

EFFECTS OF POSTPONEMENT STRATEGIES FOR MANAGING THE DESIGN PROCESS IN CONDITIONS OF UNCERTAINTY

Gil, N.¹, Tommelein, I.D.², and Kirkendall, B.³

ABSTRACT

This paper explores the effect of postponed commitment strategies for managing the design process in conditions of uncertainty. We present a generic product-process development model representative for the design process of diverse building systems in a semiconductor facility. The model synthesizes the knowledge we gained after conducting empirical research. The model comprises an initial conceptualization effort followed by a concept development phase. Between the two phases, we assume that managers can impose a time lag so as to minimize rework of concept development tasks in the event design criteria change. Simulation illustrates the effects of adopting different postponement strategies, according to the assumed nature of rework. The results show that postponing the execution of concept development persistently results in an increase of the average of the total project duration and a decrease of its variance, regardless of the lag duration. Moreover, postponement diminishes the average and variance of the total man-hours spent in concept development. Such effects are more prominent as the duration of rework tasks approximates the duration of the first time the task was performed. Set-based design, an alternative to postponement, reveals more efficient in conditions of uncertainty.

INTRODUCTION

We define postponed commitment as a managerial strategy that intentionally instructs designers to delay the execution of a task or set of tasks instead of starting them with incomplete or unreliable information inputs and criteria. The motivation for managers is the expectation that by

-
- 1 Doctoral Candidate, Constr. Engrg. & Mgmt. Program, Civil & Envir. Engrg. Dept., 215 McLaughlin Hall, U.C. Berkeley, CA 94720-1712, USA, ngil@uclink4.berkeley.edu, <http://www.ce.berkeley.edu/~nunogil/>
 - 2 Associate Prof., Constr. Engrg. & Mgmt. Program, Civil & Envir. Engrg. Dept., 215 McLaughlin Hall, U.C. Berkeley, CA 94720-1712, USA, tommelein@ce.berkeley.edu, <http://www.ce.berkeley.edu/~tommelein/>
 - 3 Senior Interior Designer, Industrial Design Corporation, 2020 S.W. Fourth Avenue, 3rd Floor, Portland, OR 97201, USA, robert.kirkendall@idc-ch2m.com

implementing such strategy designers will not need to rework the tasks in the event information inputs or criteria changes. Postponement therefore means managers believe changes are prone to occur in the future. Postponement also assumes managers expect changes may be significant to the extent they will cause design rework. Such uncertainty on the future frequently faces managers working in unpredictable environments, such as the computer industry (Iansiti 1995, Eisenhardt and Tabrizi 1995).

In this work, we study the effect of postponed commitment strategies applied to the design development processes of semiconductor fabrication facilities (fabs). We associate to design tasks decisions being made on design parameters. Consequently, as managers would opt to delay the execution of tasks they would be postponing the decision-making process on the design parameters. Our initial rationale was based on the intuition that, given the propensity for changes in design criteria in the semiconductor industry, fab designers would be better off if they would delay tasks to the last possible moment that would still let them meet the project delivery date.

The purpose of this paper is twofold. First, it advances a set of managerial considerations on the effects postponed commitment strategies may cause to the design process in conditions of uncertainty of design criteria. Secondly, it illustrates a method to explore a theoretical managerial proposition grounded on empirical research using the capabilities of a simulation environment. This paper does not focus however on the consequences of postponement strategies to the quality of the product design. We briefly address this issue at the end of this paper in light of the simulation results.

EMPIRICAL RESEARCH

Empirical research lasted approximately from November 1998 to August 2000 during which the authors have worked jointly with practitioners in the Industrial Design Corporation (IDC), a major design-build firm in the high-tech industry. We conducted approximately one hundred interviews with lead designers from different specialties, such as mechanical, chemical, architecture, structural, and electrical, managers, design coordinators, engineers and architects, and draftsmen. We visited different projects of construction and retrofitting of fabs and conducted interviews with people working in diverse divisions at client organizations. The interview process focused on the product decisions design teams have to make during design of a fab, the criteria that drive those decisions, and the hand-offs of information between specialties

and between designers and client organizations. We audio taped and transcribed all the interviews.

RELATED RESEARCH

Design processes are intrinsically complex in nature. Complexity stems from diverse factors, such as interdependencies and coordination needs between tasks carried by distinct design disciplines (Chrichton 1966, Gebala and Eppinger 1991, Malone 1997) and the iterative nature of the design process in its search for satisfying solutions (Simon 1969, Conklin and Weil 1997). Externalities such as the critical need to reduce product development time in industries where a premium exists for the first firm that hits the market, and unpredictable market demand add other layers of complexity (Eisenhardt and Tabrizi 1995, Iansiti 1995).

Many academic studies have aimed to build theory on the nature of design processes and develop tools to help managing such processes in practice. Stage-gate systems, for instance, propose the concept of gate as an entrance to a design phase or stage. The purpose of these gates is to review the product quality of the work developed and decide which actions teams should do next at the end of each phase (Cooper 1990). At a less aggregate level of abstraction, the Design Structure Matrix (DSM) models design tasks and respective interdependencies, assuming a sequential evolvment of the design process (Gebala and Eppinger 1991, Smith and Eppinger 1997a). DSM provides partitioning and tearing algorithms that order the tasks so as to minimize the information loops and the total duration of the process. DSM is however a static model in the sense it assumes a given design process but ignores the dynamic nature of such process in conditions of uncertainty.

Work in the field of computational and mathematical organizational theory produced diverse models without domain specifics that provide insight into the nature of the design process in conditions of uncertainty (Carley 1995). Jin and Levitt (1995) describe VDT, a process-information model that implements the micro behavior of actors so as to gain insight of their influence in the overall performance of complex organizational systems. Lin and Hui (1997) developed similar work to compare the performance of lean and mass organizational systems. In the field of system dynamics, Ford and Sterman (1998) studied the effects on the design process of task concurrency and the propensity of actors to conceal changes.

Other work on design process management results from empirical research. This is the case with studies on concurrent engineering and product development practices in the automotive

industry (Clark and Fujimoto 1990, Womack et al. 1990, Ward et al. 1995, Sobek II et al. 1999) and in unpredictable environments (Eisenhardt and Tabrizi 1995, Iansiti 1995, 1997, Thomke and Reinertsen 1998). These studies elucidate on the product and process benefits of different managerial strategies, such as, for instance, the early and long-term involvement of suppliers in product development, postponement of decisions, and set-based concurrent engineering.

The complexity of design processes in the AEC industry stems from many factors, some of the most representative being its fragmented and single prototype nature, the complexity of the product definition, the uncertainty of design criteria, and the criticality for fast delivery of AEC products (Crichton 1966). In recent years, practitioners have reactively used the fast-track delivery method. In fast-track projects, design specialties work concurrently and release in a coordinated and staggered fashion batches of design products so builders can start executing some building systems while designers complete the definition of the remaining ones. Generally implicit in such strategy is however an early commitment on design decisions. Such early commitment frequently results in slippage of promised project dates and extensive rework, for which construction projects are regrettably commonly known (Pietroforte 1998).

Recent work of Wood (1998) analyses to what extent a fast and flexible delivery of AEC products in the semiconductor industry environment, such as fabs and installation of tools, would help manufacturing firms meet their needs for speed and flexibility. The following work in this paper follows that line of thought.

PRODUCT-PROCESS DEVELOPMENT MODEL OF DESIGN

Figure 1 presents a generic product-process model for the design development of fabs. The model synthesizes the product and process knowledge we gained through interviews with practitioners. We define the design development process as composed of two distinct phases: an initial conceptualization effort followed by a concept development phase.

During conceptualization, designers make a first pass of the design criteria and parameters based on the needs of the client, with the help of empirical rules and historical data. Designers also evaluate to what extent the design definition of fabs the client may possess can be reused. Replication of decisions in terms of product parameters and configuration of building systems happens frequently between fabs and may even be established at a corporate level (McDonald 1998). Because of the multitude of variables involved in the design definition of fabs and the complexity of the interdependencies between building systems manufacturing organizations have

a limited understanding of how the fab design may influence the production yields. As a result, organizations typically expect to increase their chances to successfully replicate the yields already obtained in existing facilities by copying the fab definitions.

During concept development designers refine the decisions done at conceptualization in light of updated information and using sophisticated analytical tools. The model expresses concept development as a loop of three tasks: loads, sections, and layout, which we define next.

LOAD DEVELOPMENT: This task expresses the effort to decide the loads each building system should serve. The information needed to decide the loads varies by discipline. Instances of such information are the density and weight of tools in the cleanroom for structural designers and the exhaust requirements of the production process for mechanical designers.

SECTION DEVELOPMENT: This task expresses the effort to size the sections of the main elements in each building system based on the loads previously determined. Structural designers, for instance, size the thickness of the waffle slab floor and the cross-section of the subfab columns, and mechanical, electrical, and chemical designers size the cross-sections of the utility routings at critical points. In addition, designers often decide the materials for the routings since such choice affects the sizing of the sections.

LAYOUT DEVELOPMENT: This task expresses the effort to decide the routing of the utility systems in the tri-dimensional space and the location of major pieces of equipment. Thus, for instance, structural engineers will decide the grid that sets the location of the columns in the subfab, and chemical, mechanical, and electrical designers decide on the utility routings.

Internal and external conditions may drive designers to iterate through the aforementioned design loop. On one hand, designers may do several passes in the loop in their search for a satisfying solution if time allows, given the iterative nature of design (Simon 1969). Such iterations are internal in nature. They may happen even in the event designers possessed all the information they needed and design criteria would not change. For simplicity, the simulation model hypothetically assumes however a scenario where designers would only have to iterate once to find a satisfying solution, in the event design criteria would not change.

On the other hand, iteration may be caused by externalities to the design process of one specialty, such as interdependencies with other design specialties or changes in design criteria. This generic model disregards the interdependencies between concurrent design processes. Gil

and Tommelein (2000) present a more complex model that introduces concurrency between design processes and hand-offs of information between specialties.

UNCERTAINTY IN DESIGN CRITERIA

Uncertainty in criteria for fab design stems from diverse factors such as the concurrency of the design effort with the research and development of the production technology the fab will house, and the unpredictability of market demand for the product to produce inside the fab. Such uncertainty frequently translates into successive changes of the fab design criteria along the design process. Specifically, uncertainty affects two design criteria at the core of the design process—the dimensions of the cleanroom and the list of tools to install inside.

The decision on the cleanroom dimensions ties to strategic decisions the manufacturer organization must make regarding, for instance, the chip production capacity of the fab and the date the chips should hit the market. A change in the cleanroom dimensions significantly disturbs the design process. Such change causes designers to rework the initial conceptualization effort and the tasks in concept development. Changes in the list of process tools are more frequent than cleanroom changes. They result from changes of the production process technology or of tool suppliers. These changes may directly affect the location of tools and the utility loads that need to serve the tools. This work only models significant changes of the list of tools in the sense that the changes force AEC designers to recalculate the loads and downstream tasks. We neglect the impact tool list changes may have in the conceptualization effort, given the greater flexibility designers have then to accommodate these changes.

In Figure 2, we represent the pattern of uncertainty we assumed in the simulation model. We assumed the changes in the cleanroom and list of tools to be stochastically independent. We used rescaled and relocated beta random variables $[a+ (b-a) \cdot \text{beta}(\alpha_1=2, \alpha_2=2)]$ to express the time variability when a change can occur. We decreased the probabilities of subsequent changes by multiplying the probabilities of the first change in the cleanroom (probability of first change=0.5) and tool list (probability of first change=0.9) respectively by the terms of the sequences $\frac{1}{1.5}, \frac{1}{2}, \frac{1}{2.5}$ and $\frac{1}{1.25}, \frac{1}{1.5}, \frac{1}{1.75}$. In addition, we linearly increased the rescaled interval of each beta distribution (b-a) between subsequent changes. This qualitative pattern of uncertainty was cast after a consensus among lead designers. The pattern is exclusively valid to

express uncertainty in the design criteria for fabs that house complex process technologies such as leading edge microprocessors and application specific integrated circuits (ASICs).

ON THE NATURE OF REWORK

Hopp and Spearman (1996) discuss rework in the context of factory physics. They assume a machine produces defective parts and study the effects of reworking those parts in terms of performance of the production line. With the help of computer simulation, Hopp and Spearman demonstrate the negative consequences of rework to cycle time, throughput, and work in process. They conclude the longer the rework loop, the more pronounced its consequences (Hopp and Spearman 1996, pp. 362). In this work, we study the effects of postponement strategies to the design process assuming three different scenarios for the nature of rework loops.

In the first scenario we assume each time a change occurs, in the subsequent iteration the expected duration for each task will be equal to its initial expected duration, given we keep the same level of resources. Such scenario is valid both for the tasks interrupted when the change occurred and for those tasks that were done when the change occurred:

$$D_{i+1} = D_i = D_1, \forall i \quad (\text{equation 1})$$

i – number of times designers start to reiterate the task ($i=1,2,3,\dots$)

D_{i+1} – expected duration of a task in the next iteration, if a future change will not interrupt its execution.

D_i – expected duration of the task in the last iteration, if the last change did not interrupt its execution.

D_1 – expected duration for the task in the first iteration, if no interruptions occur.

In other words, we assume designers would not learn or gain process efficiencies between two iterations of the same task. Accordingly, designers would have to redevelop the rationale from scratch and revisit all decisions already made. Given that we only model changes that significantly alter the design criteria we believe this is a plausible model. Particularly, such model suits well a scenario, somewhat common in practice, in which designers other than the ones who previously executed the tasks would be in charge of reiterating the tasks because the former would be in the meantime assigned to a different project.

In the second scenario, we assume learning and efficiency gains exist between iterations. To determine the duration of a task in a rework cycle, we prorate its duration in the precedent cycle with the following equation:

- if designers had concluded the task when the change occurred:

$$D_{1,n+1} = \frac{n \cdot D_{1,n}}{n+1} = \frac{D_{1,1}}{n+1}, \forall n \quad (\text{equation 2.1})$$

- if the change interrupted the execution of the task:

$$D_{i+1,n} = D_{i,n} - T_{i,n} + \frac{n \cdot T_{i,n}}{n+1} = D_{i,n} - \frac{T_{i,n}}{n+1}, \forall n, \forall i \quad (\text{equation 2.2})$$

i, n - number of times designers have started to iterate the task ($i=1,2,3,\dots$), given a previous number of times designers have already completely executed the task ($n=1,2,\dots$)

$D_{i,n}$ - expected duration for the task in iteration i given they have already completely executed the task n times, if a design change will not interrupt its execution

$T_{i,n}$ - period of time the design team spent working on iteration i , given they have already completely executed the task n times.

$D_{1,1}$ - expected duration for the task the first time designers will execute it, if a design change will not interrupt its execution.

In the third scenario, we assume the design team adopts a set-based design strategy. Set-based concurrent engineering is an alternative to point-based design. In point-based design, designers make early commitments on a single solution and progressively refine it as the design process evolves. In contrast, in set-based design, designers work with sets of solutions that they gradually narrow as information on design criteria and customer expectations sharpens (Sobek II et al. 1999). In the model, we assume designers can predict beforehand all plausible directions design criteria may take in the future. Accordingly, designers initially start working with an exhaustive set of solutions in the sense that any future change of design criteria falls within its boundaries. As a result, designers do not need to rework the design tasks in the event a change in criteria occurs. Still, we model a fixed time penalty to express the effort designers will incur in revisiting the tasks so as to exclude the inconsistent solutions in light of the updated design criteria, as equation 3 expresses:

- if designers had concluded the task when the change occurred:

$$D_{i+1} = c \cdot D_1, \forall i \quad (\text{equation 3.1})$$

- if the change interrupted the execution of the task:

$$D_{i+1} = D_i - T_i + c \cdot D_1, \forall i \quad (\text{equation 3.2})$$

i - number of times designers start to iterate the task ($i=1,2,3,\dots$)

D_i - expected duration for the task in iteration i , if a design change will not interrupt its execution

D_1 – expected duration for the task the first time designers will execute it, if a design change will not interrupt its execution

T_i – period of time designers spent on the task at iteration i

c – fixed penalty that expresses the design effort of pruning the inconsistent solutions from the original set of solutions

Because we lacked precise information on what the fixed penalty for pruning the sets of solutions from inconsistent ones may be in practice, we assumed it to be 20 percent of the original duration of each task ($c=0.2$). A value of c equal to 0 would mean this effort was insignificant. A value of c equal to one would make the set-based design scenario perform approximately like the scenario where no learning would exist between iterations.

We acknowledge these three scenarios are far from exhausting the range of scenarios we can possibly envisage to model rework. We believe however that the additional insight other scenarios could provide to sharpen our understanding on the effects of postponement would not outweigh the increase in complexity of this paper.

SIMULATION RATIONALE

The design process simulation starts with the conceptualization effort. Once designers finish this phase, they may opt to immediately start doing the first task of concept development (LOAD) or postpone its start date. In the event designers opt for a postponement strategy, they would a priori decide the last possible day after which designers should start concept development so as not jeopardize the promised delivery date to the client. Our intuition for a postponement strategy was the following. Given designers common belief in the propensity of criteria to change along the design process they would be better off in postponing the start of concept development so as to minimize rework. By the time designers would then start concept development hopefully no more changes would occur and they could develop the design in a single pass.

This intuition translates in the following simulation rationale. We assume the client sets a priori a milestone date that defines the last possible day he will consider a significant change in design criteria. We set this milestone date to be day 120, which is 6 months after the project starts (a month in the simulation comprises 20 working days). As such, the model can account for all the changes that can possibly occur before the milestone date, whether or not designers had already completed the concept development loop. In fact, during empirical research we learned of multiple projects where a change occurred after designers had formally considered

concept development as completed. In specific cases, changes have inclusively occurred as late as when designers were detailing the design of the building systems, and occasionally the foundations were concurrently being executed on site, or even the steel structure was being erected. In these circumstances, the client did not have other alternative other than stop the construction process, eventually tear the construction down, and reiterate the design process.

Simplifying Assumptions

For clarity, we made some simplifying assumptions in this work that we pass to address:

1. We assumed each task had a deterministic duration, despite the fact that the simulation model lends itself to easily express a stochastic duration. In practice, the duration of tasks is stochastic. From our experience in experimenting with the model, such stochastic behavior caused an insignificant difference in terms of the average results for the performance variables if compared with the results of the deterministic model. Logically though, the stochastic behavior increases the variability of the performance variables.
2. We used the second rework scenario to model rework of the conceptualization effort due to changes in the cleanroom dimensions. We purposely kept this condition unchanged in all scenarios so as to be able to more clearly evaluate the effects of postponement on the subsequent phase of concept development.
3. We assumed availability of resources to execute the tasks, whether or not managers would decide to postpone the concept development effort. In practice, this may not be a trivial problem for managers. We leave such discussion to the end of this paper.
4. In case designers would adopt a set based design strategy they would work with sets of solutions instead of single point solutions. We assume designers would have the computational means so as not to increase the men-days of work per task. Actual research indicates such computational means are available (Smithers 1989, Lottaz et al. 1999) and instances in current practice suggest innovate firms use them (Sabbagh 1996).

Validation

Law and Kelton (2000 pp. 277) use the expression “face validity” to characterize a model whose results practitioners consider reasonable and consistent with the way they perceive the system behavior. Given the limited size of the model, there would be no point in trying a more precise validation of the simulation results. To maximize the conformance between the simulation model with the way practitioners perceive the design process, we walked through the model with lead

designers. The model in this paper incorporates the feedback we received from practitioners after the walk-throughs.

Discrete Event Scheduling Simulation

The model was implemented with the simulation engine SIGMA (Schruben and Schruben 2000). SIGMA is a discrete-event simulation environment based on the concept of event graph. Event graphs capture the events taking place within a system and the relationships among these events. The process simulation evolves by executing the list of future events in a chronological manner, and updating it each time a new event gets processed.

We tested 16 postponement strategies for each rework scenario, keeping constant the pattern of uncertainty in all batches of runs. We assume the conceptualization effort lasts 25 days, if no changes interrupt it, in which case designers would have to iterate the effort. One extreme scenario assumes designers would start concept development immediately after the end of the conceptualization effort. This strategy means the team would start load development on day 25 (if no changes had occurred during conceptualization) or on whatever day the conceptualization effort would end, in case a change had occurred during conceptualization. The other extreme scenario assumes designers would postpone the start of load development up to day 110 (corresponding to a lag of 85 days if the conceptualization effort had finished on day 25) so as to maximize the probability of executing concept development in a single pass. In between, we tested alternative strategies by postponing the date to start load development in increasing intervals of 5 days, from day 25 up to day 110. Naturally, we could extend such strategy up to day 120, the day when we assumed the client would stop considering design changes. We opt for not doing it given the results of the performance variables stabilized earlier.

All models were run in SIGMA. SIGMA automatically generates source code in C, which can then be compiled into executable versions with Microsoft Visual C/C++ Version 6.0. One thousand iterations of the compiled version take on the order of 10 seconds, on a Pentium 600-MHz computer running Windows 98. Source code is available (Gil and Tommelein 2000) so readers can reproduce and further experiment with alternative inputs.

Performance Variables

To evaluate the effect of postponement on the design process we defined the following performance variables:

1. Total project duration: the period of time elapsed between the day the conceptualization effort starts and the day when the last iteration of concept development finishes.
2. Total resources spent in concept development. For simplicity, we allocate one unitary resource to each task in concept development (the reader can imagine it as if a lead designer would be the sole person responsible for the task execution). As a result, the total added time spent in iterating the concept development tasks is a measure of the man-days of work spent in this phase.
3. Number of design iterations of each task. This metric includes all iterations of the design task, regardless of the state of progression when a change interrupted the task.

SIMULATION RESULTS

Design Process Model without Uncertainty

Fig. 3a shows the results of the design process simulation for a baseline scenario without changes of design criteria. The shape of curves in Fig. 3a reflects the deterministic duration we consider for each task, respectively 25 days for the conceptualization effort, and then 5, 10, and another 10 days respectively for load, section, and layout development. These are the average duration of these tasks for the design process of the process piping systems, according to empirical research (Gil and Tommelein 2000). We opted for the chemical process because the change in the tool list directly affects the calculation of the chemical loads.

Fig. 3a illustrates 3 curves, one for each following postponement strategy: 1) no postponement, 2) concept development shall not start before day 45 (corresponding thus to a postponement lag of 20 days, if the conceptualization stage evolves in a single pass), and 3) concept development shall not start before day 100. On the (X) axis we chart the simulation time. On the (Y) axis we chart the design tasks in the sequential order the design team would execute them if no changes would occur meanwhile. Each specific curve connects the points corresponding to the start and finish dates of the conceptualization effort and of the three tasks in concept development: load, section, and layout.

In this deterministic scenario without uncertainty, a postponement strategy does not bring any value in terms of resource savings. The effect of postponement is thus exclusively to delay proportionally the date of conclusion of concept development.

Design Process Model with Uncertainty

As we implement the uncertainty pattern shown in Fig. 2 in the design process simulation, the model exhibits a random behavior. Uncertainty results from the stochastic nature of changes in the cleanroom dimensions and list of tools. Each simulation run tends to evolve differently according to when changes occur and their frequency of occurrence. For each scenario we run 1000 simulations. By the central limit theorem, we can treat the independent and identically distributed 1000 observations that result for each performance variable as approximately distributed to normal random variables (Law and Kelton 2000). We calculated accordingly the average and variance of each variable. Table 1 summarizes the average and standard deviation of the number of cleanroom and tool list changes in the 120 days period we assumed the client would consider changes in design criteria. Table 1 also includes the share of these changes that fall within the initial conceptualization stage. Logically, these do not influence the effects of postponement strategies in the design process model.

Table 1. Average and Standard Deviation of Total Number of Changes

No. of Changes in the Cleanroom Dimensions		No. of Changes in the Tool List		No. of Total Changes Falling within Conceptualization Phase	
Average	Standard Deviation	Average	Standard Deviation	Average	Standard Deviation
0.72	0.83	2.14	1.29	0.88	0.66

Fig. 3b illustrates an instance of a single simulation run from a scenario characterized by no learning between iterations and no postponement lag. In this specific run, the design process was interrupted three times: first by a change in the cleanroom dimensions during section development, second by a change in cleanroom dimensions during layout development, and third by a change in the tool list after completion of concept development. Figs. 3c and 3d illustrate the results of 50 iterations from two scenarios characterized both by no learning between iterations, and without postponement lag (Fig. 3c) and with a postponement lag such that the concept development shall not start before day 50 (Fig. 3d). Figs. 3e and 3f, and 3g and 3h replicate both postponement strategies but with the rework equations for limited learning (equation 2) and set-based design (equation 3) respectively.

Fig. 4 charts the relationship between the average total project duration and the average total resources spent in concept development that results as we increase the postponement lag. Each data point in the chart and its respective one standard deviations in the (X) and (Y) axis result

from applying the central limit theorem to the set of 1000 variables that result from 1000 independent runs. In the specific case of Fig. 4, we used the rework equation that assumes no learning exists between tasks. Fig. 5 shows three other curves in addition to the curve represented in Fig. 4: one curve for the baseline scenario without uncertainty, and two curves corresponding to the rework equations with learning and set-based design. We kept the same stochastic pattern of uncertainty.

Finally, Fig. 6 charts the variation of the average number of iterations for each task and of changes falling at the postponement lag and after completion of concept development in function of the postponement strategy, for the rework algorithm with no learning.

ANALYSIS AND DISCUSSION OF RESULTS

Postponement strategies are seldom used in current practice in the design processes of fabs, at least to the extent we observed and discussed with practitioners during empirical research. The common argument invoked by practitioners is that if they adopted a postponement strategy they would be jeopardizing their ability to meet the milestone dates clients impose. In other words, designers believe that every possible day of work counts in order to meet the deadline and therefore act accordingly. Nonetheless, designers also acknowledge that frequently they iterate several times the same tasks because of criteria changes but seem resigned to accept iteration as an intrinsic characteristic of the design process. We started this work with the intuition that many of these iterations were needless and could be prevented without compromising the project deadlines if designers would adopt postponement strategies.

This work brought a more meaningful insight to such intuition. As Figs.4 and 5 illustrate, a strategy of postponement always increases the average total duration of the design process associated with a decrease of its variability, regardless of which rework equation we consider. Such effect is more prominent as learning decreases between iterations. Postponement strategies also decrease the average resources spent in concept development and its variability.

An analysis of the marginal effect of postponement in the performance variables adds more insight to these effects. Fig. 4 illustrates that initially, as we increase the postponement lag, the marginal gains in terms of decreasing the average spent resources are the most significant, without significantly increasing the average total duration of the design process. In addition, if instead of the average total project duration (μ_t), we consider the results of this variable within a

confidence interval of one standard deviation (σ_t), than the upper limit of the interval ($\mu_t + \sigma_t$) is almost coincident for an initial set of postponement strategies.

In Fig. 4, we schematically graph two rays that define in-between what we call an “*efficiency zone*” for the design process. The efficiency zone defines a set of postponement strategies that significantly decreases the variability and average of resources spent at concept development without jeopardizing the ability of designers to deliver the project before a specific milestone date, within a preset level of confidence. In the particular example in Fig. 4, the *efficiency zone* corresponds to a set of postponement strategies with a lag varying approximately between 30 to 45 days, for a confidence interval of one standard deviation. The attentive reader may have also well observed that within the efficiency zone the inferior limit of the confidence interval for the level of resources ($\mu_r - \sigma_r$) achieves its lowest point compared to alternative scenarios.

Fig. 5 illustrates that the concept of an efficiency zone tends to be more prominent in the scenarios that assume less learning exists between consecutive iterations. This is a logical result since if the length of the rework loop cycle increases designers will be better off in postponing the execution of the tasks. Fig. 5 also illustrates that design process reliability increases as the postponement lag increases because the number of task iterations decreases. Fig. 6 illustrates the decrease in task iterations and changes that fall after concept development is done (which also cause task iterations) and the simultaneous increase on the number of changes that fall in the postponement lag, as we increase the duration of the lag. The graph shows however that the number of iterations does not decrease steadily but rather fluctuates up and downward to zero along a trend line. Given the simplified model in this work, we foresee such variability as a vicissitude against the implementation of a postponement strategy in complex design processes since it leads to unequal benefits across different tasks regardless what lag gets adopted.

We envisage other potential setbacks against implementation of postponement strategies in practice, if other things are left equal. We opt to mention two we consider most significant. One setback is that we believe few project managers would let their team members get totally involved in other projects afraid they could not get them back when they would need, given the scarcity of skilled resources that managers typically have to cope with. This seems to us a fair concern and will be a challenge for AEC organizations to reassure managers the readiness of resources when they would need them afterwards.

The other setback pertains to the willingness of a client for taking risks in terms of the goals he proposes to achieve. In specific instances, it may be justifiable for a client to let the design team adopt a no postponement strategy regardless of the added costs in terms of resources spent and the loss of process reliability such strategy will cause. For a client faced, for instance, with the curve in Fig. 4, he should let designers adopt a postponement strategy in the efficiency zone, if the decision is made in terms of the upper limit of the confidence interval. But if the client decides in terms of average results designers should opt for a no postponement strategy. The outcome of this decision thus varies significantly according to the risk the client is willing to incur regarding the capability of the designers to meet the contractual obligations the client wants them to meet and designers will ultimately agree upon to get the job.

QUALITY TRANSFORMATION OF AEC ORGANIZATIONS

Practitioners in the AEC industry seldom use set-based design techniques and by and large are skeptical of its potential benefits when challenged with its use in other industries. Clients, on their side, commonly synthesize their needs with the expression “fast, cheaper, and better quality”. Clients are primarily concerned to get the semiconductor fabs done on the milestone dates they strategically set. In addition, clients also demand process flexibility. Process flexibility means clients want the freedom to change the criteria along the design process with the simultaneous reassurance designers will still meet the original milestone dates instead of invoking changes as an argument to justify delays.

Up to present, AEC practitioners have tried to meet target dates by starting the design process as soon as they can. In addition, designers also advocate the design philosophy consisting of embedding flexibility in the product definition during the initial conceptualization effort. Such flexibility aims to accommodate later, unforeseen needs for more capacity from the building systems that serve the cleanroom. (Logically, a need for less capacity would signify an increase on the degree of over design). Practitioners call this strategy “decoupled” design. A “decoupled” design translates in practice in a design process with flexibility to accommodate changes but with a tangible cost up front not all clients may agree to pay (whether or not such cost would prove itself worth in the long term). In contrast, some clients may require designers not to over design but instead develop a design “coupled” to the capacity needs the client foresees initially. In this latter circumstance, if capacity needs change the probability designers will have to rework the

design increases substantially. In this work, we consider designers follow a "coupled" strategy so significant changes in design criteria have an impact the design process.

Our work helps to understand practitioners 's rationale, given that they do not adopt set-based design and learning is limited between iterations. But the strategy commonly adopted in current practice comes with some costs. One cost is the maximization of the number of task iterations designers have to go through and of resources spent, regardless of the designer 's choice in terms of product flexibility. A second cost is the maximization of the variability of total project duration and man-hours spent.

If designers adopt set-based design, a postponement strategy hardly brings any benefits due to the robust nature of set-based design to accommodate changes in design criteria. We recall we modeled set-based design so it minimizes the need to rework the tasks in case criteria changes occur. As a result, set-based design can help designers to still gain the benefits of an early start in terms of total project duration while simultaneously not incurring in an increase of the resources spent. AEC organizations that opt to implement set-based design must however incur on an up front cost. This is the cost of investing in computational systems, and training people to use the systems. AEC organizations (with the involvement of client organizations) should therefore balance between the process and product efficiencies that may result from the use of set-based design against its up front cost.

In a recent project, designers developed three distinct solutions to install a set of tools in the cleanroom because the client had not yet chosen which specific tool vendor was going to pick but needed its install design done once he would make the decision. Designers implicitly used set-based design. Recent web ventures in the AEC industry also show awareness for the opportunities set-based design creates. Evidence thus suggests the quality transformation of AEC organizations—the goal that triggered this work (Tommelein and Ballard 1997)—is in place.

DIRECTIONS FOR FUTURE RESEARCH

This paper does not focus on many other effects of postponement strategies in the design process. Nonetheless, we find worth mentioning two effects resulting from a no postponement strategy typically followed in practice. One is its less tangible cost in terms of the level of satisfaction each designer gets from the work he does. It seems reasonable to us to state that few are the designers that appreciate having to perform frequent passes on the same tasks.

The other effect may come in terms of quality of the product definition. Work in other industries reports that implementation of postponement strategies and set-based design leads to gains in product quality (Iansiti 1995, Ward et al. 1995). Intuitively, these results make sense if extrapolated to the AEC industry. Current practice in fab design results in early commitments on the fab parameters followed by later twists if changes occur. In contrast, we suspect designers could develop product definitions of higher quality if they were to use set-based design or postpone their commitments given the conditions of uncertainty in which the design process evolves. (We preclude a scenario in which designers would over design the fab so much to the extent it would be invulnerable to changes in design criteria because such scenario would be unreasonably costly up front to the client). We hope to pursue research on this avenue in future work.

This paper focused on the design process. A comprehensive study of postponement must however probe on its consequences to construction process. Empirical research indicates opportunities exist for practitioners to selectively start building specific parts of each system while holding off other parts. As a result, the negative impacts of postponement on the total duration of the design process may not be even so significant, if builders are properly involved in the design process. Such will be the subject of forthcoming work.

ACKNOWLEDGEMENTS

This research was funded by grant SBR-9811052 from the National Science Foundation, whose support is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors and do not necessarily reflect the views of the National Science Foundation. Financial support from the Portuguese Foundation of Science and Technology, through a scholarship awarded to Mr. Nuno Gil, is also gratefully acknowledged.

We are most indebted to Robert Kirkendall, senior architect at IDC, who diligently lined up most of our contacts in the fast-paced semiconductor industry. Thanks are also due to Prof. Lee Schruben, at U.C. Berkeley, for having developed SIGMA, a most versatile and useful simulation environment. Last, but not least, we owe thanks to all people interviewed, for the time and knowledge they shared with us.

REFERENCES

- Carley, K. M. (1995). "Computational and Mathematical Organization Theory: Perspective and Directions." *Computational and Mathematical Organizational Theory*, 1 (1), 39-56.
- Clark, K. B., and Fujimoto, T. (1991). *Product Development Performance*. Harvard Business School Press, Boston, Massachusetts.
- Conklin, E. Jeffrey and Weil, William (1997). "Wicked Problems: Naming the Pain in Organizations." Group Decisions Support Systems, Inc. Reprinted with permission on the web at <http://www.3mco.fi/meetingnetwork/readingroom/gdss_wicked.html, Oct. 98>.
- Cooper, R. G. (1990). "Stage-Gate Systems: A New Tool for Managing New Products." *Business Horizons*, May-June, 44-54.
- Crichton, C. (1966). *Interdependence and Uncertainty, A Study of the Building Industry*. Tavistock Publications Limited.
- Eisenhardt, M. K. and Tabrizi, B. N. (1995). "Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry." *Administrative Science Quarterly*, 40, 84-110.
- Ford, D. F., and Sterman, J. D. (1998). "Dynamic Modeling of Product Development Processes." *System Dynamics Review*, 14 (1), 31-68.
- Ford, N. F. and Sterman, J. D. (2000). "Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects." [Working Paper, System Dynamics Group, MIT](#).
- Gebala, D. A., and Eppinger, S. D. (1991). "Methods for Analyzing Design Procedures." *Design Theory and Methodology, DE-Vol. 31*, ASME.
- [Gil and Tommelein \(2000\). "Effects of Postponement Strategies for Managing the Design Process in Conditions of Uncertainty." Technical Report No. 2000-?. Construction Engineering and Management Program, Civil and Environmental Engineering Department, University of California Berkeley, CA.](#)
- Hopp, W. J. and Spearman, M. L. (1996). *Factory physics. Foundations of Manufacturing Management*. The McGraw-Hill Companies, Inc., 668 pp.
- Iansiti, M. (1995). "Shooting the Rapids: Managing Product Development in Turbulent Environments." *California Management Review*, 38 (1), Fall, 37-58.
- Iansiti, M. (1997). "From technological potential to product performance: an empirical analysis." *Research Policy*, 26, 345-365

- Jin, Y., and Levitt, R.E. (1996). "The Virtual Design Team: A Computational Model of Project Organizations." *Computational & Mathematical Organization Theory*, 2 (3), 171-196.
- Law, A. M. and Kelton, W. D.(2000). *Simulation Modeling and Analysis*. The McGraw-Hill Companies, Inc., 760 pp.
- Lin Z., and Hui C. (1997). "Adapting to the Changing Environment: A Theoretical Comparison of Decision Making Proficiency of Lean and Mass Organization Systems." *Computational & Mathematical Organization Theory*, 3 (2), 113-142.
- Lottaz, C., Clement, D., Faltings, B., and Smith, I. (1999). "Constraint-Based Support for Collaboration in Design and Construction." *J. of Computing in Civil Eng.*, 13 (1), 23-35.
- Malone, T.W. (1987). "Modeling Coordination in Organizations and Markets." *Management Science*, 33 (10), 1317-1332.
- McDonald, Chris J. (1998). "The Evolution of Intel's Copy EXACTLY! Technology transfer Method", *Intel Technology Journal*, Q4'98.
- Pietroforte, R. (1997). "Communication and governance in the building process." *Construction Management and Economics*, 15, 71-82.
- Sabbagh, K. (1996). "21 st Century Jet." Ed. Scribner, New York, NY, 366 pp.
- Schruben, D. A. and Schruben, L. W. (1999). *Event Graph Modeling using SIGMA*, Custom Simulations.
- Simon, H. A. (1969). "The Sciences of the Artificial." Massach. Inst. of Tech., (3rd ed 1996).
- Smith, R. P. and Eppinger, S.D. (1997a). "A Predictive Model of Sequential Iteration in Engineering Design." *Management Science*, 43 (8), 1104-1120.
- Smithers, T. (1989). "AI-based Design vs. Geometry-based Design or Why Design Cannot be Supported by Geometry Alone." *CAD*, 21 (3) 141-150.
- Sobek II, D. K., Ward, A. C., and Liker, J. K. (1999). "Toyota's Principles of Set-Based Concurrent Engineering." *Sloan Management Review*, Winter, 67-83.
- Thomke, S. and Reinertsen, D. (1998). "Agile Product Development: Managing Development Flexibility in Uncertain Environments." *California Management Review*, 41 (1), 8-30.
- Tommelein, I. D. and Ballard, G. (1997). "Coordinating Specialists." *Technical Report No. 97-8*. Construction Engineering and Management Program, Civil and Environmental Engineering Department, University of California Berkeley, CA.

Ward, A., Liker, J. K., Cristiano, J. J., and Sobek II, D. K. (1995). "The Second Toyota Paradox: How Delaying Decisions can Make Better Cars Faster." *Sloan Mgt. Review*, Spring, 43-61.

Womack, J.P., Jones, D.T., and Roos, D. (1990). *The machine that changed the world*. Harper Collins, New York, NY, 323 pp.

Wood , S. C. (1997). "The Impact of Tool Delivery Times on the Optimal Capacity and Value of Semiconductor Wafer Fabs." *Proc. of the IEMT Symposium*, Austin, Texas, October 13-15.

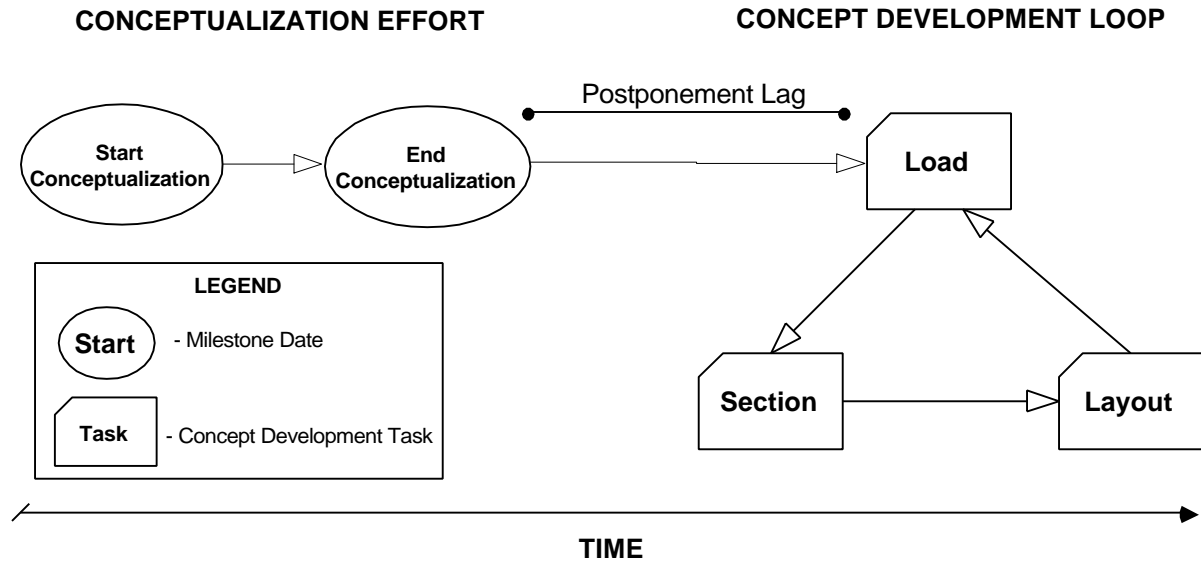


Figure 1 Design Development Process

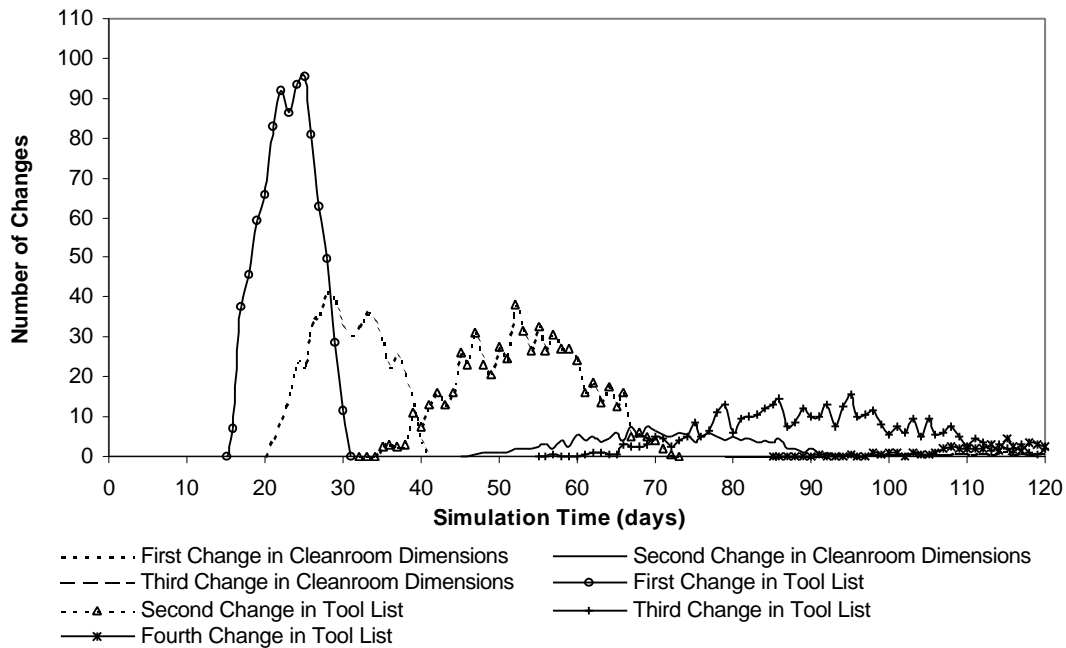


Figure 2 Frequency of Changes in Cleanroom Dimensions and Tool List for 1000 Simulation Runs

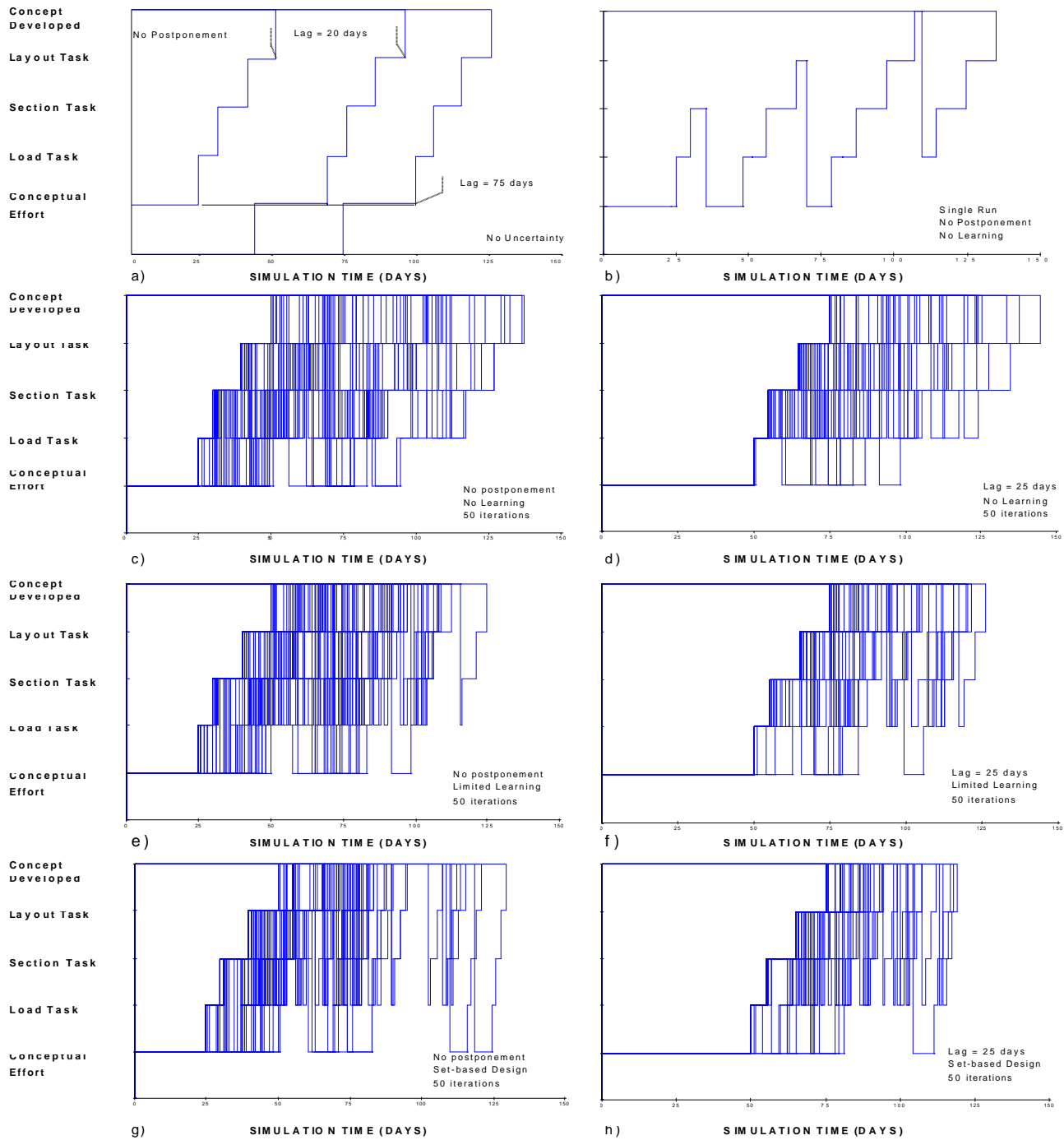


Figure 3 - Instances of Simulation Outputs of Design Task Progression versus Time : (a) No Uncertainty; (b) Single Run; (c) and (d) 50 Runs with No Learning; (e) and (f) 50 Runs with Limited Learning; (g) and (h) 50 Runs with Set-Based Design

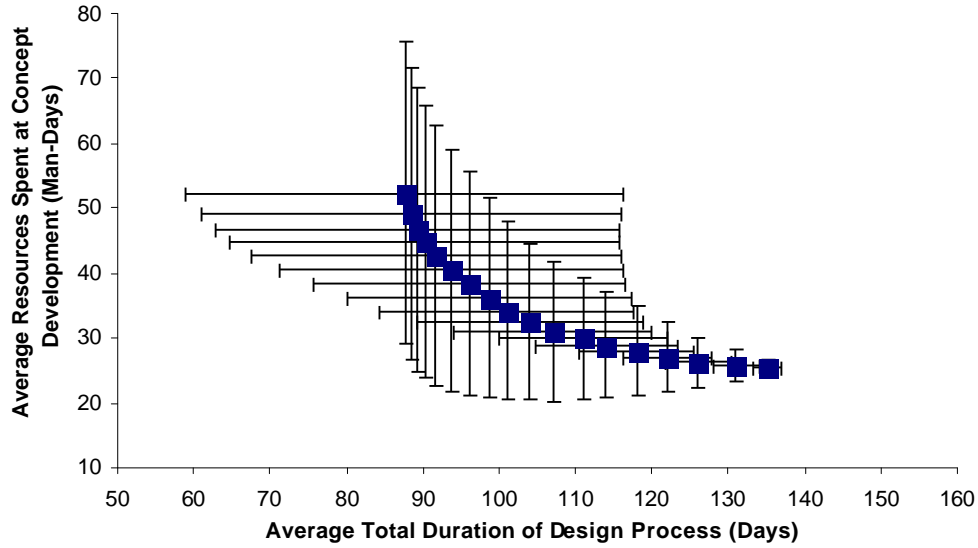


Figure 4 Total Duration of Design Process vs. Resources Spent at Concept Development in Function of Postponement Strategy [Scenario: no learning rework algorithm]

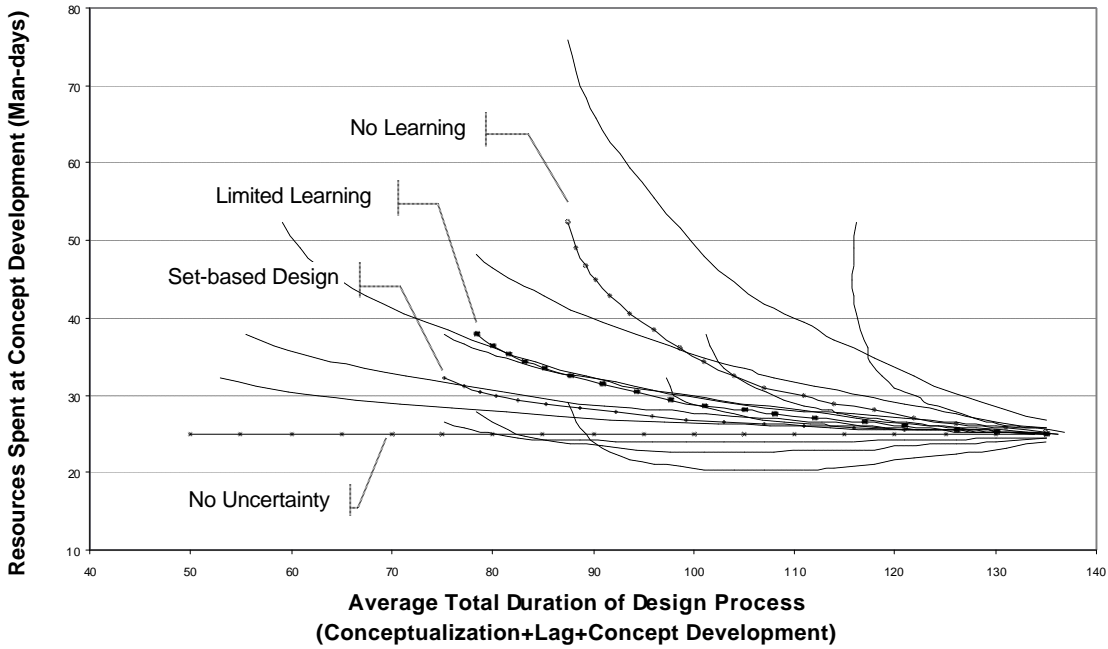


Figure 5 Total Duration of Design Process vs. Resources Spent at Concept Development in Function of Postponement Strategy

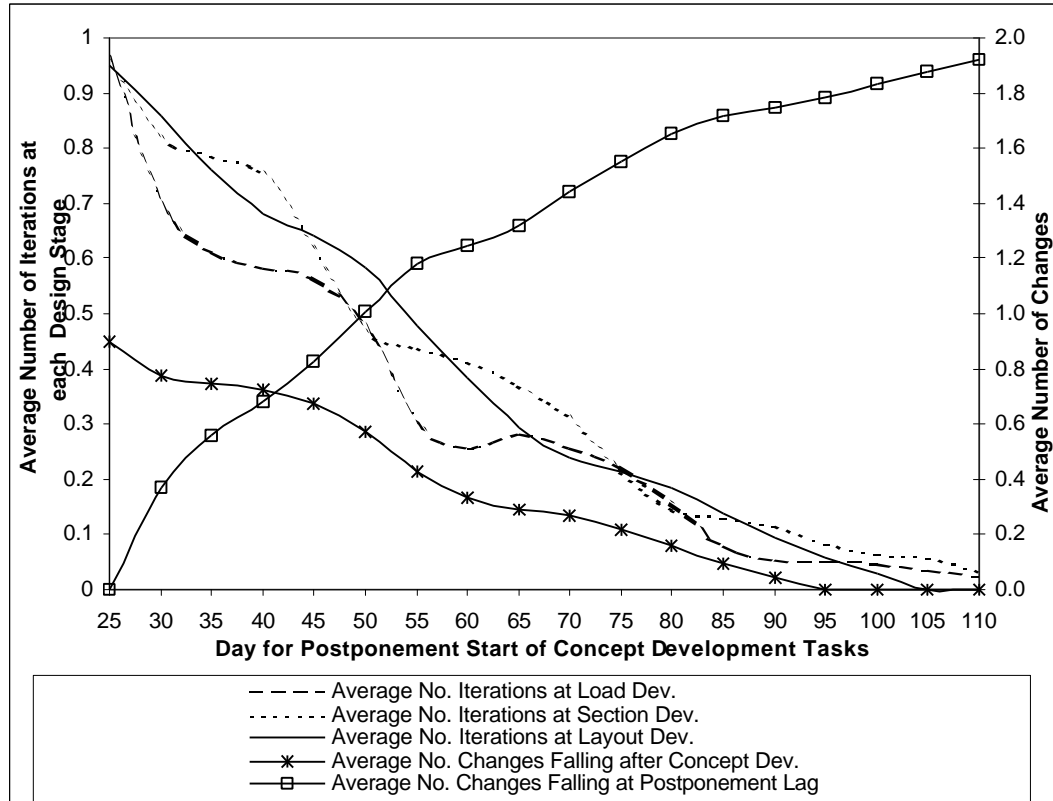


Figure 6 Variation of Average Number of Task Iterations and Changes in Function of Postponement Strategy (scenario: no learning algorithm for rework)