THE ROLE OF UNCERTAINTY REDUCTION IN CONCURRENT ENGINEERING: AN ANALYTICAL MODEL AND AN EMPIRICAL TEST

by

C. TERWIESCH *
and
C. H. LOCH **

96/17/TM

* PhD Candidate, at INSEAD, Boulevard de Constance, Fontainebleau 77305 Cedex, France.

** Assistant Professor of Operations Management, at INSEAD, Boulevard de Constance, Fontainebleau 77305 Cedex, France.

Printed at INSEAD, Fontainebleau, France.

The Role of Uncertainty Reduction in Concurrent Engineering: An Analytical Model and an Empirical Test

Christian Terwiesch and Christoph H. Loch

INSEAD

March 1996

Abstract

Concurrent engineering, or overlapping development activities, is a widely discussed tool to reduce development leadtime. However, overlapping may not be beneficial if the overlapped activities are uncertain and not independent. This article combines recent results from the organization theory and concurrent engineering literature to develop a model of two overlapping development activities. Uncertainty is incorporated via the average rate of engineering changes that occur in the upstream activity, creating rework for downstream. Dependence is modeled by the delaying impact these modifications impose on the downstream activity. The model predicts that the effectiveness of overlapping activities in reducing overall development time is moderated by the organization's capability to reduce uncertainty early upstream. Further, fast uncertainty reduction makes it optimal to use more overlap. These effects of uncertainty reduction are tested on data drawn from 140 completed development projects across several global electronics industries.

1 Introduction

With intensifying competition in many industries, time to market has been emerging as an important factor for success in the 1990s (e.g., Blackburn 1991). Many tools have been proposed to accelerate the product development process, among those the concept of concurrent engineering, or overlapping of activities. The success story of this technique has been described in numerous articles. However, existing research has not sufficiently addressed the impact of the technical and organizational context on the effectiveness of overlapping activities. In particular, previous research exhibits a strong bias towards stable and mature environments (Iansiti 1995). This leads one to ask under what environmental conditions concurrency has an accelerating effect, and when it may not be appropriate.

The absence of a product development theory valid across industries is observed by Brown and Eisenhardt 1995, who explicitly emphasize differences in the development process between mature products (e.g., cars and mainframes) and high uncertainty products (e.g., microcomputers). Such differences across industries are supported by several other studies (e.g., Loch *et al.* 1996, Cordero 1991). Given the potential benefits of concurrent engineering and its wide use in practice, these shortcomings in our current understanding of this technique are relevant from a practical as well as from an academic perspective.

The contribution of this article is twofold. First, we present an analytical model that incorporates contextual variables such as the uncertainty of the development process and the speed of uncertainty reduction. The model allows us to derive the optimal level of concurrency and to compare its effectiveness in different environments. We also conduct a sensitivity analysis on the optimal concurrency value across different levels of the contextual variables. Second, we test the main insights derived from our model on a sample of 140 completed development projects across several global electronics industries. Thus, for the first time, we empirically test a mathematical

model of concurrent engineering on a large sample.

The article is organized as follows. Section 2 reviews the relevant literature on concurrent engineering. In Section 3, we present our mathematical model and derive the optimal level of concurrency as a function of the contextual variables. The main result of the model is tested in Section 4. The article ends with a discussion of our results and a preview on future research.

2 Literature Background

Given the importance of concurrent engineering for reducing development times, the subject has drawn substantial attention across disciplines. On one side, there is an extensive body of literature on managing the organizational aspects of overlapping development activities. On the other side, there is a rapidly growing number of articles using mathematical models to analyze concurrent engineering from an industrial engineering or operations management perspective. Both approaches are influenced by the landmark study of Clark and Fujimoto 1991, but show substantial differences in underlying theory and methodology. We will now briefly review these two perspectives.

The organizational perspective is grounded on data, ranging from case studies to large scale questionnaires. Imai *et al.* 1985 and Takeuchi and Nonaka 1986 were the first to report in what way faster development processes can be achieved by overlapping activities. They also coined the metaphors of "relay race" (one specialist passes the baton to the next, also referred to as the "over-the-wall mode") and "rugby team" (a cross functional team on the project).

In their famous study of product development practices in the world automotive industry, Clark and Fujimoto 1991 showed that overlapping activities accelerated the product development process. With their construct "overlap ratio", they were the

first to operationalize concurrent engineering and to identify a significant accelerating effect on project duration. In addition, Clark and Fujimoto examined the organizational context, in which overlapping activities is beneficial. Using an information processing framework (Galbraith 1973, Tushman and Nadler 1978), they identified intensive communication as a key success factor. These ideas were refined in further studies, including those by Wheelright and Clark 1992 and Clark and Wheelright 1994.

Most studies so far have been conducted in the automotive industry, thus exhibiting a strong bias towards complex and mature industries (Iansiti 1995). In their study of the world computer industries, Eisenhardt and Tabrizi 1995 identify substantial differences across different market segments and are thus able to address this "maturity bias". For the stable and mature segments of mainframes and microcomputers, the authors find that overlapping development activities significantly reduces time-to-market. However, in rapidly changing markets such as printers and personal computers ("high velocity environments" in the words of Eisenhardt 1989), overlapping is no longer found to be a significant accelerator. Eisenhardt and Tabrizi argue that compressing the development process only yields a time reduction if the underlying process is repetitive and predictable.

Adler 1995 uses an information processing framework to analyze the design / manufacturing interface. The problem of ensuring the fit between product and process development reflects the concept of two interdependent departments. As the tasks of designing a product and developing a suitable manufacturing process are not fully predictable and performed by different organizational units, uncertainty results. Although this study does not focus on concurrent engineering, it provides two interesting dimensions of uncertainty that we will use as a theoretical basis for our model. Fit novelty is defined as the number of exceptions with respect to the organization's existing experience base in product/process fit problems. From a nor-

mative viewpoint, greater fit novelty requires a higher information flow between the groups responsible for product and process design. Fit analyzability is the difficulty of the search for an acceptable solution for a given problem. These problems can be more or less numerous, according to the fit novelty measure. Fit analyzability indicates whether a problem takes more or less time to be resolved. If a problem is not solved during the current phase, it delays the continuation of the development process downstream.

The use of mathematical models to gain understanding of some aspects of concurrent engineering is very recent. Krishnan *et al.* 1994 develop a framework to support the overlap decision between two development activities. The framework introduces the concepts of upstream evolution and downstream sensitivity on which the model in this article builds. Upstream evolution is defined as the reliability of preliminary information released by the upstream activity. If one takes the creation of a design parameter (e.g., an axle diameter) as an example for the upstream output, one can plot the set of possible outcomes as depicted in Figure 1. Initially, the interval for the parameter is wide, then narrows down over time and finally converges to the outcome parameter. The speed of this convergence can be defined as $\varepsilon(t)$ and is called the evolution function. Thus, from an organizational design perspective, fast evolution represents early uncertainty reduction. Downstream sensitivity is a measure of dependence and describes how severely deviations from the preliminary information (created upstream) delay the downstream activity. Dichotomizing both evolution and sensitivity into two values "high" and "low", Krishnan *et al.* 1994 recommend different forms of concurrency for four cases, conjecturing that faster evolution and less sensitivity allow more concurrency.

Figure 1 about here

The inherent limits to concurrency are described in a model by Hoedemaker *et al.* (1995). The authors develop a simple model of one phase in a development process

being decomposed into parallel submodules. They then derive, that due to communication costs, rework probabilities, and integration time, the optimal level of decomposition (concurrency) should not be pushed too far. This result emphasizes that concurrency is not free and does impose a tradeoff. However, it only looks at concurrency within a development activity, rather than at the overlap between subsequent activities.

Ha and Porteus (1995) analyze a product design for manufacturability project. They develop the "how frequent to meet" problem as a dynamic program. Concurrency offers advantages from parallel development and from an early discovery of flawed product design. Similar to a quality inspection problem in production, these gains have to be traded off with time penalties for cross functional meetings. The Ha and Porteus model presents a different situation from the one in the present article: the two tasks in their model are inherently interdependent and must be carried out in parallel, because severe quality problems result otherwise. The key question is how often to meet and update ("how far to let one activity run ahead"). In our model, we describe a true up- and downstream situation, where a sequential process is "natural", and overlap results in errors from using premature information.

Both the organizational and the operations management perspectives described above offer specific insights. The advantage of the managerial perspective is a strong theoretical grounding in organization theory with first steps towards an empirical validation. However, Adler 1995 points out that existing organizational models are weak in explaining time phenomena during development projects. Along the same line, Eisenhardt and Tabrizi 1995 underline the need for a more dynamic perspective of organizations, to truly understand the product development process. In this limitation, organizational models can be complemented by the operations perspective which has a long tradition of time-based modeling.

Summarizing the contributions of both research streams, we can postulate three de-

sirable properties for a model of concurrent engineering. First, a model should be linked to existing theory or be grounded in inductive research. Second, it should explicitly include the dynamics of the process, allowing a trade-off between overlap gains and potential drawbacks. Finally, the model should produce testable hypotheses, allowing an empirical validation of its predictions.

3 The Model

Starting theory-building with a mathematical model has strengths and weaknesses. The major weakness is that the resulting theory is not grounded on data. This weakness can be overcome in two ways. First, one can test assumptions of the model by comparing them with the corresponding constructs in reality. Second, in line with the paradigm of positive economics, the model can be tested based on its predictions. In this article, we follow the latter path, and we perform the former, at least partially, by linking the assumptions and constructs of our model to the organizational and operations management literature.

The major strength of mathematical model-building and simulation is the perfectly controlled environment. The researcher can, at almost no cost, compare different contexts for which extensive cross-sectional and/or longitudinal data collection would be required in an empirical study. No measurement error or sample size constrains the analysis. In addition, model building has further advantages over simulation. First, results can be directly obtained by using calculus, without resorting to statistical inference. Second, it is possible to gain insights on the relative importance of different variables by conducting sensitivity analysis. Finally, the model and its assumptions are fully transparent to the reader and can thus be the subject of scientific debate.

To capitalize on these strengths, we start our theory-building with a mathematical model. We will now present the model set-up, followed by the definition of our key

constructs. We then derive the level of optimal concurrency including a sensitivity analysis on environmental variables. The section ends by deriving a testable hypothesis about the impact of concurrency on project completion time, across different speeds of uncertainty reduction.

3.1 Model Set-Up

Consider the timing of a project with two phases of length t_1 and t_2. In the product development process, t_1 could be the time of product design and t_2 the time of process design. Similarly, in software development, the first phase could be specification development and the second coding. We call the first phase upstream activity and the second downstream activity. In the "over the wall" approach, the total completion time is $t_1 + t_2$. Now, some proportion λ of downstream can be done in parallel to t_1. If t_2 is larger than t_1, no more than $\frac{t_1}{t_2}$ can be parallelized. Let

$$\Lambda \;:=\; [0, min\{1, \frac{t_1}{t_2}\}] := [0, \lambda^{max}] \tag{1}$$

be the interval of all possible levels of overlapping. λ provides a continuous measure of concurrency. The project completion time T can be written as a function of the concurrency level λ (see Figure 2):

$$T \;=\; t_1 + (1 - \lambda)t_2 \tag{2}$$

However, concurrency has a drawback. As the upstream activity is not fully predictable, new information will arise in the course of upstream problem solving, which cannot be anticipated by downstream. This unpredictable information (e.g., new ideas, modifications) create rework for downstream, because work already completed has to be adapted to the upstream changes. The rework lengthens the overall completion time T, creating a trade-off between starting downstream early and the rework from upstream modifications.

This trade-off is now explored in more detail.

Figure 2 about here

3.2 Creation of Rework, Downstream Sensitivity

In our model, we assume sequential dependence as defined by Thompson 1967. Thus, the workflow only goes in one direction (from upstream to downstream). Preliminary information can be passed from upstream to downstream, but downstream must include the final upstream information result, even if it differs from the preliminary information transmitted earlier. In the product development context, this means the product is to be manufactured the way product design wants it to be. Process engineers have to include all modifications, even if they are communicated after $t_1 - \lambda t_2$ (start of process design). In software development, we find such a dependence between the specification and the coding of a system. This approach to concurrent engineering is also known as the "Flying Start Concept" (Blackburn *et al.* 1994).

Sequential dependence has been chosen instead of interdependence as it is closer to the compression approach described in Eisenhardt and Tabrizi 1995: the gain of overlap is to "squeeze" the development process together. If development tasks are inherently interdependent, concurrency (in the form of joint problem-solving) is natural, and already reported in Van de Ven *et al.* 1976. Ha and Porteus 1995 offer a mathematical representation of project management with interdependent tasks.

We model the creation of upstream modifications by a stochastic process with mean $\mu(t)_{t=0..t_1}$. The downstream rework caused by each modification depends on how much work has already been done. We define the impact function $f(t)$ as the time it takes to change previous downstream work, if one modification occurs at $t \in [0, t_2]$ units of downstream time. This approach is based on the concept of downstream sensitivity, developed by Krishnan *et al.* (1994). Changes in the preliminary information received so far will delay the downstream activity. The more downstream has progressed in its work, the more cumulative work has to be modified, and the greater

the impact of an upstream modification becomes. Hence, $f(t)$ is non-decreasing.

Observe the relationship between the model assumptions and Adler's definitions of fit novelty and fit analyzability (Adler 1995). Usually, upstream passes preliminary information to downstream based on its existing experience base. In our model, μ is the occurrence rate of deviations from this preliminary information, and therefore maps the definition of fit novelty. Downstream sensitivity, as modeled by the impact function $f(t)$, captures the idea of fit analyzability, which is defined by Adler 1995 as the time it takes to resolve a *given* fit problem. Including an upstream modification in the beginning of the downstream activity is a rather analyzable task. The modules relevant to the modification can be easily identified and then changed accordingly (low value of $f(t)$). These changes become less analyzable with a growing complexity of the partial product or system already completed.

We abstract from communication problems, assuming that all modifications are immediately transferred. The role of information exchange and cross functional meetings is analyzed in Loch and Terwiesch 1996. We focus on linear impact functions: $f(t) = kt$, where k is a constant denoting the slope of the function. In practice, the downstream progress may accelerate over time (e.g., due to learning effects). Similarly, $f(t)$ may grow faster than linearly if modifications not only require rework, but make large parts of achieved progress obsolete. The linearity assumption is a simplification that allows us to obtain closed form solutions in the mathematical model. Our qualitative results, however, are robust with respect to this computational assumption, and can be shown to be valid for any non-decreasing impact function $f(t)$.

3.3 Evolution and Uncertainty Reduction

The rate of change generation, $\mu(t)$, is determined by the progress evolution of the upstream task. If upstream uncertainty is resolved quickly (the outlines of a design

are stabilized early), $\mu(t)$ is high at first and then drops as stability is reached. If, in contrast, convergence to a design in upstream occurs late, $\mu(t)$ is low initially and then rises as the design concept evolves at the end. We assume that the stochastic process generating modifications in the upstream task is a non-homogenous Poisson process with rate $\mu(t)$. The Poisson process is a realistic and frequently used model of several individuals working in parallel on the same problem and all being independent potential sources of modifications. The non-homogenous Poisson process is a mathematically tractable model that allows the inclusion of the dynamics of uncertainty reduction (via the time dependent μ). We model $\mu(t)$ as a *linear* function:

$$\mu(t) = \mu_0[1 + e(2\frac{t}{t_1} - 1)]. \tag{3}$$

The parameter $e \in [-1, 1]$ is called the *uncertainty reduction parameter.* When $e > 0$, then the arrival rate of modifications $\mu(t)$ is increasing over time, corresponding to slow uncertainty reduction. When $e < 0$, $\mu(t)$ is decreasing, corresponding to fast uncertainty reduction. When $e = 0$, modifications are generated as a stationary process ($\mu(t) = \mu_0$). In all three cases, the total expected number of changes generated over the time of task 1 is the same, namely $\mu_0 t_1$. Thus, e represents the evolution of changes (uncertainty reduction), and μ_0 represents the total number of changes, or the overall level of uncertainty. Figure 3 relates the uncertainty reduction parameter e and the mean arrival rate of modifications $\mu(t)$ to the evolution function by Krishnan *et al.* 1995.

Figure 3 about here

3.4 Optimal Concurrency

We have now defined all the elements of the model and can state the time-to-market optimization problem:

$$\min_{\lambda \in \Lambda} \quad t_1 + (1-\lambda)t_2 + E[R(\lambda)] \tag{4}$$

To obtain the optimal level of concurrency, we first have to find a closed form expression for the expected amount of rework $E[R(\lambda)]$. The influence of concurrency on rework is influenced simultaneously by evolution and sensitivity.

Intermediate Result 1: The expected amount of rework $E[R(\lambda)]$ given the concurrency level λ can be written as:

$$E[R(\lambda)] = \int_0^{\lambda t_2} kt\mu(t_1 - \lambda t_2 + t)dt \tag{5}$$

$$= \frac{1}{6}\frac{k\mu_0(\lambda t_2)^2}{t_1}(3t_1 + 3et_1 - 2e\lambda t_2) \tag{6}$$

Proof: To prove (5), we use the fact that the arrival of modifications follows a non-homogenous Poisson process with rate $\mu(t)$. Observe first that when an event (modification) occurs at time t' units of upstream time, downstream has progressed $t' - t_1 + \lambda t_2$. Thus an idea occurring at t' has impact of $k(t' - t_1 + \lambda t_2)$. The expected rework at time t' for the interval $[t', t' + \Delta t']$ is

$$E[R_{t'}] = k(t' - t_1 + \lambda t_2)[\mu(t')\Delta t' + o(\Delta t')]$$

If we shrink the interval length $\Delta t'$ to an infinitesimal dt', by definition of the Poisson process $o(\Delta t') = 0$. We can then obtain the total expected rework by integrating over time:

$$E[R(\lambda)] = \int_{t_1 - \lambda t_2}^{t_1} k(t' - t_1 + \lambda t_2)\mu(t')dt'$$

Shifting upstream time t' to the downstream time $t' = t_1 - \lambda t_2 + t$, thus $dt' = dt$, we obtain

$$E[R(\lambda)] = \int_0^{\lambda t_2} kt\mu(t_1 - \lambda t_2 + t)dt$$

which completes (5). To prove (6), we substitute (3) for $\mu(t)$, which gives us a polynomial expression in t within the integral:

$$E[R(\lambda)] = \int_0^{\lambda t_2} kt\mu_0[1 + e(2(\frac{t_1 - \lambda t_2 + t}{t_1} - 1)]dt$$

Integrating the polynomial within the given boundaries yields the desired result. □

Figure 4 about here

Observe that the concurrency level λ has two effects within (6). First, there is a positive quadratic effect which results from compounding a longer concurrency period over which modifications occur, and the growing impact per modification. Second, there is a cubic effect which is negative if uncertainty reduction is slow ($e > 0$): more overlap includes the early upstream time periods with relatively few modifications. If $e < 0$, this effect is positive, because more overlap includes earlier upstream time with more modifications.

This trade-off between activity overlap, impact, and evolution is summarized in Figure 4. Depending on upstream progress, modifications are created at rate $\mu(t^{up})$ (here time is measured as upstream time). A modification at $t_{occurrence}$ delays downstream, depending on how far downstream has advanced in their time t^{down}. Finally, rework is determined via the impact function $f(t_{occurrence})$ (measured downstream). As an illustration, Figure 4 describes the case of only one modification occurring.

With equation (6) we rewrite the overall objective function as:

$$E[T(\lambda)] = t_1 + (1 - \lambda)t_2 + \frac{1}{6}\frac{k\mu_0(\lambda t_2)^2}{t_1}(3t_1 + 3et_1 - 2e\lambda t_2) \tag{7}$$

We show in the appendix (intermediate result 2) that the objective function is convex in λ, so that the first order condition

$$\frac{\partial E[T]}{\partial \lambda} = -t_2 - \frac{k\mu_0 t_2^2 \lambda(e\lambda t_2 - t_1 - et_1)}{t_1} = 0 \tag{8}$$

uniquely characterizes the optimal concurrency level if its solution is within Λ.

As our primary research objective is to investigate the contextual influence of uncertainty reduction (evolution) on the effectiveness of overlapping activities, we will now analyze how expected project completion time ET and optimal concurrency λ^* depend on e. Theorem 1 will show how the time reduction effect of overlap grows with the organization's capability to quickly resolve uncertainty.

THEOREM 1: *For any overlap of activities λ, a change in λ yields a higher reduction of project completion time in cases of fast uncertainty reduction (small e) than for slow uncertainty reduction (large e).*

Proof: We have to show that $\frac{\partial^2 E[T]}{\partial e \partial \lambda} > 0$. Computing the cross-partial derivative yields

$$\begin{aligned} \frac{\partial^2 E[T]}{\partial e \partial \lambda} &= -\frac{k\mu_0 t_2^3 \lambda^2}{t_1} + k\mu_0 t_2^2 \lambda > -\frac{k\mu_0 t_2^3 \lambda}{t_1}\frac{t_1}{t_2} + k\mu_0 t_2^2 \lambda \\ &= -k\mu_0 t_2^2 \lambda + k\mu_0 t_2^2 \lambda = 0 \end{aligned} \tag{9}$$

as $-\lambda > -\frac{t_1}{t_2}$. Thus the time reduction effect $-\frac{\partial T}{\partial \lambda}$ is decreasing in e, which completes the proof. □

The intuition for this result is that slow uncertainty reduction causes modifications to occur late, which makes them costly because a great deal of work has to be modified. Thus the rework "cost" of overlap is higher than if uncertainty can be reduced quickly. In other words, benefits of concurrency are smaller in the case of slow uncertainty reduction. Following up on our review of the literature, this result offers an explanation, as to why concurrency "works" in some cases but not in others. In addition to comparing the time reduction effects, one can analyze how the optimal concurrency changes across different levels of the contextual variables, thus conducting a sensitivity analysis on λ^* depending on e, k, μ_0. This is our second analytical result; the proof is given in the appendix.

THEOREM 2: *Slower uncertainty reduction (increasing e), higher impact k, and a higher rate of modifications μ_0 all separately decrease the optimal degree of overlap.*

3.5 Development of Hypothesis

Theorem 2 provides a normative result and can easily be transformed into a testable hypothesis. Projects that reduce uncertainty early in the development process will face fewer modifications toward the end of the upstream task, and hence should use more overlap. If we assume that all projects use the optimal level of overlap (fulfill their first order condition as derived in our model), we can hypothesize:

HYPOTHESIS: *Projects with fast uncertainty reduction use more concurrency than projects with slow uncertainty reduction.*

To test this hypothesis, we divided our sample into two subgroups, below and above the median value of the uncertainty reduction measure. Support of the hypothesis would require a significant difference in overlap across these two subsamples. However, comparing the mean overlaps across the two subgroups did not show any significant difference (see next section for details on measurement and data collection). Thus, Theorem 2 was not supported by our data.

In the derivation of Theorem 2, we are assuming that all projects minimize time-to-market by using the optimal overlap. This is a rather strong assumption, and one which we now want to relax for further analysis.

In contrast to Theorem 2, Theorem 1 is valid for any overlap of activities λ, not just for λ^*. It proposes that the time reduction from overlapping will be larger in cases of fast uncertainty reduction (small e) than for slow uncertainty reduction. This situation is illustrated in Figure 5.

Figure 5 about here

Consider two levels of uncertainty reduction $e^{high} > e^{low}$. In both cases completion time is $T = t_1 + t_2$, if no overlap is used. We know that $E[T(\lambda)]$ is convex in both cases. The maximum potential time reduction from concurrency is $t_1 + t_2 - ET[\lambda^*]$. Theorem 1 predicts that this maximum benefit (the difference between $t_1 + t_2$ and

the lowest point of the curve) is smaller in the case of slow uncertainty reduction (large e) than for fast uncertainty reduction. This statement holds for any λ whereas Theorem 2 only considers the *optimal* concurrency level. We use Theorem 1 together with the missing support of our initial hypothesis to form an *a posteriori hypothesis* that explicitly considers projects that have sub-optimal concurrency:

HYPOTHESIS: *In projects with fast uncertainty reduction, the benefits of concurrency are larger than in projects with slow uncertainty reduction.*

This hypothesis suggests a moderation (or contingency) perspective to uncertainty reduction: the time reduction impact of concurrency is moderated by uncertainty reduction. Arnold 1982 and Venkatraman 1989 both underline the need for a correspondence between theoretical modeling and statistical testing of moderation effects. They distinguish between the degree (strength) of the moderation and its form. Hypotheses about the degree of moderation state that the amount of variance accounted for by the independent variable differs depending on the level of the moderator. The appropriate statistical test for this is comparing correlations between the dependent and the independent variable across different levels of the moderator. Our new hypothesis is concerned with the *form* of the moderation effect, "Does a change in the independent variable always yield the same amount of change in the dependent variable?" The corresponding statistical test for this case is comparing regression coefficients across different values of the moderating variable (Arnold 1982, Venkatraman 1989). Given our hypothesis, we use multiple regression analysis as our primary statistical tool.

Our statistical analysis only considers the effect of uncertainty reduction, because we do not have data on k and μ_0. The investigation of their impact, which our model suggests to be similar to the impact of e, must be left to future research.

4 Empirical Test

Our analysis is based on a sample of 102 electronics companies in the US, Japan, and Europe. During 1992-1993, these companies completed detailed questionnaires on operations and strategy as part of the "Excellence in Electronics" project jointly undertaken by Stanford University, the University of Augsburg and McKinsey & Company. Parts of the sample have already been used for other research projects (e.g., Eisenhardt and Tabrizi 1995, Terwiesch *et al.* 1996). Many leading companies agreed to participate in the survey, providing us with data on 12 of the 25 leading computer producers and four of the six biggest TV manufacturers, to cite two industry examples. The unit of analysis of our work is the individual development project. Each participating business unit contributed two new product development projects.

4.1 Data Collection

Our analysis of product development in the electronics industry is only one part of a larger data collection effort. In addition to product development, the overall instrument contained questions on marketing, manufacturing, finance and top management. The product development part consisted of a group of general questions concerning product development practices of the business unit, and a subsection for each of two specific projects. These were used for the research presented in this article. To avoid biases coming from hindsight reasoning and retrospective sensemaking, we focused on technical questions with closed form answers.

We organized the 204 projects into 14 product groups such as TV, mainframe, PC, printer, etc. This grouping allowed us to compare similar development projects with one another and to standardize certain measures within a product subsample (see below). Some of the projects were small, peripheral modifications involving only

one or two engineers. Since our research focus is on product development projects, we decided to omit all projects with less than five person years from our statistical analysis. Two other projects were excluded because their technical content was unique in the sample, prohibiting benchmarking with others. The remaining sample included 140 observations.

4.2 Measures

The duration of a development project is not only influenced by overlap: previous research identified and confirmed the importance of several other predicting variables (e.g., Eisenhardt and Tabrizi 1995, Terwiesch *et al.* 1996). We include these predictors in our regression analysis for two reasons. First, leaving out variables which influence the dependent variable (project duration) can potentially create biases. Second, in addition to the hypothesized moderating effect on overlap, uncertainty reduction could also have similar effects on these other variables. The additional effects of uncertainty reduction are thus interesting by-products of our statistical analysis.

Since our research focus is on development time, we used the *standardized project duration* as our dependent variable. Project duration was defined as the time from the first project meeting until product stabilization was achieved. The standardization was performed by taking the difference between the project duration and its industry subsample average, divided by the industry subsample average. That is, a project of average length in its product group was given the measure zero.

Although projects within a subsample are homogenous concerning the developed product, they still can differ in their technical content. In this article we are not interested in this *size effect*, but in the effect of different project management decisions. Since it is reasonable to assume that larger projects take longer than small ones, we controlled for this size effect by including a control variable in the regres-

sions. Size was measured by project effort (in person years) and standardized as previously described.

In the questionnaire, a development project was structured into six phases: pre-development study (to completion of basic product requirements), conceptual design (to specification of all product functions), product design and engineering (to system testing release), system testing (to production release), final process development and scale-up (through completion of pilot production run) and production start-up (to stabilization). We measured *overlap* as the sum of the overlaps between subsequent phases divided by the gross duration of the project without deducting overlap (i.e. the sum of the development phases). The higher this ratio, the more overlap was used in the project. Similarly, we defined *testing* as the ratio of the testing phase duration and the sum of the other phase durations, and we measured *use of preliminary information* by the residual project time when first preliminary specifications were written.

Time between milestones was measured by the average number of weeks between two officially scheduled project reviews. Only milestones with a detailed project review were included. We measured the number of design *iterations* by asking how many redesign iterations the product took before stabilization. A redesign iteration was defined as a modification of more than 10% of product components. Prototyping is a typical example of such a type of iteration, whereas debugging does not classify as an iteration. As products in the electronics industry are significantly influenced by their software, we used as our measure the larger of the number of hardware iterations and software iterations. Finally, we included the *redesign intervals* (measured in months) of products in the focal business unit. This was the only measure that was taken from the non-project specific part of the questionnaire. As the magnitude of these items might substantially differ across product subsamples, all three were standardized in the same way as project duration.

Figure 6 about here

Whilst the previous measures could be directly derived from the questionnaire, uncertainty reduction had to be constructed by combining different items. An operationalization of the residual uncertainty over the project duration is not straightforward. As a proxy, we used the three milestones "preliminary information release", "detailed specification defined" and "specifications frozen". These are well-known industry terms, which we link with relative phase durations to create an uncertainty curve. As an illustration, consider two projects. Project A (left in Figure 6), reaches the level of preliminary information after 10% of project time, detailed information after 30% and freezes the specifications after 50%. Project B releases first information after 20%, detailed after 50% and freezes at 90% of the total project time. This provides us with the desired uncertainty curve. As a measure of uncertainty reduction, we used the area of the shaded rectangles in Figure 6 . Uncertainty reduction is faster for Project A which has the higher shaded rectangle area.

The reader might observe that the measurement of overlap and uncertainty reduction is extended over the course of the whole project (not just two tasks). Our mathematical model could easily be extended to include multiple tasks. However, such an extension would substantially complicate its analysis without creating additional insights. Both hypotheses would remain unchanged. Further, considering the whole project allows us to benefit from established measures (i.e., the overlap ratio provided by Clark and Fujimoto 1991) and to make our findings consistent with previous studies (e.g., Eisenhardt and Tabrizi 1995).

Table 1 about here

4.3 Regression Results

Model 1 shows the control effect of project size on the dependent variable. However, only 23% of the variance in project duration can be explained by size. The second regression model adds the variables that we expected would influence project duration. The estimated coefficients, model fit and significance are also reported in Table 1. Thus, there is a significant overlap benefit across levels of the contextual variable "uncertainty reduction". More overlap yields shorter project duration. A high proportion of project time dedicated to testing has a negative effect, but is not statistically significant. Frequent milestones significantly reduce project duration. Long redesign intervals create a higher technical content of the project and therefore delay project completion.

The positive beta coefficient of iterations was surprising, especially given that Eisenhardt and Tabrizi 1995 reported an acceleration effect. We see two reasons for this deviation. First, despite some intersection, the sample compositions differ substantially: we excluded projects with less than five person years while including other industries such as consumer products (e.g., TV, telephones) and measurement products (e.g., medical testing), which Eisenhardt and Tabrizi 1995 excluded. Second, this result reflects the direct delay impact of iterations: although design-build-test loops offer important learning opportunities, each iteration needs extra time. Releasing preliminary information early had no significant impact on project duration. The explanatory power increases dramatically: an adjusted R^2 value of over 44% is substantial given the cross-sectional nature of our study and especially the heterogeneous sample composition.

We now turn to the hypothesized impact of uncertainty reduction on project duration. Table 1 indicates at first sight that uncertainty reduction is not influencing project duration. However, the absence of significance only describes the direct effect of uncertainty reduction. Our hypothesis corresponds to its indirect (moderating)

effect.

Table 2 about here

4.4 Moderating Effect of Uncertainty Reduction

To explore the moderation effect of uncertainty reduction, we performed a subgroup analysis. As we (*a posteriori*) hypothesized uncertainty reduction to be the contextual variable, we divided our sample in two halves, below and above the median uncertainty reduction score. Support of our hypothesis would require significant differences across the two subsamples. The results are reported in Table 2. In terms of our theoretical model, this sample split corresponds to one group with the upper completion time curve in Figure 5 and one group with the lower curve.

Model 3 describes the subsample with fast uncertainty reduction. Overlap is significant at the 1% level indicating that early uncertainty reduction makes overlap more successful. Observe further, that the beta coefficient describing the acceleration effect of concurrency is, in absolute terms, substantially higher than in Model 2. Model 4 includes the observations that have a slower uncertainty reduction than the median. The significant influence of overlap (1% in Model 3) disappears. This goes in line with our moderating view to uncertainty reduction and confirms Figure 5: the benefit of concurrency is higher for fast than for slow uncertainty reduction. The empirical result suggests that the upper curve in Figure 5 is overall so flat, that the benefit is not significant.

The significant influence of testing observed in Model 3, disappeared in Model 4 (and even switched to a positive sign). That is, testing in projects with fast uncertainty reduction seems to have a delaying, rather than an accelerating effect. In the case of slow uncertainty reduction, we can observe that testing, instead of delaying, now becomes a crucial predictor for short development times. Its beta coefficient is,

in absolute terms, far higher than in the overall regression and highly significant (0.1% in Model 4). In addition to testing and overlap, other variables also change significance: the concept of frequent milestones seems no longer applicable in the case of late uncertainty reduction. If the path of the project can not initially be predicted, milestones can not be defined and are thus no longer an effective way of time reduction. However, if the project is highly predictable, milestones provide a useful tool for project management of keeping diverse activities coordinated and remain in control of the total project. This result is remarkable, as Eisenhardt and Tabrizi make "time between milestones" part of their experiential strategy. Our observation that the effect of frequent milestones on project duration is moderated by uncertainty reduction provides an interesting alternative explanation. Iteration keeps its delaying impact in both subsamples and remains unchanged from Model 2. Finally, release of preliminary information has a greater impact for slow uncertainty reduction than for fast uncertainty reduction.

The different beta coefficients and significance levels reported in Table 2 suggest that uncertainty reduction has a moderating effect on project duration rather than a direct effect. However, to formally support our hypothesis, we need a t-test on the differences in beta-coefficients between Model 3 and Model 4 (see our discussion above). Using our previously defined median split, we introduce a dummy variable "Uncertainty Reduction" defined as 1 for cases with slow (in comparison with the median, corresponding to large e) uncertainty reduction and 0 for the cases with a fast reduction. The dummy, as well as its interaction with all our seven independent variables, is then included in the regression. We used the interaction with all variables for two reasons. First, we do not have a theoretical basis for excluding an interaction between uncertainty reduction and any other variable (not including the interaction would force the corresponding moderating effect to be zero). Second, Table 2 indicates that not only overlap (as hypothesized) but also other explaining

variables change across different levels of uncertainty reduction. Table 3 summarizes the resulting regression.

Table 3 about here

Overlap retains its significant effect on project duration. The corresponding interaction effect with uncertainty reduction is significant, which provides us with the required formal test of our hypothesis. The reader may observe that for slow uncertainty reduction (interaction dummy is 1) the effect almost disappears (-1.179+.992=-.187), which again supports Figure 5. For fast uncertainty reduction the resulting impact is at -1.179. This is substantially above the coefficient reported in Table 1 and indicates the substantial power of activity overlap if the organization is able to reduce downstream uncertainty early in the process.

5 Discussion

Research on concurrent engineering has not sufficiently addressed the influence of contextual variables on the effectiveness of overlapping development activities. Most research has either focused on mathematical modeling or described good practices of concurrency without explaining theoretically the time reduction effect. Starting with a simple model, we combine insights from organizational theory and analytical modeling. Theorem 1 shows the moderating effect of uncertainty reduction on concurrency effectiveness. All three separately influence the optimal level of concurrency as shown in Theorem 2. The main insights derived from our model are analyzed based on the data drawn from 140 completed development projects in the global electronics industry. We first tested the influence of uncertainty reduction on overlap. To our suprise, faster uncertainty reduction was not combined with more overlap. This finding is of substantial managerial interest, as our further regression analysis identified, consistent with Theorem 1, that only projects with fast

uncertainty reduction are able to benefit from overlapping activities. Together with Theorem 2 this strongly suggests that projects in our sample could have reduced their project duration by choosing the overlap level in line with Theorem 2.

Our study is based on data from a relatively large, heterogeneous sample. Highly significant results, and at the same time, a relatively good measure of fit increase the generalizability of our research findings. Two issues have to be discussed from a methodological perspective. First, following our discussion of moderating perspectives from above, it is interesting to observe that the overall model fit does not significantly change compared to Table 1. A test statistic for this is given in Jaccard *et al.* 1990 and turns out to be not significant. Thus, our hypothesis that uncertainty reduction moderates the *form* of the relationship between overlap (testing) and project duration is supported, but there is no evidence in our data referring to the *strength* of this moderation.

If the uncertainty distribution over development time is unfavorable for overlapping activities and can not sufficiently be reduced by defining standards and architectures, the project organization has to search for other means of uncertainty reduction. The use of prototypes is a well-known project management decision in such a contingency (e.g., Wheelright and Clark 1992). Instead of following an overlapped phase process, design-build-test loops are used as a learning facility. In that case, a project then experiences a highly non-linear and iterative process which relies on experiencing product performance based on testing. The regression reported above suggests testing as an alternative way of reducing development time for projects where fast uncertainty reduction can not be obtained. The corresponding beta coefficient changes in the opposite direction to the one of overlap. Such an approach is consistent with the Eisenhardt and Tabrizi concept of "experiential strategy" that relies on frequent iterations and the rapid building of experience.

6 Conclusion and Future Research

This article links two, up to now distinctive, streams of research on concurrent engineering. The emerging construct of uncertainty reduction is found to significantly moderate the impacts of overlap on project duration. This view of concurrent engineering creates several opportunities for future research.

We did not address the question of where uncertainty reduction originates. Rapidly changing markets or uncertainty inherent to technology may force project teams to freeze their specifications late. On the other hand, uncertainty reduction may be an organizational capability that can be learned over the course of several projects. For example, Rosenkopf and Tushman 1994 report that when a technology advances to its "era of incremental improvement", the corresponding technology community gains more experience in product development, which leads to more efficient problem solving behavior. One example of such an evolution is the change in Microsoft's development process from trial and error in the 1980s (Gill and Iansiti 1991) to a streamlined and fine-tuned process today (Cusumano and Selby 1995). A finding that uncertainty reduction is technology specific would make generalizations across industries more difficult.

Second, we used a dummy variable in our interaction analysis, thus losing information compared to using a continuous measure. Including product terms for the interaction would require a finer measure instrument for uncertainty reduction. The development of such a new measure is an other interesting opportunity for future research.

Third, this article focused on the role of uncertainty reduction. Similar research is needed to understand the role of dependence (k in the model) and uncertainty level (μ_0). In addition, the role of communication provides an interesting opportunity for further conceptual and empirical work. Further empirical research is also needed to test whether project teams have become more effective in finding the "right" level

of overlap for their specific situation.

Finally, additional research is needed to integrate the construct of uncertainty reduction into existing organizational theory. With the emergence of a "dynamic organizational paradigm that captures key features of continuous adaptation" (Eisenhardt and Tabrizi 1995), a more dynamic view to classic constructs such as uncertainty is required (see also Adler 1995). As classic organizational models provide little understanding of how and why organizational processes are fast and flexible, including findings from engineering or operations management in multi-disciplinary research, seems a promising direction.

The authors are grateful to Charles Galunic and Michael Lapré for their helpful comments during the development of this article. We also wish to thank Martin Niederkofler and Lothar Stein (McKinsey & Co) for their cooperation.

7 References

Adler, P.S., "Interdeparmental and Coordination: The Case of the Design / Manufacturing Interface", *Organization Science*, 6 (1995), 147-167

Arnold, H. J., "Moderator Variables: A Clarification of Conceptual, Analytic, and Psychometric Issues", *Organizational Behavior and Human Performance*, 29 (1982), 143-174

Blackburn, J.D. (ed.) *Time Based Competition: The Next Battleground in American Manufacturing*, BusinessOne Irwin, Homewood, 1991

Blackburn, J.D., G. Hoedemaker and L.N. Van Wassenhove, "Concurrent Software Engineering: Prospects and Pitfalls", *INSEAD Working Paper* 94/65/TM, 1994

Brown, S. and K. M. Eisenhardt, "Product Development: Past Research, Present Findings, and Future Directions", *Academy of Management Review*, 20 (1995), 343-378

Clark, K.B. and T. Fujimoto, *Product Development Performance: Strategy, Organization and Management in the World Auto Industry*, Harvard Business School Press, Cambridge 1991

Clark, K. B. and S. C. Wheelright (Editors), *The Product Development Challenge*, Harvard Business Review Book Series, 1994

Cordero, R.,"Managing for Speed to Avoid Product Obsolescence: A Survey of Techniques", *Journal of Product Innovation Management*, 8 (1991), 4, 289-294

Cusumano, M. A. and R. W. Selby, *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*, The Free Press, 1995

Eisenhardt, K.M., "Making Fast Strategic Decisions in High-Velocity Environments", *Academy of Management Journal*, Vol. 32 (1989), No 3, 543-576

Eisenhardt, K.M. and B. N. Tabrizi, "Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry", *Administrative Science Quarterly*, 40 (1995), 84-110

Galbraith, J., *Designing Complex Organizations*, Addison Wesley, 1973

Gill, G. K. and M. Iansiti, "Microsoft Corporation: Office Business Unit", *Harvard Business School Case*, # 9-691-033, 1991

Ha, A. Y., E. L. Porteus, "Optimal Timing of Reviews in Concurrent Design for Manufacturability", *Management Science*, Vol. 41 (1995), No. 9, 1431-1447

Hoedemaker G.M., J.D. Blackburn, and L.N. Van Wassenhove, "Limits to Concurrency", *INSEAD Working Paper*, 1995.

Iansiti, M., "Technology Integration: Managing Technological Evolution in a Complex Environment", *Research Policy*, 24 (1995), 521-542

Imai, K., I. Nonaka, and H. Takeuchi, "Managing the New Product Development Process: How the Japanese Companies Learn and Unlearn", in *The Uneasy Alliance*, K.B. Clark, R.H. Hayes, and C. Lorenz Editiors, Harvard Business School Press, 1985

Jaccard, J., R. Turrisi, C. K. Wan, *Interaction Effects in Multiple Regression*, Sage University Paper (72), 1990

Krishnan, V., S. D. Eppinger, and D. E. Whitney, "A Model-Based Framework to Overlap Product Development Activities", *Working Paper MIT*, WP# 3635-93-MS, 1994

Loch, C. H., L. Stein, C. Terwiesch, "Measuring Development Performance in the Electronics Industry", *Journal of Product Innovation Management*, Vol.13 (1996), 3-20

Loch, C. H., C. Terwiesch, "Communication and Uncertainty in Concurrent Engineering", INSEAD Working Paper, 1996

Rosenkopf, L. and M. Tushman, "The Coevolution of Technology and Organization", in *Evolutionary Dynamics of Organizations*, J. A. C. Baum and J. V. Singh, Oxford University Press 1994

Takeuchi, H. and I. Nonaka, "The New Product Development Game", *Harvard Business Review*, 64 (1986), 137-146

Taylor, H.M. and S. Karlin, *An Introduction to Stochastic Modeling*, Academic Press, 1994

Terwiesch, C., C. H. Loch, M. Niederkofler, "Managing Uncertainty in Concurrent Engineering", Proceedings of the 3rd EIASM Conference on Product Development, 1996

Thompson, J.D., *Organizations in Action*, McGraw- Hill, 1967

Tushman, M.L. and D.A. Nadler, "Information Processing as an Integrating Concept in Organizational Design", *Academy of Management Review*, 21 (1978), 613-624

Van de Ven, A. H., A. L. Delbecq, R. J. Koenig, "Determinants of Coordination Modes within Organizations", American Sociological Review, 41 (1976), 322-338

Venkatraman, N., "The Concept of Fit in Strategy Research: Toward Verbal and Statistical Correspondence", *Academy of Management Review*, Vol. 14 (1989), No 3, 423-444

Wheelright, S. C. and K. B. Clark, *Revolutionizing Product Development*, The Free Press, 1992

Wirth, N. *Algorithmen und Datenstrukturen*, Teubner Stuttgart, 1975

Appendix

Intermediate Result 2: On Λ, $E[T(\lambda)]$ is convex in λ

Proof: The second derivative of (8) is

$$\frac{\partial^2 E[T]}{(\partial\lambda)^2} = k\mu t_2^2[1 + e(1 - 2\lambda\frac{t_2}{t_1})] \tag{10}$$

By model setup, $1 - 2\lambda\frac{t_2}{t_1} \geq -1$, thus

$$\frac{\partial^2 E[T]}{(\partial\lambda)^2} \geq k\mu t_2^2 \, min\{1+e, 1-e\} \geq 0$$

since $e \in [-1, 1]$. □

Proof of Theorem 2

Solving the first order condition of (7), we get two solutions:

$$\lambda_{1,2} = \frac{(1+e)t_1}{2et_2} \pm \sqrt{\frac{(1+e)^2 t_1^2}{4(et_2)^2} - \frac{t_1}{k\mu_0 t_2^2 e}} \tag{11}$$

For notational convenience, let

$$\begin{aligned} A &:= \sqrt{\frac{(1+e)^2 t_1^2}{4(et_2)^2} - \frac{t_1}{k\mu_0 t_2^2 e}} > 0 \\ \lambda_1 &:= \frac{t_1 + et_1}{2et_2} - A \\ \lambda_2 &:= \frac{t_1 + et_1}{2et_2} + A \end{aligned}$$

We focus on the impact of uncertainty reduction. The proof for k and μ_0 is easier and can be done directly by inspection of the optimal solution. As the objective function (7) is cubic in λ, the position of the local minimum depends on the sign of e. For $e > 0$, the minimizing solution to the first order condition is in λ_1 and for $e < 0$ it is in λ_2. Further, if $e = 0$, the solution is $\lambda^*(e = 0) = \frac{1}{k\mu_0 t_2}$. In summary:

$$\lambda^*(e) = \begin{cases} \lambda_1 \text{ if } & e > 0 \\ \frac{1}{k\mu_0 t_2} & e = 0 \\ \lambda_2 \text{ if } & e < 0 \end{cases}$$

Simplifying the expressions for λ_1 and λ_2 we can use l'Hôpital's rule to show continuity of $\lambda^*(e)$:

$$\begin{aligned}
\lim_{e\to 0^+} \lambda^*(e) &= \lim_{e\to 0^+} \lambda_1 \\
&= \frac{1}{k\mu_0 t_2} = \lambda^*(e=0) \\
&= \lim_{e\to 0^-} \lambda_2 = \lim_{e\to 0^-} \lambda^*(e) \qquad (12)
\end{aligned}$$

Given continuity, it now is sufficient to show $\frac{\partial\lambda^*}{\partial e} < 0$. Using the implicit function theorem, we know

$$\frac{\partial\lambda^*}{\partial e} = -\frac{\left.\frac{\partial^2 E[T]}{\partial\lambda\partial e}\right|_{\lambda=\lambda^*}}{\left.\frac{\partial^2 E[T]}{\partial\lambda^2}\right|_{\lambda=\lambda^*}}$$

where the denominator is positive by Lemma 2 and the nominator is positive by Theorem 1. Thus λ^* is decreasing in e. □

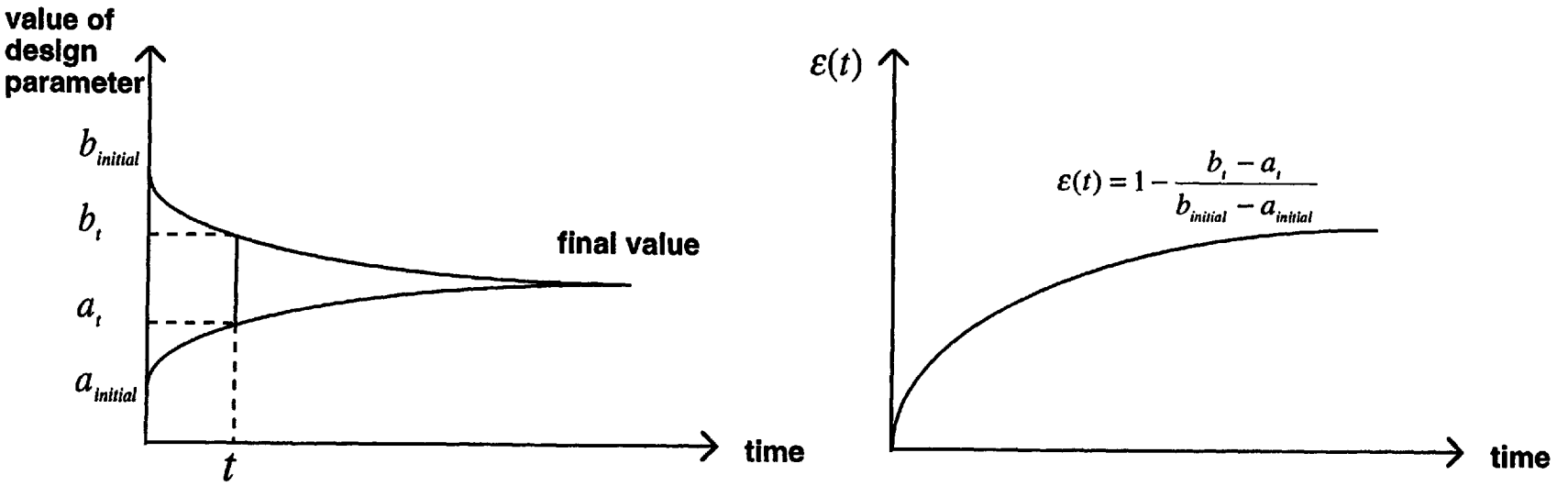


Figure 1: Evolution Function (Krishnan *et al.* 1995)

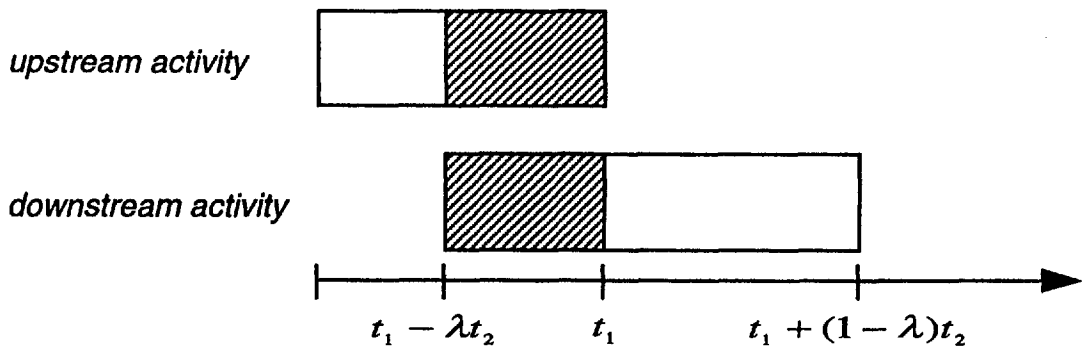


Figure 2: Set-up of the model

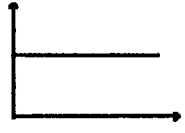
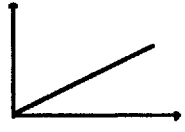
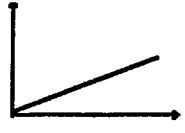
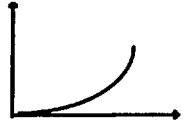
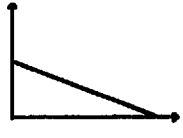
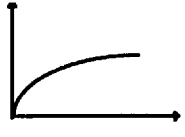
e	Rate of Poisson Process $\mu(t)$	Graph of $\mu(t)$	Krishnan's $\varepsilon(t)$
0	μ_0		
1	$2\mu_0 \frac{t}{t_1}$		
-1	$2\mu_0\left(1-\frac{t}{t_1}\right)$		

Figure 3: Uncertainty Reduction

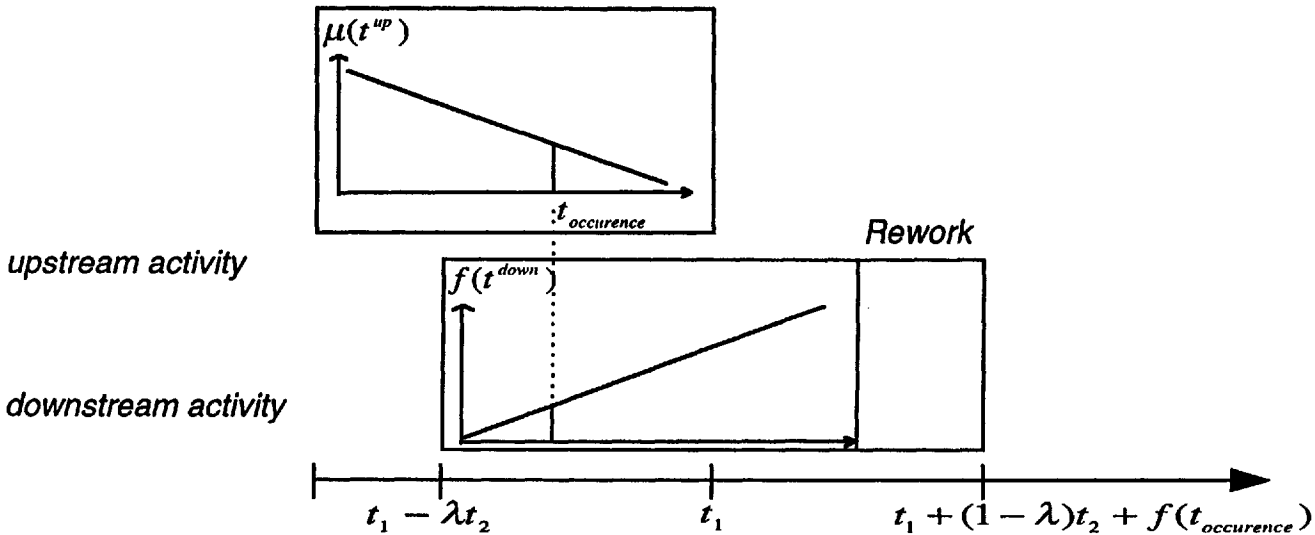


Figure 4: Relation between Uncertainty Reduction and Impact Function

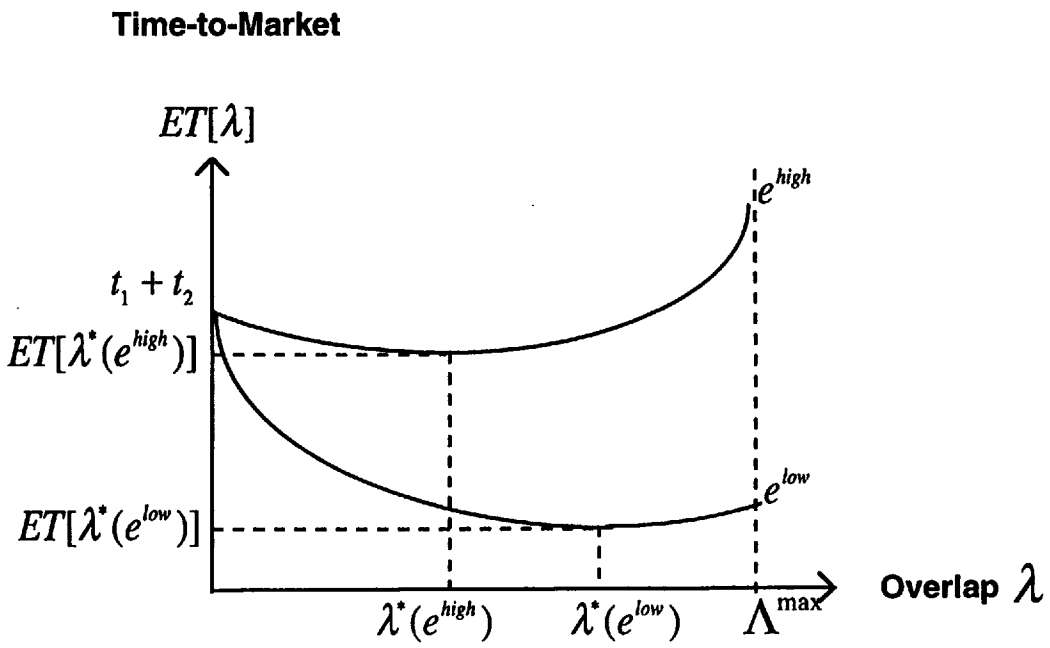


Figure 5: Benefits of Overlap compared for slow and fast Uncertainty Reduction

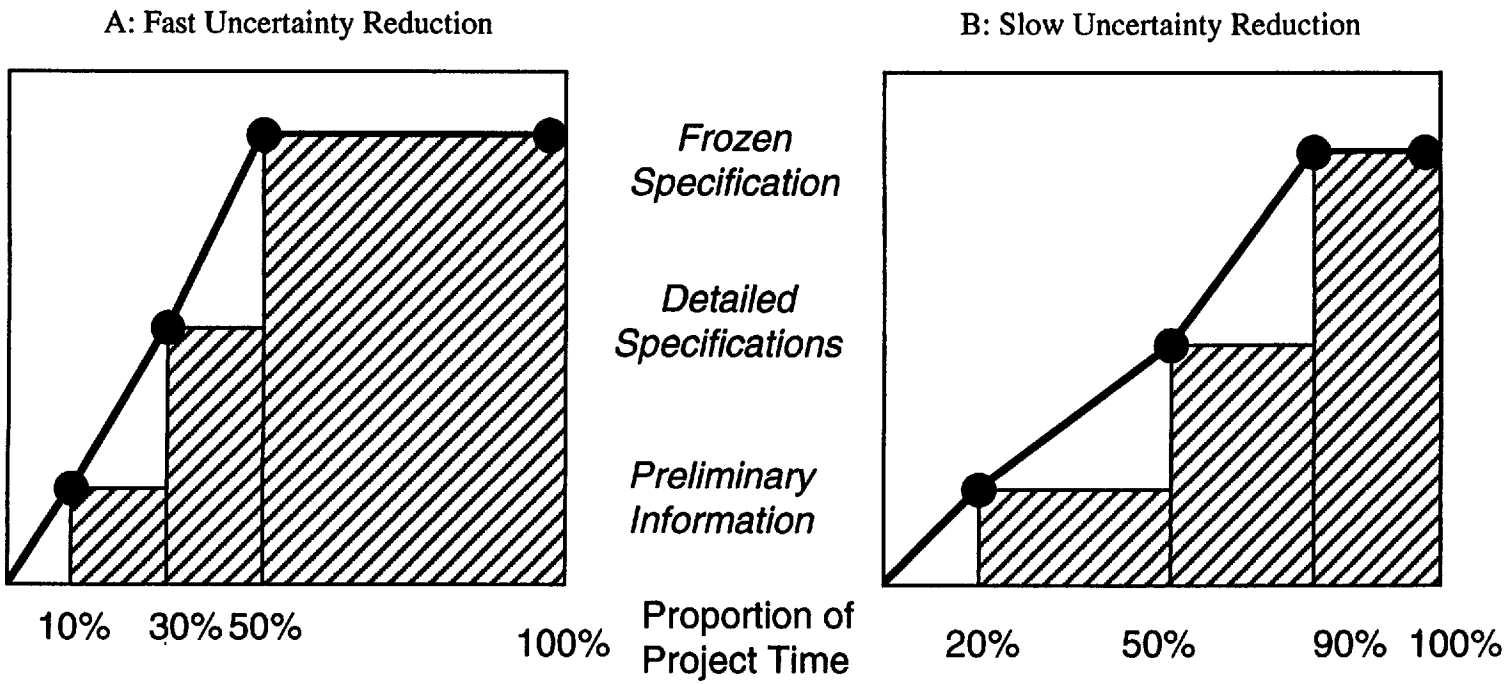


Figure 6: Measurement of Uncertainty Reduction; Project A (left) with Fast Uncertainty Reduction and Project B (right) with Slow

Variable	*Model 1*	*Model 2*
Project Size	*.163****	*.115****
Overlap		*-.652****
Testing		*-.518*
Time between Milestones		*.169****
Iterations		*.137****
Redesign Intervals		*.112***
Preliminary Information		*-.047*
Uncertainty Reduction		*.044*
Adj. R^2	*.230****	*.445****

<.10; **<.05; *<.01; N=140*

Table 1: Results of Regression Analysis for Development Time

Variable	*Model 3*[a]	*Model 4*[b]
Project Size	*.124***	*.119***
Overlap	*-1.001***	*-.378*
Testing	*.191*	*-2.060***
Time between Milestones	*.195***	*.102*
Iterations	*.195***	*.141***
Redesign Intervals	*.081*	*.143**
Preliminary Information	*-.018*	*-.088***
Adj. R^2	*.445***	*.485***

*<.10; **<.05; ***<.01

[a] *N=70: early uncertainty reduction*

[b] *N=70: late uncertainty reduction*

Table 2: Split Sample Analysis with uncertainty reduction

Variable	*Model 5*
Project Size	*.129****
Overlap	*-1.179****
Testing	*-.010*
Time between Milestones	*.183****
Iterations	*.194****
Redesign Intervals	*.086*
Preliminary Information	*.008*
Interaction of Dummy with	
Project Size	*.014*
Overlap	*.992***
Testing	*-1.478***
Time between Milestones	*-.014*
Iterations	*-.068*
Redesign Intervals	*.060*
Preliminary Information	*-.084*
Adj. R^2	*.459****

*<.10; **<.05; ***<.01; N=140

Table 3: Overall Regression with a Dummy variable for Uncertainty Reduction