capacity utilization profiles encountered at each step in the process, which allows the identification of process bottlenecks. Based on the utilization profiles, we outline our theoretical framework for understanding how high utilization coupled with process variability causes congestion, that is, competition for scarce resources and long non-value-added times. This leads to improvement strategies presented at the end of the article.

Mapping the Process

The first step in understanding why it takes so long from the detection of a problem to the implementation of the ECO consists of *mapping* the process (Figure 1). A process map reveals the structure of the process, but additional data on processing times and capacities must be collected to understand throughput time performance.

Structural Map

The process structure was drawn based on the descriptions of the individuals involved. A problem in a component, or in interactions between components, of the CCS is usually detected while testing prototypes. Prototypes may be virtual (existing in a CAD model), clay models, or physical models of varying completeness. When the problem is clearly identified and reproduced, and when a candidate solution strategy has been identified, an official ECO is created. At this point, the ECO approval process, depicted in Figure 1, begins (it is a subset of the activities considered in [21]). A detailed design is proposed by CCS engineers to resolve the problem. This solution must be simulated for effectiveness by the computer simulation group and then approved by the project manager (who also seeks input from the functional engineering departments), and by accounting, who examine the cost implications of the change. If approval is not granted, an alternative design must be developed.

Figure 1 here

Once implementation is authorized, the purchasing department asks the supplier to include the change in the next batch of prototype parts. When new parts with the ECO implemented arrive for prototype construction, an evaluation of the new design solution can be made. In some cases, the changed part proves ineffective (for example, when the design of the CCS system has changed during the time the ECO was resolved), in which case a new ECO must be generated.

Figure 1 contains two possible iteration loops. The first iteration occurs if approval is not granted, for example, when the change increases manufacturing cost unacceptably. The second loop occurs if, against expectation, the redesigned parts still contain problems. Both loops are incorporated in the frequency of ECOs arriving, i.e. some proportion of the arriving ECOs are re-issued ECOs.

Figure 1 shows only half of the actual group size for each engineering resource, as only about half of the available engineering capacity was consumed by the development project that we focused on. The other half of capacity was spent on other ongoing projects. In our host organization, most engineers (excluding the small project management organization) remained in their functional units and thus worked simultaneously on multiple projects. Several of them reported in interviews that this not only caused problems concerning their management of priorities, but also required them to frequently switch their attention from one project to another, causing a significant time-loss from “diving into the project again”.

The mere structure of the process already reveals that resolving an ECO consists of a long sequence of steps involving numerous people. In the terminology of Business Process Reengineering [13], the process contains several “bureaucratic” activities (such as accounting approval) and handoffs between groups. However, an elimination of accounting approval was not under discussion in the host organization, as they felt the cost control of changes to be of high importance.

Data Collection

Each ECO must be processed by several *resources*, comprising the engineers, project manager, accounting analyst, and purchaser. Each resource must perform *tasks*, which require a processing time. Each task takes between a few minutes (e.g., decision by project management) and a few hours (e.g., design proposal). Processing times are not tracked exactly in the organization. Therefore, we obtained *estimates of the averages* in our interviews with the engineers and the project manager. These estimated averages are shown in Figure 1 under the boxes corresponding to the engineering groups, and they are explained below.

An ECO takes an average of two hours to develop a solution proposal. Simulation requires a set-up of 30 minutes to perform data preparation of the files each time a new type of problem is tackled. Simulation itself takes 1.6 hours. The simulation group processes ECOs in batches of two (they pick out of their in-basket two problems, similar in structure), in order to economize on the set-ups (marked as *s = 0.5 hours* in Figure 1). Subsequently, an ECO needs, on average, 45 minutes to be checked for its cost impact, 45 minutes for new parts to be ordered, and ten minutes of the project manager's attention for approval. The project manager (who is extremely busy) discusses ECOs only once a week, during the weekly project team meeting, when approval or rejection decisions are made on the spot (each problem-solving engineer attends only to present his/her status, so the additional burden on them is small).

Two engineers devote their time to the first step; one person performs each of the other steps. All employees work five times eight hours (40 hours) a week. CCS and simulation engineers devote their entire time to this ECO process. The accounting and purchasing specialists have other responsibilities, but give priority to this process, so their effective utilizations and throughput times can be calculated without regard to other work. The project manager has many responsibilities and decides on ECOs once a week, during the weekly project meeting. Consistent with our observations of the process in the host company, the project manager in the example indeed decides on all ECOs on the spot during the meeting; no ECO must remain another week unaddressed.

A critical new concept not discussed in the process map Figure 1 is that the process must handle a *stream* of ECOs over time: 20 ECOs arrive, on average, per week at random points in time (including the "re-issued" ECOs mentioned in Figure 1). The arrival rate is, again, not tracked systematically. We inferred the average rate from the database of ECOs processed over the course of several months, and we cross-checked the estimate with the engineers.

Process Utilization Profiles and Bottlenecks

After having drawn a process map and collected basic data, the next step in process analysis consists of understanding the capacity and utilization profiles of the resources

involved. We now ask the question: do the resources have enough capacity to satisfy the work demanded from them by the incoming ECO stream? In other words, is total capacity consumption for each engineer and analyst less than or equal to the capacity available?

For each resource, we can calculate the *utilization* as the ratio of total capacity consumed and capacity available. For example, at CCS engineering the utilization is:

CCS Utilization = 50% = 2 hours/task x 0.5 tasks/hour / (2 people).

The simulation engineer encounters a complication, namely *set-ups*: every time he/she prepares the simulation software for a different type of problem, files must be loaded, parameters adjusted, etc. (for an overview of the set-ups encountered see [21]). The engineer, therefore, tries to regroup the ECOs in *batches* of 2 similar problems, in order to economize on the set-ups. Batching is a very old and frequently encountered principle in processes of all kinds (the earliest reference is Harris 1913 [13]). Batching has, however, a downside stemming from the time a task has to wait in order for its "cohorts" in the same batch to be processed. Thus, an individual ECO is not implemented directly on occurrence, but rather batched with other changes, thus lengthening the ECO lead time.

With batches, the resulting utilization of the simulation engineer becomes:

Simulation Utilization = 92.5% = (1.6 hrs/task + 0.25 hrs of set-up/task) x 0.5 tasks/hour

Note that the engineer must batch in order to manage the workload: with batches of 1, the utilization would be *(1.6 hrs/task + 0.5 hrs of set-up/task) x 0.5 tasks/hour = 105%*. That is, he/she would not be able to accomplish all work without overtime.

In general, we can describe the utilization for a resource as follows. We call R the overall throughput rate of the workgroup (the volume of the ECO stream to be handled, 20 per week, corresponding to 0.5 per hour). As we discussed above, R is externally given, determined by the number of ongoing projects. We call p the average processing time for a task performed by the resource in question (for example, 2 hours at CCS engineering). The simulation engineer organizes his/her work in batches of b tasks to be done together, and every time the engineer switches from one batch to another, he/she must spend s time units in set-ups.

With these problem data, the utilization of the engineer becomes $u = R\ (p + s/b)$. It is measured in %, and it consists of the fraction of time the engineer is busy with processing ($R\ p$) and set-ups ($R\ s/b$). In other words, R represents the number of "jobs" or problems that arrive, on average, per time unit, and p is the "workload" (in time units) that each job carries with it, on average. The product of the two represents the fraction of time the engineer is busy with processing this type of job.

The reader can see immediately in this expression that the average amount of time spent on set-ups decreases when the batch size *b* increases, which is, of course, the precise reason why people batch. That is, less total time is spent on set-ups if they are spread over more units in a batch.

The utilization profiles are summarized in Table 1. It implies that in total (on average in the long run), all engineers and analysts have enough capacity to accomplish their workload: all utilizations are below 100%. Simulation is closest to a full load with a utilization of 92.5%. The processing resource with the highest load in a process is referred to as a *bottleneck*. A bottleneck limits the throughput volume of the process, and close attention should be paid to it.

Station	CCS	Simulation (batch = 2)	Cost analysis	Project Manager	Purchasing
Utilization, in %	50%	92.5	37.5	8*	37.5

* The project manager decides on ECOs once per week. He spends 92% of his time on coordination activities. Note that this is similar to batching one week worth of ECOs, but it is not motivated by reducing set-ups.

Table 1: Utilization Profiles in the ECO Process

Upon discussing the utilization profiles, the project manager intuitively agreed that simulation was the bottleneck. He did pay close attention to it, but more in the sense of "fire-fighting," when long delays occurred, than in the sense of systematic process improvement.

An Explanatory Framework of Congestion

We have now determined that the process is capable of accomplishing its workload, that is, the observed long lead times do not stem from sheer overload. However, even with loops in the process, the total reported times to perform the activities do not exceed one day. In order to understand the long lead times, we must take a more detailed look at the dynamic behavior of the process.

When regarding an individual task in the ECO process, such as a design proposal by CCS engineering, we find that the total throughput time of this task comprises three components[1]:

- *Processing time:* the actual time it takes to process the task, e.g., the time it takes the engineer to analyze the data.

[1] We ignore "travel time" from one engineer to the next, as we found it in our study to happen relatively quickly, compared to the overall lead times reported above. Most of the information was submitted electronically or via fast courier services. In organizations with a lower level of electronic integration, the time an ECO spends "travelling" might be significant. Transfer times can easily be included in our framework by just describing the transfer as an activity in itself.

- *Waiting in the batch:* the time the task has to wait in order for its "cohorts" in the same batch to be processed. For example, the first ECO in the simulation engineer's batch waits for the second to arrive before the batch starts, and after the first ECO in the batch is simulated, it also waits for the second to be processed before both proceed to the next process step. In other words, batching economizes on set-ups, but it lengthens the *de facto* processing time.
- *Waiting time:* the time the task remains pending, e.g., the time the problem data remain on the engineer's desk before he/she takes action.

A familiar example of waiting time from our day-to-day life is the checkout in a supermarket where customers queue up for their turn. Other familiar examples include the check-in at an airport, telephone call centers, or restaurants. In such situations, the time it takes to get service is substantially driven by the time *before* the *actual* service starts. As waiting time is beneficial for neither the customer nor for the service provider, we also refer to it as non-value-added time.

In many manufacturing facilities, non-value-added times (mainly waiting time) account for 70 - 90% of throughput times (see, e.g., [14]). In product development organizations, the situation is similar: projects often take much longer than the work content alone suggests [1, 2]. In the ECO process in our host organization, a one-day waiting time resulted for an operation of less than one hour.

Waiting is intimately connected to *variability* in processing and work arrival patterns. To understand its effect, consider first a smoothly running assembly line where variability is absent. Jobs (such as metal parts to be assembled) arrive like "soldiers marching" in time with the line. Every operation is highly structured or automated, and thus processing times at each step are the same for all work parts. In this situation, the line can be loaded until the bottleneck (the slowest station) reaches a utilization of 100%, without any difference for the lead time (throughput time) of the line. It is simply the sum of the processing times at all stations (of course, if the line is loaded beyond the bottleneck's limit, work will start piling up).

The above-described perfectly regularized assembly line is an exception, as far as operating environments go. The situation faced in the ECO process (and in product development processes and many manufacturing processes in general) is much more difficult. First, ECOs do not arrive like "marching soldiers". Product development is a complex process that is much harder to predict than an assembly line, thus work arrives in far more random patterns. Let us call the time between two subsequent ECO arrivals the *interarrival time*[2]. Interarrival times vary considerably from one time to the next: sometimes, several ECOs appear within an hour, whereas occasionally it takes a day for the next one to be created.

[2] From the perspective of the ECO process, ECOs do externally "arrive," as their creation is prompted by unforeseeable problems.

Second, not all ECOs require the same processing time. Some are difficult and take several hours in their detailed solution design, while others require minor modifications that can be made within a few minutes. This is the case even within classes of comparable complexity, which is what we consider in our process analysis. In addition, sometimes an ECO must wait because the engineer must work urgently on a different project, which for the purpose of throughput time performance is the same as if processing took longer.[3]

Variability has an extremely detrimental effect on the processing engineer. Although the engineer has enough capacity to manage the work in the long run, random fluctuations may cause him/her to "fall behind" temporarily, when a few ECOs happen to arrive in quick succession, or when an ECO proves difficult, and takes much longer than the normal. During this time, a backlog of unprocessed ECOs accumulates for this engineer. The fact that the engineer has enough capacity in the long run implies that he/she will be able, eventually, to "work off" this backlog. Such a backlog corresponds, of course, to waiting. But how long will this waiting be?

The time to work off the backlog strongly depends on the "slack capacity," or, in other words, 100% minus the capacity utilization, of the engineer. The higher the utilization, the longer it will take him/her to work off the backlog (in addition to handling the extra work arriving in the meantime). It turns out that when the utilization is high (slack is low), surprisingly large backlogs may occur, and they may stay for an unexpectedly long time before being worked off. This explains the long waiting times we observed in the ECO process.

"Slack capacity" in this context does not mean that the engineer is sitting around idle. As in all professional environments, there is always work, in the form of problem-solving or the creating of new ideas or designs. Slack means that there is some "background" work in the sense that it can be put aside at times of high pressure. Such slack provides the engineer with the flexibility to respond to variability-related backlogs. We now need to make this intuitive explanation precise.

We need to introduce measures for variability in addition to those of throughput rate *R*, processing time *p*, batch size *b* and set-up time *s* as defined above. A natural and widely-known candidate for this is the *standard deviation*. We can measure the standard deviation of the interarrival time and of the processing time at each resource (engineer or analyst). This measure is, however, not perfect because it is an absolute measure: if the CCS engineer's and the accounting analyst's tasks both have a standard deviation of 1 hour, are they equally variable? The answer is no, since the CCS engineer's standard deviation is only a fraction of the average task time, while the analyst's standard deviation

[3] For simplicity of exposition, we have incorporated capacity needs of other projects by "cutting the group size in half" in Figure 1. This is correct only as a long run average, but a simplification for every day dynamics: the engineers in Figure 1 share work across projects. Sometimes they are held up by other projects, and sometimes they get help from the other half of the group, which overall increases variability of processing.

is larger than the mean task time. Therefore, a better measure of variability is the *ratio of the standard deviation over the mean*. It is referred to as the coefficient of variation (CV), and we call it CV_a for the ECO arrivals and CV_p for the processing times.

ECOs arrive irregularly, and vary in their complexity, so the CVs of both arrivals and processing are relatively high. The CVs are not measured and tracked in our host organization, which is not unusual. Even in organizations executing routine processes, variability is often not measured. We, therefore, had to estimate the CVs. A reasonable and widely used estimate is a value of 1 (the standard deviation equals the mean[4]).

Our intuitive explanation above suggests that waiting times will increase both with the variability (since the chance of "temporary falling behind" increases) and the utilization (since the slack to work off the backlog decreases). The Pollaczek-Khintchine formula makes the above explained intuition precise[5]:

$$\text{Wait} = \frac{1}{2}(CV_a^2 + CV_p^2)\frac{u}{1-u}(s+bp). \qquad (1)$$

Note that u is the utilization as we explained it above, and $(s + bp)$ is the time needed to process one batch. The formula behaves as we expected from the above explanation. When utilization goes from 90% to 95%, the ratio $u/(1-u)$ goes from 9 to 19, which corresponds to a very steep increase at high utilizations. The waiting time becomes more and more dominant as utilization increases. In addition, wait increases quadratically with the two coefficients of variation.

The highest utilization encountered in our process is 92.5% at simulation. Due to the very steep increase at high utilizations, waiting in Formula (1) is *much* higher at the bottleneck station than at less loaded engineers. This explains some of the long waiting time observed in the process.

The average *total* throughput time is the sum of the average waiting and processing times of the batch, including set-up and the average time an individual task waits for its batch cohorts to be done. We can summarize the throughput time in the following formula (where u is now replaced by $R(p + s/b)$).[6] The CV for the batch (marked by the upper bar) may differ from the CV of individual processing times[7] because some averaging occurs over a batch. For example, it is unlikely that the first and the second problem in a batch of 2 *both* have a very long or a very short processing time. Thus, the processing

[4] A CV of 1 technically corresponds to exponentially distributed processing and interarrival times. This is a standard distribution used for high-variability situations, and it has repeatedly proven a good approximation [1,15,18]. In many well-controlled manufacturing environments, the CV is much lower than 1.

[5] The Pollaczek-Khintchine formula can be looked up in many books on manufacturing or queueing, e.g., Hopp and Spearman [15]. It holds as an approximation with good accuracy when utilization is high, which is exactly when waiting times matter.

[6] This formula is based on work by Karmarkar *et al.* [16]. An intuitive derivation can be found in Hopp and Spearman [15], p. 290 f. Again, this is an approximation that is accurate for high utilization levels.

[7] This holds if the processing times are not strongly correlated.

time of the full batch tends more toward the average, exhibiting lower variability than the individual problem to be solved.

$$\text{Throughput time} = \frac{1}{2}\left(\overline{CV^2}_a + \overline{CV^2}_p\right)\frac{R(p+s/b)}{1-R(p+s/b)}(s+bp)+s+bp. \qquad (2)$$

This formula characterizes the typical throughput time behavior at the individual resource, which is graphically represented in Figure 2. On the left-hand side, we see that as utilization approaches full load, or 100%, the throughput time dramatically increases because the engineers no longer have the slack to deal with unexpected events (expression *u/(1-u)* in the formula). Work spends more and more time in the in-baskets.

The right-hand side of Figure 2 demonstrates how batching mitigates the congestion problem by spreading the set-up over more tasks and thus reducing utilization: *u = R (p + s/b)*. This comes, however, at a cost: when the batch size becomes large, the marginal congestion benefit decreases, but the waiting time for the batch cohorts continues to grow linearly with the batch size. Thus, there exists a point with a "best" batch size.

Figure 2 here

If one succeeds in reducing variability or set-up times, two positive effects result: first, the whole throughput time curve shifts downward, and the optimal batch size shrinks (shifts to the left) as well. When the set-up time *s* is very small, batching no longer has a benefit. This can be seen in the formula: if *s=0*, the utilization becomes independent of batch size, since no set-up work is saved by batching.

This analysis applies to a single engineer or workgroup. The strong non-linearity of the congestion effect increases the importance of the bottlenecks: if one workgroup alone is highly utilized, its throughput time will dominate that of the whole process. In a process consisting of several operations, such as our ECO process, additional interactions exacerbate congestion even further. For example, purchasing can only start its work if *all* previous activities have been completed, which further contributes to long waiting times: only after cost analysis is done *and* the project manager has agreed to the proposed solution can parts be ordered.

Station	**CCS**	**Simulation (*b* =2)**	**Cost analysis**	**Project Manager**	**Purchasing**
TPT (average throughput time), in hours, by formula (2)	2*	25	1	20**	1
TPT, in hours, simulated***	3	25	1	20	8
Total TPT, in hours	56				

* This TPT is calculated with an "average" processing time p = 1 hour in the utilization, since the two engineers can work on two ECOs in parallel. Processing itself still takes 2 hours.

** The project manager decides on ECOs once per week, so on average, an ECO is pending half a week.

*** Discrete-event simulation with the software package SLAM; analysis runs over 40 000 jobs.

Table 2: Average Throughput Times in the ECO Process

With the help of the above formula, we can now find the average throughput time (TPT) at each step, which is summarized in the first row of Table 2.[8] In addition, we show the TPT as found by discrete-event simulation with a commercial software package in row 2. The average throughput times at the stations of the process add up to a total of 56 hours. It is the sum of all individual TPTs, except for the time for the cost analysis, which runs in parallel to, and is dominated by, the TPT for the project manager's decision.

The results as calculated by Formula (2) and as simulated (rows 1 and 2 of the table) largely agree, with one exception that is worth discussing: the formula predicts a fast turnaround at purchasing, while the simulation shows an average delay of a whole day. This is because we assumed in the formula, for lack of better information, a CV of 1 for the incoming job flow at purchasing. It turns out, however, that as the project manager releases a large number of ECOs (about 20) once a week, they arrive at purchasing in large "packs," and purchasing must "work off" this backlog over the next two days. This leads to an average delay of one day. In the context of the formula, such "lumpiness" of an ECO arrival stream corresponds to high variance, and thus an SCV much higher than 1.

The important implication of Table 2 is that *an ECO takes on average almost one and a half weeks (and in some cases much longer) to go through one iteration of the ECO process, although it has on average only 5 hours' work invested in it.* With 25 hours on average, the bottleneck station (simulation) contributes almost half of the throughput time. The project manager also slows down the process because he authorizes ECOs only once a week, adding, on average, 20 hours to the TPT.

Table 2 summarizes the *average* TPT of an ECO. However, the average is not sufficient to describe the performance of this process: throughput times are themselves variable, so they must be described in the form of a *distribution*. The tail of this distribution determines the *service level* the process can offer. Figure 3 demonstrates that the project manager needs to allow as much as *2.5 weeks* beforehand, in order to be 90% confident that an ECO will indeed be resolved. For example, if the project manager wants to estimate whether a newly-arisen ECO will be affected by another change in another component, he needs to be aware of the fact that the ECO may remain open for *three* weeks, not only 1.5 as the average TPT suggests.

Figure 3 here

[8] The stations are analyzed as if they were isolated from one another. This is exact when the CVs are equal to 1, and it provides a reasonable approximation for more general cases. In addition, the SCV of a batch of two can be shown to be 0.5 (the distribution of processing two exponentially distributed tasks in series becomes an Erlang-2 distribution).

We used the throughput time average and distribution from Table 2 and Figure 3 to calibrate our model analysis with the project manager. This is necessary (and usually done) when the input data for a model is not well tracked and must be estimated. The project manager has a good feeling for the resulting throughput time distribution, so arrival rate and processing time parameters were slightly adjusted until the model matched the throughput time performance observed by the project manager. [9]

The project manager, thus, recognized the throughput time performance. However, he was shocked to see the ratio of processing time to total throughput time (and thus the value added portion) implied by this analysis. Being an electrical engineer by training, he readily saw the conceptual similarity between the long throughput times in his organization and the response time performance in computer networks, which is also driven by congestion stemming from scarce capacity combined with variability.

This example shows how congestion can lead to TPTs that are many times the raw processing time. Thus, the example contributes to explaining the throughput times observed in the CCS case. In addition, Figure 3 explains the "self-fulfilling prophecy syndrome": the engineers know very well, of course, that the lead time for an individual ECO follows a distribution (not only for the process as a whole, as shown in Figure 3, but also at each engineer) and cannot be predicted beforehand. They know this from experience, although they typically do not know the precise shape of the distribution. Moreover, no one wants to be caught not living up to his/her promises. So what does an engineer answer when asked when an ECO can be resolved? He/she will give the 90^{th} or 95^{th} percentile of the distribution. However, if every step in the process indicates the 90^{th} percentile for the expected TPT, the resulting estimate for the process as a whole will be ridiculously conservative and make any planning very hard for the project manager. The project manager in our host organization complained bitterly about this planning paradox, which occurs not only in the ECO process but which seems to be a typical problem in project management [12].

Discussion: Opportunities for Improvement

The interesting question for managers, of course, is what concrete improvement possibilities exist. Congestion problems can be easily avoided by just adding extra capacity, although, for obvious financial and political reasons, this approach is out of the question. Similarly, variability can be controlled (for example, by rigorous quality management) in many manufacturing environments, but it is inherent in the ECO process, as each ECO problem is unpredictable. The objective of the improvement methods presented below (summarized in Figure 4) is to improve ECO lead times without adding extra capacity and without dreaming of a regularized process.

[9] The reader may recall that in the real process in our host organization, some ECOs had to go through several problem-solving loops, which delayed them even further. This was incorporated by iteratively adjusting the ECO arrival rate, together with the project manager, until the throughput time distribution was approximately correct.

Opportunities for Improvement: Flexible Capacity

The first improvement strategy addresses the basic source of queueing problems, the mismatch between when capacity is needed and when capacity is provided, by increasing the flexibility of the server. Remember that in the discussion above, as well as in the presented formula, the utilization (the relationship between capacity available and capacity required) must be less than 100% on average, as otherwise one would fall behind in the long run. Thus, if it were possible to provide the server with capacity at the moment it is required, queueing could be completely avoided. Now consider the simulation engineer, who faces the highest utilization of all with 92.5%, working up to two hours' overtime per week (for example, during lunch or at night), whenever an ECO backlog piles up on his/her desk. On the other hand, he/she may go home earlier if the workload is light. Thus, in the long run, the engineer does not work more than 40 hours a week, but the work is provided just at the time when it is needed.

With this flexibility, the *effective* utilization factor in our example goes down to 88% (keeping the batch size constant at 2). Formula (2) and discrete-event simulation both show that this reduces the average throughput time for simulation from 25 hours to 14 hours, a reduction of 44% at this station.

Figure 4 here

Opportunities for Improvement: Merging Tasks

The first improvement strategy is targeted at an individual server, whereas the second strategy of merging tasks looks at multiple servers collectively. Consider the three tasks of financial analysis, approval by the module project manager, and that of ordering parts. The ECO in the current situation must queue at each of the three servers, risking waiting times at each of them. In our example, we introduce a manager whose time is devoted to performing cost analyses and ordering parts, and who also has the authority to approve ECOs on the spot. This manager can approve ECOs flexibly during the week (not only in the weekly meeting).

In our example, this corresponds to one server facing an average processing time per ECO of $(0.75 + 0.17 + 0.75) = 1.67$ hours, or a utilization of 83%. As a result, the total average throughput time of this part of the process is reduced from 28 to 10 hours, a reduction of 64%. In the company we studied, we indeed found a number of "ECO managers" who combined the work that was previously done by separate organizational entities. We also observed that some module project managers approved ECOs on the spot while walking around the engineering cubicles (rather than only once a week).

Opportunities for Improvement: Balancing the Workload

The third strategy of balancing workload is based on the observation that process lead times are frequently dominated by one single activity, referred to as the "bottleneck" activity, the one with the highest utilization, and which determines the speed of the whole system. In our CCS development case, the bottleneck is easily identified as the aerodynamics simulation engineer with a utilization of 88%. Because of the high expertise required for the simulation activity, the corresponding group is permanently short of engineers. This makes the group almost incapable of responding quickly to the requested ECO evaluations. To make matters worse, the group spends a significant amount of its time reworking CAD models created by other CCS engineers, in order to bring the models to an accuracy level required for the simulation software. Thus, about one third of this group's precious time is wasted on work that could equally be performed by CCS engineers.

If this preparation work is shifted in our example from the simulation engineer to the CCS engineers (who do not need to batch – they can perform the set-up for every ECO), capacity utilizations become better balanced, at 62.5% for CCS engineering and 80% for simulation engineering. As a result, average CCS TPT goes up from 3 to 4 hours, but simulation time shrinks from 25 to 8 hours. Thus, the total average TPT for both stations is reduced from 28 to 12 hours, or by 57%. This example demonstrates how non-linear the impact of utilization on TPT is: the gain from reducing the utilization of simulation from 92 to 80% far outweighs the loss from increasing CCS's utilization from 50% to 63%.

Opportunities for Improvement: Pooling

The fourth strategy, that of pooling, or sharing workloads, among engineers, is based on reducing specialization in the development organization, requiring the capability of the engineers to assume a broader technical responsibility. Pooling is often efficient from a queueing perspective for three reasons. First, utilizations are balanced within the pooled group. Second, it cannot happen that one worker is starved of work while another has tasks waiting in his/her in-basket. Third, if one individual ECO happens to be very complicated and time-consuming, the subsequent ones are not "stuck" behind it, but can (at least slowly) bypass it via the other pooled servers.

Pooling, however, may also have a downside. First, and most obviously, the engineers may have to go through "mental set-ups" and become less productive if spread across different tasks. Second, pooling may increase the processing variability if different types of jobs, although homogeneous among themselves, but very different across types, are pooled. If, for example, ECOs for the filter box (requiring air flow analyses) and for electrical motors (requiring electrical design) were to be pooled, one engineer would be responsible for total CCS ECOs and face a multitude of very different tasks. This could

increase the variability of the workload and thus queueing effects, even if the engineer was perfectly cross-trained.[10]

Pooling may not be possible for pure research tasks, which require profound expertise in one specific domain. However, we found it to be typically applicable for engineering tasks such as ECOs, which entail relatively standard operations. Engineers can share work on similar components, such as air and water ducts for different parts of the CCS, or on the analogous components for different car development projects which progress in parallel.

We are not including the pooling improvements in our numerical example, because the trade-offs involved are complex and would force us to complicate the example to an extent which would hinder simple exposition. Of course, the trade-offs are accessible to evaluation by simulation modeling.

Opportunities for Improvement: Managing Batching Problems

Before we discuss concrete actions aimed at reducing batching, we demonstrate the potentially large effect using our numerical example. Keeping all processing times unchanged for comparison, we first ask whether a batch size of two at the simulation group is a good choice. Examining the utilization, it becomes evident that the simulation group has no other choice but to batch, as they would become overloaded (u = 100%) if they processed the problems as they come, incurring a set-up every time. Furthermore, a larger batch size offers no further improvement, i.e., the waiting within the batch more than offsets the reduced congestion.

However, a reduction in the set-up time makes a great difference. Suppose the data preparation could be shortened from 30 minutes to 5 (e.g., via more compatible file formats and more consistent preparation of the data by the problem-generating engineers. These are two proposals under discussion at the company).

Holding the batch size constant at b =2, utilization falls to 82%, and the average TPT of the simulation engineer is reduced from 25 hours to 10, a reduction of 60%. Moreover, batching is no longer necessary: the simulation engineer can now process the problems as they come, in spite of the set-ups. This increases utilization to 84% (in comparison to maintaining the batch) and also variability (as there is no longer averaging of processing times within a batch), while decreasing the batch processing time. In our example, the trade-off comes out exactly even: reducing the batch to 1 is equivalent to keeping the batch at 2, with an average TPT of 10 hours.

After having examined the importance of batching and set-ups in the example, we now discuss possible actions to reduce the reasons that make batching necessary.

[10] The trade-off is too complicated to be meaningfully included in our simplified example, so we provide no estimate of the potential benefit.

The only way of improving the trade-off from Figure 2 (between having to incur set-ups and incurring long batch processing times) lies in addressing the sources of batching, i.e. the set-ups, reducing batch sizes through a set-up cost reduction. In our companion paper [22], we mention communication technologies and "rapid" or "virtual prototypes" as possible ways of reducing set-ups [8,19,24].

The improvement most relevant to the process example described in this paper addresses mental set-ups. These refer to the fact that an engineer, returning to a problem after having worked on something else, needs some time to understand and master the problem again. Moreover, he/she needs to go through the physical action of re-loading all the CAD files and data sources that are directly relevant to the problem. Our host company has already partially resolved this by allowing the engineer to access all electronic drawings from his/her CAD workstation. Most of the more senior engineers we talked to still remember vividly the time when an engineer had to go to the drawing archives and physically take out the drawings for a vehicle subsystem.

Despite the progress of CAD technology that we witnessed in our host organization, substantial differences persist across companies as to how easily these systems can be used and thus how large the set-ups for getting started are. While some companies have achieved easy-to-use CAD [24], many others still wrestle with complicated CAD systems and a lack of CAD-trained engineers. Engineers who schedule themselves special "CAD-days", as they are forced to reserve a CAD station in advance, are still the rule in some industries. A second opportunity of reducing mental set-ups lies in the division of work between engineers. It is advantageous to have engineers devote their time to components requiring profound functional expertise (e.g., ASIC technology in the control unit of the CCS system). However, in situations more integrative in nature (e.g., packaging of a cooling circuit), it may be better to have an engineer assigned to a vehicle project. Aligning work assignments with the knowledge requirements of the tasks saves the engineer substantial change-over costs between technologies or projects.

Opportunities for Improvement: Incentives

The "self-fulfilling prophecy" paradox, which makes process lead time estimates over-conservative, must be addressed by changing incentives. Whenever people in an organization are held responsible for meeting the promised deadlines for each individual task, they will react by giving very conservative estimates. Goldratt [12] proposes that the project manager elicits estimates from the engineers which they fail to meet half of the time, and actually enforces this "success rate"![11]

However, this requires, first, that engineers are also held accountable for the average of the estimates, since they would otherwise still be free to under-promise and then to

[11] This policy would get the median of the distribution, not the mean, which resolves the conservativeness bias.

procrastinate. Second, such a measure requires a strong will by the project manager to let go of trying to control each individual ECO, and look only at the distribution, which goes against all natural project management instincts. The project manager in our host organization was not willing to make that step.

Opportunities for Improvement: Summary

The improvement approaches are summarized in Figure 4. Each of these strategies can dramatically reduce ECO lead times without taking the brute force measure of adding capacity. Above, we have discussed the improvements in isolation. In combination, their effect is even more powerful, although they do not add linearly, but with decreasing returns (as congestion decreases, improvements become less drastic).

In order to see the benefit of combining the above improvements, consider the situation where simulation capacity is flexible, set-up times have been reduced, and the accounting, purchasing and ECO authorization activities have been merged (pooling and capacity offloading are less urgent now and thus not included). Figure 5 shows the resulting TPT of the ECO process: not only is the mean reduced almost by a factor of three to 20 hours, but also the tail of the TPT distribution has shrunk to one week. The project manager can now be reasonably certain that an ECO will be done within a week, thus greatly reducing the risk of new ECOs interfering with it [10,17].

Thus, the resulting streamlined process will also be advantageous from a quality perspective, as shorter lead times reduce the risk of rework being incurred by interacting changes [21, 22, 23, 25]. Finally, engineers can obtain immediate feedback on the effectiveness of their changes, which helps them to develop a better understanding of problems and solutions (see [14, 21]).

Figure 5 here

The above-discussed improvement actions may require significant investments, for example, when systems have to be developed, or when engineers have to be trained, or in the case of the organization having to be changed when merging tasks. In order to evaluate the attractiveness of such process improvements, the *value of time* must be estimated: what is it worth to the organization to gain one week in time-to-market? Models of the value of time exist (e.g., [3,9,26]), but not in sufficiently operational form to be used for the evaluation of improvements in the ECO process. This is an important area of future research.

Implications for the Host Company

The above-described findings were presented to the project manager and several key functional department heads in development. The group was astonished by the amount of non-value added time in the ECO process, but accepted the results of the analysis once they thought about it and connected it to their daily experience.

Two implications of the study were accepted as immediately implementable: making capacity more flexible via overtime, and continuous ECO approval rather than only once a week in the project meeting. Two other recommendations were considered for implementation, after some more study and preparation. First, a training program was discussed to enable the CCS engineers of performing data and file preparation, thus off-loading some tasks from the simulation group. Second, a *partial* merging of tasks was proposed, where all engineers involved in one specific ECO would be temporarily dedicated to solving this ECO, for example, for a week. This would be tested on important ECOs, which would thus be solved very quickly, without anything else intervening.

Pooling of resources was not further pursued. For example, it was not deemed appropriate to pool project management and accounting or purchasing tasks, as this was not compatible with the organizational structure. Batching was pursued in a broader context, by attempting to convince all parties involved to communicate earlier and not "sit" on their partial solutions longer than absolutely necessary. This is further discussed in detail in [22].

Summary

In this article, we have outlined a process-based view of ECO management. We have shown that many of the problems related to ECOs have their roots in a complicated and congested administrative process. Additional reasons for long process throughput times are notably congestion and batching. While previous studies have hinted at these causes for waiting time in the context of manufacturing and services [6,11,15] or on a high aggregation level with complete development projects [1,2,4,5], we show in this article how congestion and batching influence engineering processes at a more detailed level.

Our analysis provides a theoretical explanation for the data that we collected in [22], as well as for previous studies, especially the one by Blackburn [4, 5]. It also provides a starting point in the search of improvement strategies, as we have shown in the cases of flexible work times, the grouping of several tasks, workload balancing, the pooling of resources, and the reduction of set-up times.

The objective of the processing network framework in this article is to provide a conceptual explanation for some of the phenomena that we (and other researchers [5, 27]) have observed. Although previous studies have shown that it is possible to apply queueing concepts such as arrival and service rates, variability and utilization to engineering organizations, considerable effort is required to operationalize and measure these concepts in an ongoing project. As the objective of the present article is more qualitative than quantitative, we have not yet fully addressed the corresponding methods of data collection (e.g. how to measure utilization) and implementation (e.g. managing organizational change). Further research will be required also along these lines.

The model presented here simplifies the complexity of engineering projects in order to illuminate the structure of the problem. Future research will have to provide richer, more detailed models that better describe the multitude of tasks flowing in the processing network called "development organization". In particular, the methodology applied in this article can be used to estimate the benefits from using CAx technologies in managing ECOs. Such technologies may in the future be capable of automatically detecting problems in the current design (to some degree, this capability already exists, e.g., for fit problems in packaging). Automatic problem detection and ease of including changes in virtual prototypes will bring about a fundamental reconsideration of the ECO process.

We hope that, based on the example of the ECO process, this article contributes to a view of NPD as a process that can be managed in order to achieve fast turnaround times, without having to compromise the creative elements of engineering problem-solving.

References

1. Adler, P. S., A. Mandelbaum, V. Nguyen, and E. Schwerer. "From Project to Process Management: An Empirically Based Framework for Analyzing Product Development Time," *Management Science* 41, 1995, 458 - 484.
2. Adler, P. S., A. Mandelbaum, V. Nguyen, and E. Schwerer. "Getting the Most Out of Your Product Development Process," *Harvard Business Review*, March - April 1996, 134 - 152.
3. Bayus, B. L. "Speed-to-Market and New Product Performance Tradeoffs," *Journal of Product Innovation Management* 14, 1997, 485 - 497.
4. Blackburn, J. D. "New Product Development: The New Time Wars," Chapter 5 in: *Time-Based Competition*, Homewood: Business One Irwin, 1991.
5. Blackburn, J. D., "Time Based Competition: White Collar Activities," *Business Horizon*, July-August 1992.
6. Chen, H., J. M. Harrison, A. Mandelbaum, A. Van Ackere, and L. M. Wein. "Empirical Evaluation of a Queueing Network Model for Semiconductor Wafer Fabrication." *Operations Research* 36, 1988, 202 - 215.
7. Clark, K. B. and T. Fujimoto, *Product Development Performance: Strategy, Organization and Management in the World Auto Industry*, Harvard Business School Press, 1991, Cambridge.
8. Cusumano, M. A., and R. W. Selby, *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technologies, Shapes Markets, and Manages People*, New York: The Free Press, 1995.
9. Datar, S., C. C. Jordan, S. Kekre, S. Rajiv, and K. Srinivasan. "Advantages of Time-Based New Product Development in a Fast Cycle Industry." *Journal of Marketing Research* 34, 1997, 36 - 49.

10. Eastman, R. M., “Engineering Information Release Prior to Final Design Freeze,” *IEEE Transactions on Engineering Management* EM-27, 1980, 37 - 41
11. Fitzsimmons, J. A., and M. J. Fitzsimmons. *Service Management for Competitive Advantage*. McGraw Hill 1994.
12. Goldratt, E. M. *Critical Chain*. Great Barrington: The North River Press, 1997.
13. Hammer, M., and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution.* New York, Harper Business, 1993.
14. Harris, F. W. “How many Parts to Make at Once?” *Factory: The Magazine of Management* 10(2), 1913, 135-152
15. Hopp, W. J., and M. L. Spearman. *Factory Physics*. Irwin, 1996.
16. Karmarkar, U. S., S. Kekre, S. Kekre, and S. Freeman. “Lot Sizing and Lead Time in a Manufacturing Cell.” *Interfaces* 15, 1985, 1 - 9.
17. Krishnan, V. “Managing the Simultaneous Execution of Coupled Phases in Concurrent Product Development.” IEEE Transactions on Engineering Management, Vol. 43, No. 2, May, 1996.
18. Loch, C. H. “Operations Management and Reengineering,” *European Management Journal*, June 1998.
19. Sabbagh, K. *Twenty-First Century Jet*. New York: Scribner, 1996
20. Soderberg, L. G. “Facing Up to the Engineering Gap.” *The McKinsey Quarterly*, Spring 1989.
21. Terwiesch, C., C. H. Loch, and M. Niederkofler. “Managing Uncertainty in Concurrent Engineering.” *Proceedings of the 3rd EIASM International Product Development Conference*, 1996, 693 - 706
22. Terwiesch, C., and C. H. Loch. “Managing the Process of Engineering Change Orders.” INSEAD Working-Paper, revised, March 1998.
23. Terwiesch, C., C. H. Loch, and A: De Meyer. “A Framework for Exchanging Preliminary Information in Concurrent Engineering Processes, INSEAD Working Paper, November 1997.
24. Thomke, S. H. “Managing Experimentation in the Design of New Products and Processes.” Harvard Business School Working Paper 96-037, 1996.
25. Ulrich, K. “The Role of Product Architecture in the Manufacturing Firm.” *Research Policy* 24, 1995, 419 - 440.
26. Vesey, J. T. “The New Competitors: They Think in Terms of Speed to Market.” *Academy of Management Executive* 5, 1991, 23 - 33.
27. Wheelwright, S.C, and K.B. Clark, *Revolutionizing Product Development*. New York: The Free Press, 1992.

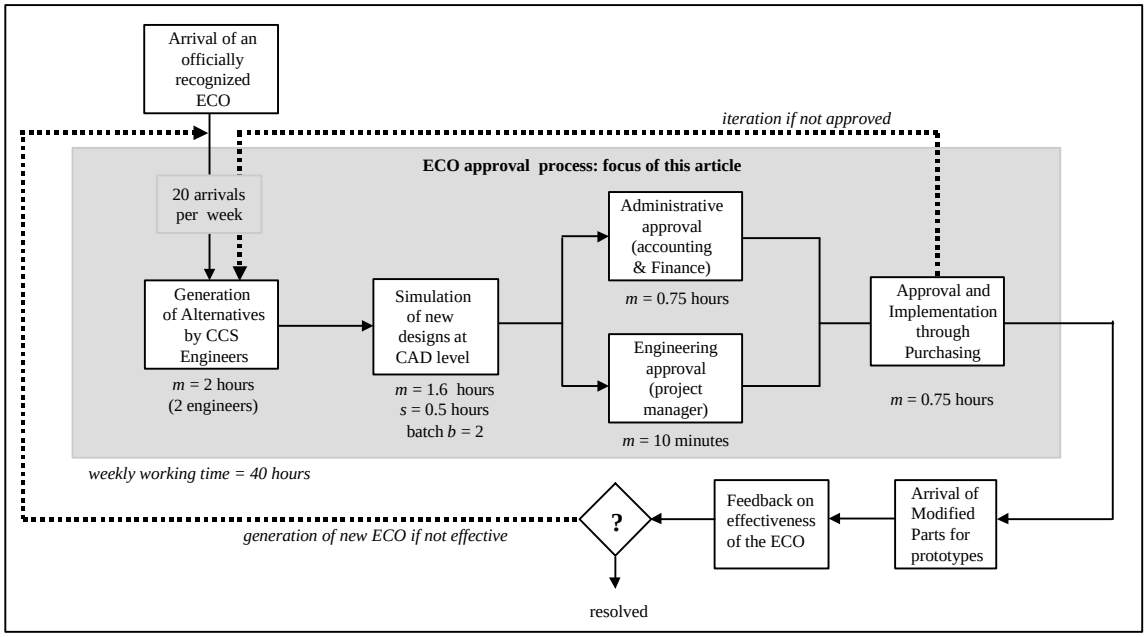

Figure 1: An Illustration of the ECO Process

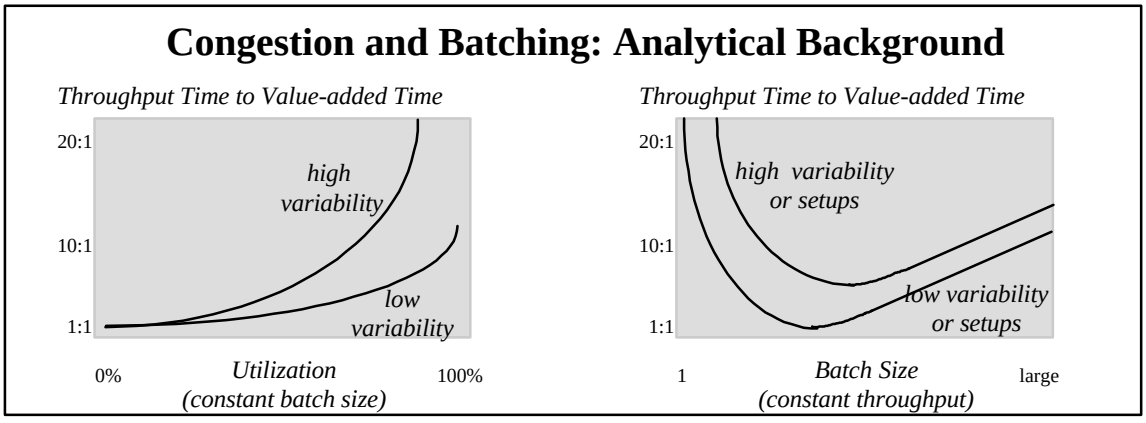

Figure 2: Congestion and Batching in the ECO Process

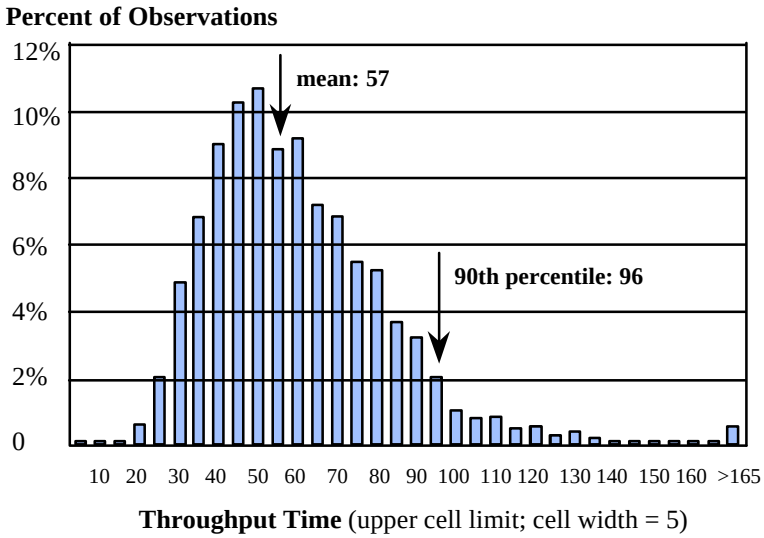

Figure 3: ECO Throughput Time Distribution

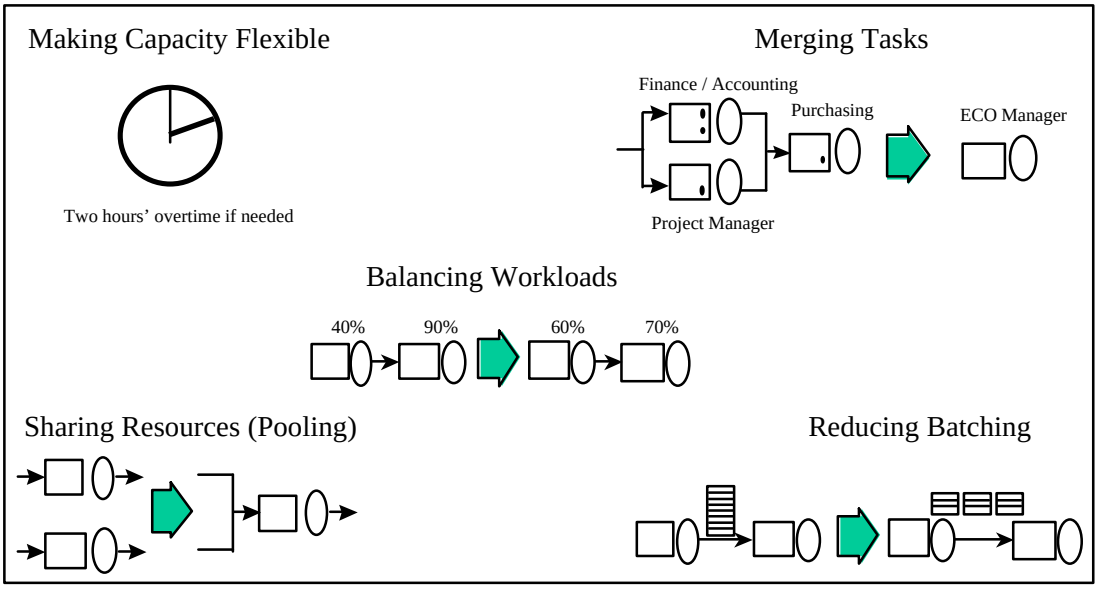

Figure 4: Five Strategies for Reducing Congestion and Batching in the ECO Process

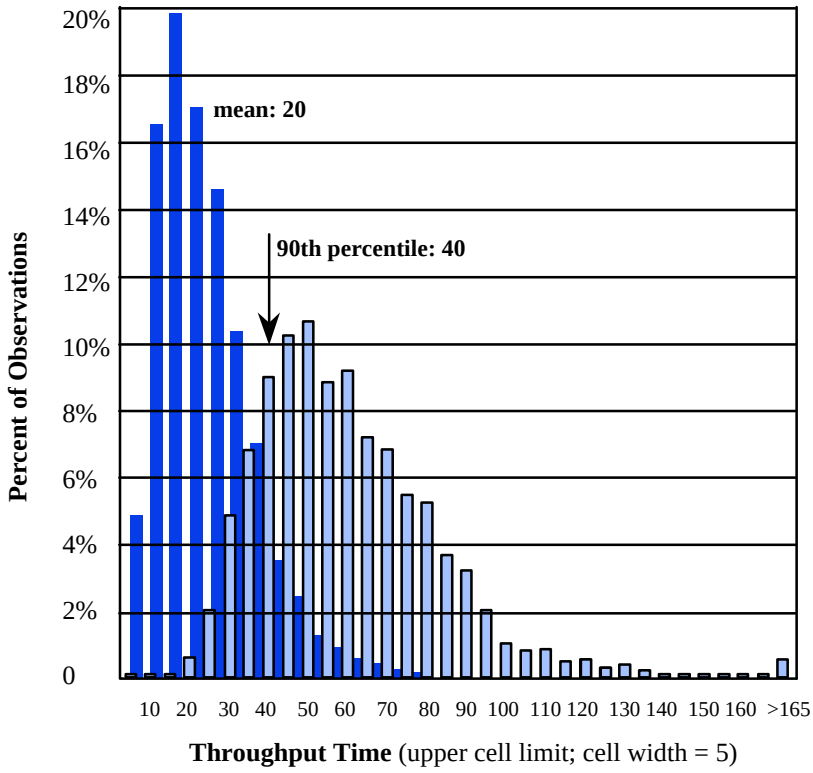

Figure 5: ECO Throughput Time Distributions before and After Improvements