Concurrent Engineering
http://cer.sagepub.com/

Sequential versus Concurrent Engineering––An Analogy

Biren Prasad

Concurrent Engineering 1995 3: 250
DOI: 10.1177/1063293X9500300401

CONCURRENT ENGINEERING: Research and Applications

Sequential versus Concurrent Engineering—An Analogy

Biren Prasad

Electronic Data System, Delphi Automotive Systems, General Motors, Troy, MI

1. Introduction

At the beginning of the information age, product manufacturers were wrongly led to believe that the control of information flow was the key component to success. Many manufacturers, thus, gave adequate power to the functional organization to control the flow of information. The organization common sense for this "era of control" then was hierarchical setup that led to the creation of many tall "silos"–multilayered control structures. Product was task-focused and engineering was sequencing-based. During this age of control, linear product development appeared to be the right thing to do. The product design and development (PDD) process was obtained in phases. The phases were essentially queued in consecutive order and segmented with some hard breaks between phases. Inputs from individual departments were sought in an ordered sequence as shown in Figure 1. Tasks were clearly labeled as requirements definition, product definition, process definition, or delivery and support. The tasks of the upstream phase were completed before tasks for a downstream phase were started. For example, requirements preceded product definition. Design engineers dominated the process (often drawing-based) in the earliest stages of conceptual design and preliminary development. Marketing experts gave the needs to the designers, who determined product specifications.

In most sequential engineering processes, it was customary for the market research department to determine customer or user needs and throw its sales projection data over the wall to planning. The planning department developed the technical requirements for the product and threw its specifications over the wall to the product engineering group. This group then designed and developed the product on their own, in near-complete isolation from the production process. Later, the prototype was handed over to manufacturing so that their engineers could arrange to manufacture the product on a large scale. Seldom when a product develops this way will it go to production in one shot. Several engineering change orders are issued to fix problems, but due to time pressure, attention can only be given to fix the major show-stoppers (problems). Some of the major problems manufacturing engineers encountered were:

- unsuitable product design for production
- unavailability of adequate manufacturing equipment
- tight tolerance, which could lead to extra work and high scrap generation
- problem with parts assembly
- inability to utilize the existing production equipment, tooling, automatic assembly, etc.

The feedback at each stage was in the form of errors, changes, and corrections. It took several rounds of negotiation and a long lead time before all pertinent design and manufacturing problems (errors, changes, and corrections) could be resolved. After a period of time, procurement experts then get involved to ensure that the necessary parts and materials are available for the assembly process. Next, marketing and sales personnel come in to introduce the product to its targeted market. The Test Analysis and Fix (TAAF) process is often used to ensure component-level reliability. Analysis is conducted off-line and is treated as a way to fix design problems, not as a way to prevent them from happening. Should a design change be necessary, the control is returned to the front and the serial process is repeated. These difficulties occurred as a result of a process, which is commonly referred to today as "Serial Engineering."

2. Serial Engineering Analogy

In Serial Engineering, each design phase starts mostly when the previous one is completed. During this process, any incomplete data is passed through a function or phase and augmented with new data only to be passed on to the next phase. The process is analogous to a relay race team that has to run x miles between the collective efforts of n runners (see Figure 2). Each runner is placed at a post equidistantly (x/n miles) apart. Each runner is starting from a stationary position. The first runner takes off and passes the baton to the second runner at the end of x/n miles, the second to the third, the third to the fourth, and so on. The last (nth) runner finishes the race.

If v_i is the maximum speed an ith runner can attain starting from its stationary position, then average speed to reach

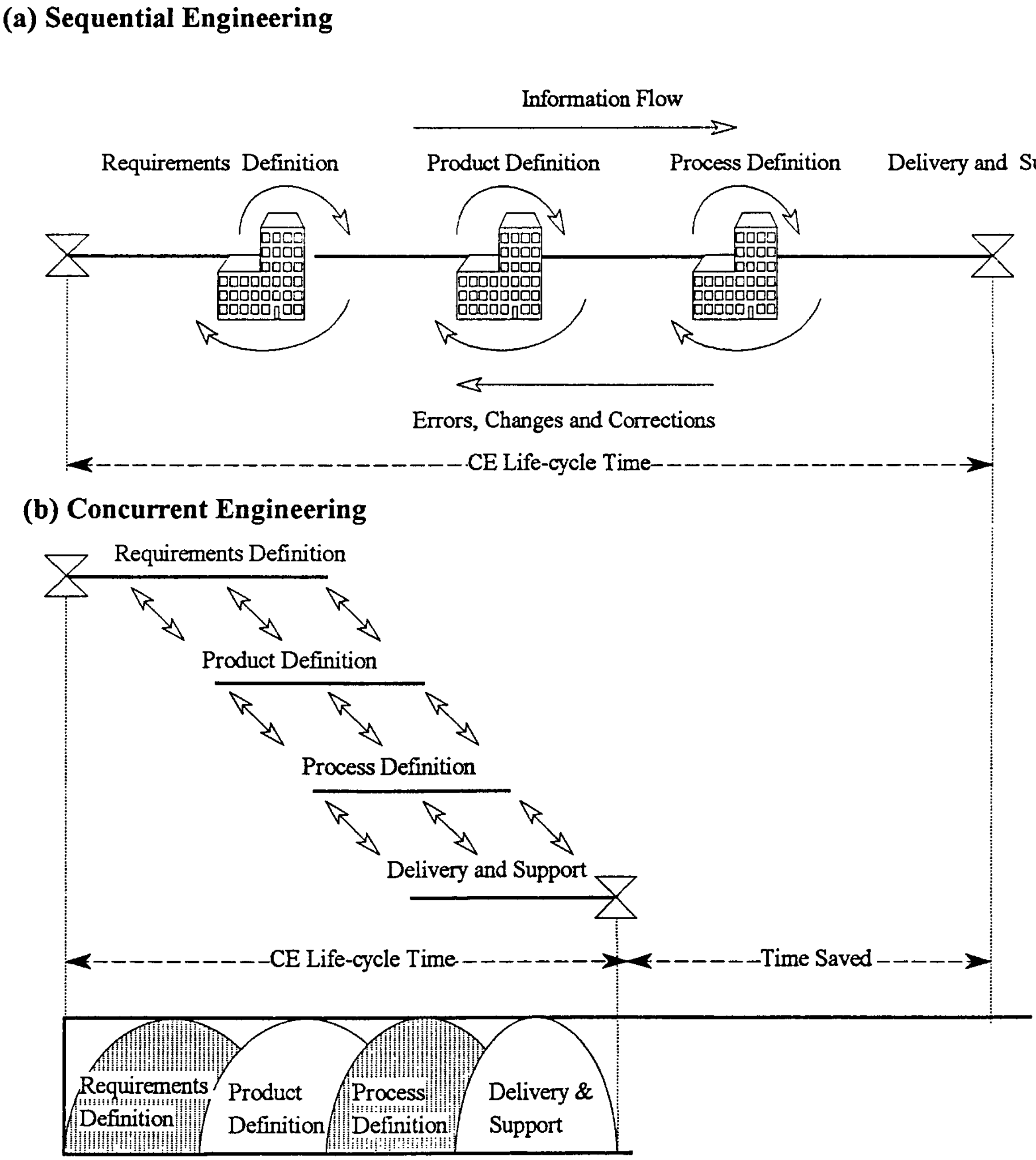

Figure 1. Sequential versus Concurrent Engineering.

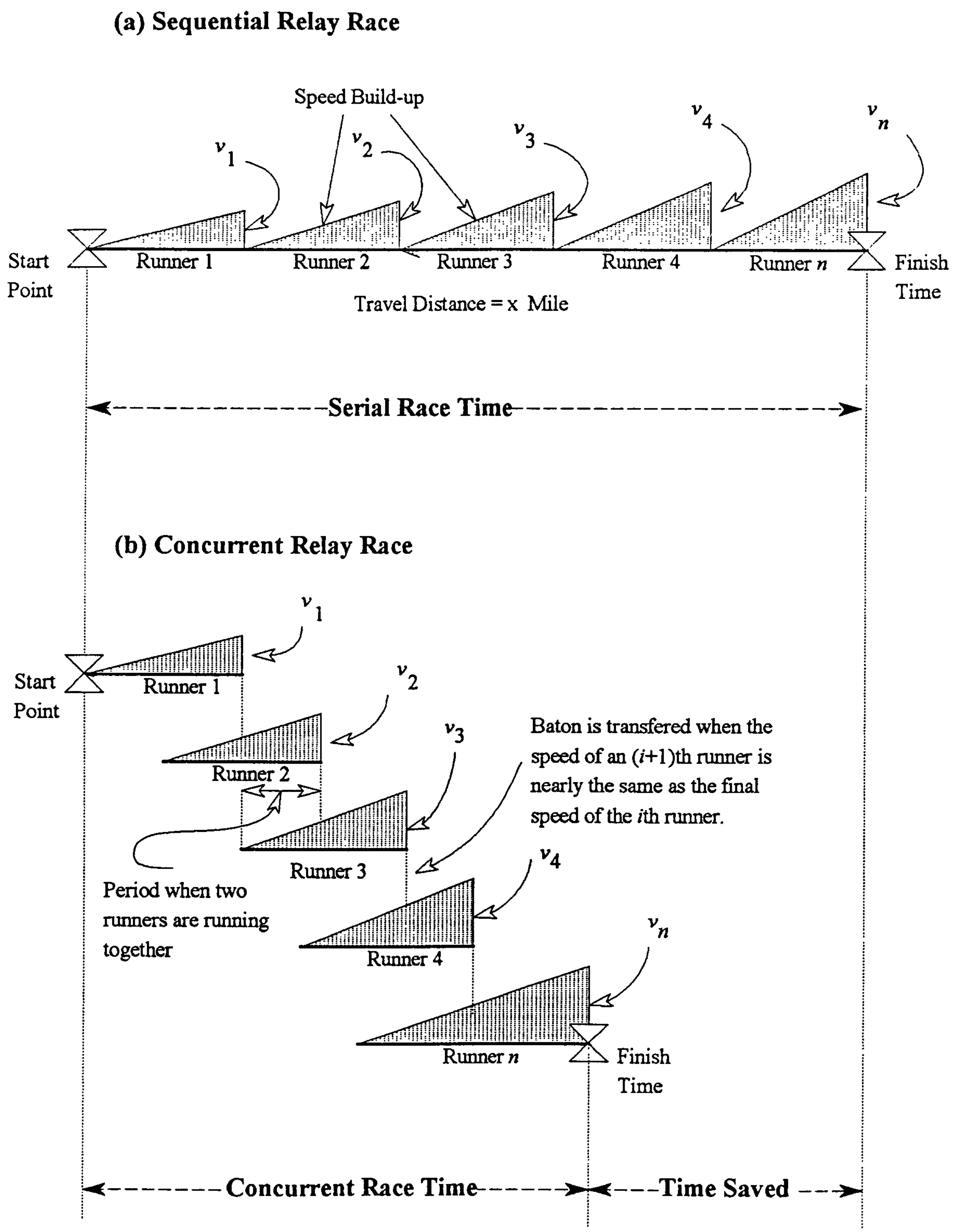

Figure 2. Analogy of CE to a relay race.

its peak from its initial stationary position $= v_i/2$ miles/hour; for $i = 1, n$. The time taken to finish the run can be computed as follows: time it takes for an ith runner to traverse (x/n) miles $= (x/n) * (2/v_i)$. Total time it takes for all the runners to finish the relay race will be the sum of individual time. If we call this type of run a serial-relay run, the total time T is

$$T(\text{serial-relay}) = \sum_{i=1}^{i=n} (x/n) * (2/v_i) \tag{1}$$

In the serial engineering process, the effort distribution is pretty much the same as the relay race, except the runner is replaced by a group of disciplinary experts (product development team), the distance to be traversed is replaced by the job to be finished, relay zones are replaced by the life-cycle phases for the product development, speed of the runner is replaced by the lead-time for completing a phase, milepost is equivalent to the milestone set by the management or design reviews at the completion of each phase, and completion of the run is replaced by the design and development of a product. The total life-cycle time is usually a multiple of the sum of each of its individual phases. This type of approach is also known by other names, including sequential engineering, linear or time-phase engineering, and the chimney method. If we denote this process as SE in short, then:

$$\text{Total life-cycle time (SE)} = [T_{requirements} + T_{product\text{-}def} + T_{process\text{-}def} + T_{del\text{-}and\text{-}sup}] * R_{factor} \tag{2}$$

where SE stands for Serial Engineering and $T_{requirements}$, $T_{product\text{-}def}$, $T_{process\text{-}def}$, and $T_{del\text{-}and\text{-}sup}$ represent the time it takes to finish a particular phase. R_{factor} is an equivalent repeat factor.

In SE, there are no consistent design, analysis, and documentation methodologies for conducting the business. Last minute engineering changes and higher product engineering costs result due to the lack of timely analysis of manufacturing considerations during early phases of product development. Preparation of data for analysis often takes too long, and by the time results are made available, the design often gets changed making the analysis useless. The result of such serial engineering has always been long design and build lead time, unnecessary complexity, and a scramble to obtain quality products. This shows in the high number of late-in-cycle engineering changes, design with excessive number of parts, retrofits, high customer complaints, manufacturing confusion, blaming mentality, high scrap and rework, increased field support and service, and only marginal customer appreciation or satisfaction compared to a competitor product. As a result, R_{factor} value for SE usually ranges above a factor of 2 or more.

$$R_{factor}\ (\text{for SE}) >>> 2.0 \tag{3}$$

In recent years, computers have been extensively used in the product cycle. Product development cycle is now frequently carried out with the aids of computer tools such as CAD, CAM, CAPP, CII, etc. These tools automate the discrete manual process of product design without affecting the inherent linear (sequential) nature of the process. Although it was thought that this would decrease the development cycle time and increase the communication between design and manufacturing, this has not occurred. This is because, by automating the functions within a phase, it only reduces the time required for that phase, it does not affect the manual or serial process of passing the enriched information between the phases. A significant time is lost in maintaining a serial nature of the process and performing the manual control of the phase interfaces or intraphase data integration [3]. This has created a proliferation of "islands" of departments due to the functional structure of the organization. Lack of management incentives for cooperation and inflexible culture of the organization to change have also impeded improvements in efficiencies and productivity. The impact of automation on the product development cycle has been, therefore, very little due to the unchanged inherent process [3].

The serial approach to product design, development, manufacturing, and marketing has several other shortcomings:

- It is based on the premise that a new phase cannot start until an old phase is completed and signed off. This usually means lengthening the product development cycle time.
- The linear input to product development implies that a significant portion (50% to 80%) of manufacturing cost may be committed before manufacturing engineers have a say in the product design.
- Due to the time loss, the final product may not remain suitable or viable for the market that was initially targeted at product launch.

3. Concurrent Engineering Analogy

In the 1990s, a new life-cycle management approach has emerged that focuses on "time" as critical force [2]. It analyzes the tasks' timing and duration across its entire organization with the goal of reducing tasks' time without any apparent loss in value to the finished product or service. This new approach focuses on tasks' time in all areas of product development that are value-added—from engineering to manufacturing, and from customer order to delivery. This management process is commonly referred to as Concurrent Engineering (CE). In CE, the tasks and phases run in parallel with feedback as and when needed, as shown in Figure 1. Each area of product development cycle has its own life-span that overlays the entire process. This series of smaller life-spans are shown in Figure 1 using "blinders." CE has brought these blinders closer together so they now overlap.

CE is analogous to a variation of the "serial-relay race" described earlier (see Figure 2). It is not essential that each runner be placed at an equal distance (e.g., x/n miles) apart. Second, it is not essential that a runner has to start only after the previous runner has completed the x/n miles run. A runner can start earlier from his milepost, enough for him to catch up the speed and become equal to or greater than the speed of his partner, when the time comes to transfer the baton. The first runner stops when the second runner has achieved the first runner's speed. This way, the speed throughout the race remains fairly constant. If v_i is the maximum speed of each runner, time T for the serial-relay and concurrent-relay can be computed as follows:

$$\text{The average speed for the race} = \left[\sum_{i=1}^{i=n} v_i\right] \Big/ n \qquad (4)$$

The time the runner takes to traverse a total of x miles, would be:

$$T(\text{concurrent-relay}) = \frac{(x * n)}{(\Sigma v_i)} \text{ unit of time} \qquad (5)$$

If $v_i = v_o$, that is the same for all i's, $i = 1, n$; then it is not difficult to show [comparing Equations (1) and (5)] that concurrent-relay race would be 50% more efficient than the serial-relay race. If v_i is a linearly increasing speed from 1 to n miles/hour, that is,

$$v_1 = 1; \quad v_2 = 2; \quad v_3 = 3; \ldots \text{and } v_n = n \qquad (6)$$

then from Equation (1)

$$T_{serial\text{-}relay} = \left[(2x/n) * \sum_{i=1}^{i=n} (1/i)\right] \qquad (7)$$

and from Equation (5)

$$T_{concurrent\text{-}relay} = [2x/(n + 1)] \qquad (8)$$

It may be noted that the Equation (7) for the total time in the serial-relay case is divergent for a large value of n. However, in the concurrent-relay case, Equation (8) is still a finite sum. Similar to the concurrent-relay race case, in the CE case, it is assumed that the concurrent work-groups can start working on their portion of the work much before the previous work-group is finished. While the second work-group comes to a speed for understanding what the first group has done so far, real useful information can start flowing between the groups at frequent intervals to make both the units more productive. Instead of a one-time transfer of the baton in the relay race, the transfer between the consecutive work-groups is continuous and teamwork-like. The previous work-group stops their task, when no more input from the first work-group is required. The process continues when all the work-groups have made both their individual and team contributions. Individual contribution is when the teams are working alone (when there is no overlap) and team contribution is when they are working jointly as a team for a portion of the tasks which are overlapped. The passing of the baton ceases when an artifact is finally manufactured and delivered to the customer. A comparison of steps involved in serial- and concurrent-relay race is summarized in Table 1.

It is usually necessary to use several computer-assisted programming tools to support the CE process. These tools should be capable of communicating with each other rather than creating isolated "islands of automation." Not surprisingly, the cumulative time it takes to complete all the phases and tasks is much smaller than that of "Sequential Engineering." R_{factor} usually ranges between 0.25 to 0.75

$$R_{factor} \text{ (for CE)} \cong 1/2 \qquad (9)$$

The word "Engineering" in CE is used in a generalized sense. It is meant to include involvement of personnel from all required disciplines: engineers or nonengineers. The total process, from product development to mass production and sales, involves many nonengineers, including persons from the procurement, marketing, and sales departments. Hence, in Concurrent Engineering, all major parties involved in getting the product to market contribute to the development of the product. In contrast to the traditional sequential, linear, iterative (uncontrolled), and functional departmental practices, the Concurrent Engineering approach requires a parallel interactive (managed), and cooperative multidisciplinary team approach to product and process development.

4. Concluding Remarks

Just as craft manufacturing leveraged people's skill, muscles, and dexterity (economy of skill), Concurrent Engineering is about leveraging teamwork (economy of cooperation) to handle information and make informed decisions. Instead of using a human strength (physical power and muscles), CE is more directed toward utilizing the teams' intellectual power. These teams are not just the select employees, but all the parties that are involved in the product realization process.

Table 1. A comparison table for serial-relay and concurrent-relay race.

Subjects	Serial-Relay	Concurrent-Relay	Concurrent Engineering
Who are the runners?	Individual runners.	A team of runners or partners.	Concurrent teams.
Distance to be traversed	Mileposts are placed equidistantly apart.	Distance traveled by each runner is not necessarily the same.	Tasks or jobs to be done are not necessarily independent of each other.
Relay-Zones	Every runner starts from a stationary position. The mileposts are placed at the start points. The end point of a previous runner and start point for a next runner is coincident.	The start point may be placed earlier from the milepost. This allows the second runner time to build up speed before the baton is turned over. It means there are overlapped running zones.	The phases overlap. Two tasks are run simultaneously until the point when information build-up is adequate for the next task to proceed on its own.
Speeds	Every runner starts from his/her stationary position (zero speed) at each milepost.	Every runner starts from his/her stationary position, but much before the finish point of a previous runner.	Information is built up to a point that can be useful for the early completion of the next task.
Transfer-Points	When a runner reaches the designated milepost.	When the speed of the new runner equals or exceeds the previous runner's speed.	The information build up is sufficient to allow the next task to proceed independently.
Milepost/Objectives	Minimize the time for traversing the distance between two consecutive mileposts.	Minimize the total time for traversing the whole run.	Minimize the time it takes to do the entire job, not the completion of individual tasks.
Completion Points	When the baton is finally passed to the last person in the race.	When the baton reaches the last post (how it gets there is immaterial).	When the whole job is successfully done (irrespective of order in which it is executed).
Results/Outcome	All runners contribute to the best of their abilities.	Best of the individual and their combined team abilities/efforts.	Collaborative efforts.

References

1. Prasad, B. 1995. *Concurrent Engineering: Fundamentals, Volumes 1-2*. Englewood Cliffs, NJ: Prentice Hall.
2. Prasad, B. 1995. "On Influencing Agents of CE," *Concurrent Engineering: Research & Applications—An International Journal*, 3(2):78-80.
3. Prasad, B. 1995. "Is Planning a Strategic Requirement for CE Success?" *Concurrent Engineering: Research & Applications—An International Journal*, 3(3):164-166.