

TECHNICAL RESEARCH REPORT

Controlling Iteration in Product Development Processes

by Jeffrey W. Herrmann

TR 2005-91



ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.

ISR is a permanent institute of the University of Maryland, within the Glenn L. Martin Institute of Technology/A. James Clark School of Engineering. It is a National Science Foundation Engineering Research Center.

Web site <http://www.isr.umd.edu>

Controlling Iteration in Product Development Processes

Jeffrey W. Herrmann

*Department of Mechanical Engineering and Institute for Systems Research,
University of Maryland, College Park, MD 20742, USA*

Abstract

Iteration is a dominant though unwelcome feature of product development. Controlling this iteration can reduce the time needed to complete product development projects. Management must decide how long each project should spend in a design stage before it undergoes a design review (or test). This paper presents a model of iteration that can be used to make this decision. The paper considers the single-project case and two scenarios in which multiple projects require a common reviewer. These results give some insight into how managers should control iteration to improve their product development processes.

Keywords: Engineering design, product development, decision-making.

Please send correspondence to

Dr. Jeffrey W. Herrmann
Department of Mechanical Engineering
University of Maryland
College Park, MD 20742
301.405.5433 TEL
301.314.9477 FAX
jwh2@umd.edu

1 Introduction

Iteration is a dominant though unwelcome feature of product development. For example, in the quick study phase of General Motors' advanced vehicle development process, a great deal of iteration occurs as the design team identifies and solves numerous technical problems. Design reviews are a common cause of iteration in product development processes. In many firms, a product development project must perform a given stage (or task) and then undergo a review before it can go on to the next stage (or task). Despite the efforts of the design team, the project may fail the review. If so, the project must repeat that stage (or task). Controlling this iteration can reduce the time needed to complete product development projects.

If the review is a test to determine how well the design performs, failure can occur due to the design team's limited ability to predict the actual performance of the design. A key limitation is the use of models to estimate design performance (e.g., deflection of a beam, circuit board reliability, machining time, or vehicle maneuverability). These models, based on scientific theory, experimental results, and experience, may be inaccurate or imprecise. Thus, the actual performance of a design will vary from the predicted, and a design that should be adequate will fail. Roser *et al.* (2003) discuss the way that such models lead to engineering changes and the choice of which design change to perform.

If the review is a chance for management to evaluate the design, failure can occur because a manager finds a flaw (e.g., a possible safety hazard) that the design team failed to consider or because the design team made a mistake that they did not correct. (Costa and Sobek, 2003, call this *rework iteration*.) If the review includes feedback from customers, failure can occur because the customers did not initially articulate their true requirements (which are different from the specifications that the design team tried to satisfy). After evaluating and rejecting the design, the customers provide a change in direction. (Costa and Sobek, 2003, call this *behavioral iteration*.)

This paper analyzes a simple model of a product development process in which each project must complete a single stage and pass a subsequent review or else repeat the stage. The purpose of this analysis is to gain insight into this fundamental phenomenon that will help managers control their product development processes. A key feature of the model is that the probability of passing the review is a function of the time spent performing the stage that precedes the review. That is, the design team is never certain that the design is satisfactory.

In our model, the design's quality is measured as the probability that it will pass the review. This increases as the design team works on the design more, finding better solutions, discovering more relevant information, double-checking their analysis, and allowing new issues to appear so that the team can resolve them before the review.

Another key feature of this model is that the review takes time. The particular activities necessary for a review depend upon the type of review, but they include constructing a prototype, planning and conducting tests or experiments, analyzing the results, preparing documents and presentation materials, and holding the review.

This situation presents an important question for management, who must decide how long each project should spend performing the stage that precedes the review. Management's primary goal is to minimize the expected time until the project completes a successful review. Avoiding iteration, while desirable, is not the primary goal since complete certainty may, in some cases, be unachievable. We also consider a risk-averse manager who needs to meet a deadline. Note that a multiple-stage process is a sequence of

single-stage processes. Minimizing the expected time needed for each stage will minimize the expected time needed for the entire process.

This paper extends the basic model to examine product development organizations that conduct multiple product development projects at the same time. This follows the idea that a product development organization is a *decision production system* that, in the same way that a factory produces finished goods, processes information by making decisions. See Herrmann and Schmidt (2002) for more details about this perspective.

Ha & Porteus (1995) study a related though different problem. Their model assumes that a certain amount of design work must be done, but this design work introduces random flaws. Management needs to schedule reviews to find the flaws quickly, since the penalty for a design flaw is a delay needed to repeat the work that was done since the flaw occurred. (The rework introduces no new flaws.) However, reviews take time, so too many reviews delay the project as well. Such a model may be appropriate for a routine design process that has a well-understood amount of work to do and flaws that are straightforward to fix.

The model presented in this paper is more appropriate for innovative design processes that involve creativity and more technical risks. Pich *et al.* (2002) describe a selectionist approach to project management in the face of complexity or an inaccessible environment that makes learning ineffective. In this domain, a design team may choose to try multiple concepts, either in parallel or in sequence. The model presented in this paper can be used to describe this sequential strategy by considering each iteration as a separate attempt to reach the design requirements. A review can confirm that the team has found a satisfactory solution, so having a review quickly is a chance to complete the project as soon as that happens. Thomke (2003) is an excellent discussion of how a firm can learn from experimentation. He presents some case studies of firms that are doing so.

Thomke and Bell (2001) present a model that covers sequential testing and develop a policy for minimizing the total cost of testing and redesign. Unlike Thomke and Bell, this paper focuses more generally on design reviews (which could involve testing) and minimizing the expected time needed to complete a project.

There are other causes of iteration, including the coupling of design tasks on different subsystems or components that are not independent (see, for example, Smith and Eppinger, 1997). A subsystem or component will be considered multiple times as the design process moves from conceptual to detailed design (called *design iteration* by Costa and Sobek, 2003).

The remainder of this paper is organized as follows. Section 2 presents the model of the product development organization. Section 3 analyzes the single-project case. Section 4 discusses the multiple-project case when there is an external arrival process. Section 5 discusses the multiple-project case when the number of active projects is held constant. Section 6 concludes the paper.

2 The Basic Model

A project is an instance of a design team creating a new design by following the product development process. We assume that the product development process has a single stage (the design stage), followed by a review (the design review) that determines whether the design at that point is satisfactory. If so, the project is done. If not, the project must repeat the design stage and return to the design review. The project must repeat this loop until the project passes the review.

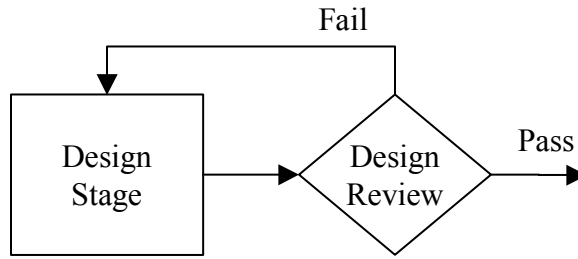


Figure 1. The Basic Model.

Let t denote the time spent on the design stage, which is a decision variable. Let r denote the time needed to conduct a design review (this is given). Let $p(t)$ be the probability of passing the review, which depends upon the time spent on the design stage. Intuitively, it is reasonable that $p(t)$ is an increasing function that approaches 1 in the limit. Thus, we assume the following shape for this function (illustrated in Figure 2):

$$p(t) = 1 - e^{-at}, \text{ for } t > 0.$$

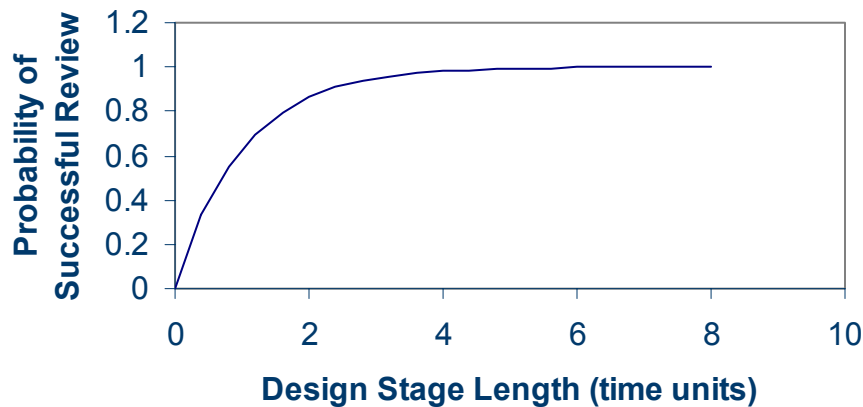


Figure 2. Graph of probability of successful review version time of design stage.

We assume that, if the project fails the design review and the design team has to repeat the design stage, the probability of success increases in the same way that it did the first time. Certainly, the probability of success must return to zero at the beginning of the design stage (since the existing design is known to be bad). It is conceivable that the probability of success could increase faster depending upon how much work was completed in the previous stage. This model we leave for future analysis, since it would increase the number of parameters in the model, and the basic model is a reasonable and practical approximation.

Furthermore, we assume that the project spends the same amount of time on the design stage each time it is performed. This is reasonable since, given that the project must perform the design stage, completing the project does not depend upon how many times the project has previously performed the design stage. That is, there is a memoryless property. The expected time needed to complete the project is independent of how long the project has taken to date. Thus, the policy should remain the same each time.

3 Single Project: Minimize Expected Duration

Given t and $p(t)$, the number of iterations for a project has a geometric distribution. The expected number of iterations (including the successful one at the end) is $1/p(t)$.

The total time for a project from its start to its completion is $t + r$ times the number of iterations. Thus, $T(t) = (t + r)/p(t)$ is the expected duration of the project. This convex function has a single optimal solution at the point t^* that satisfies $e^{at^*} - at^* = ar + 1$. Given a review time r , minimizing the expected duration of a project requires choosing the optimal time t^* for the design stage. Choosing t too small leads to a large number of iterations, while choosing t too large leads to long stages. Figure 3 shows the shape of $T(t)$ for different values of r .

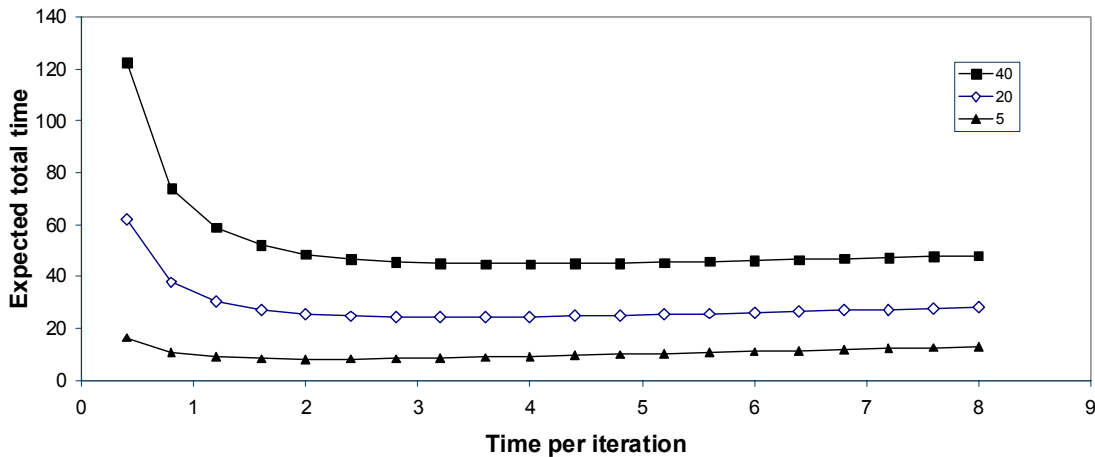


Figure 3. Graph of expected total time versus time per iteration for different values of review time.

Consider now the impact of the review time r . As r approaches 0, t^* and $T(t^*)$ approach 0. As r increases, t^* increases and $T(t^*)$ increases. In some sense, the design stage t is similar to a batch size that is used to avoid setups (in this case, reviews). Instead of worrying only about the optimal batch size (in this case, t^*), it also makes sense to reduce the setup cost (the review time r), which allows smaller batches (quicker reviews) and improves the overall performance.

This result, for an innovative design process, is quite different than that of a more routine design project where a design team can develop a design that will certainly pass the review (or test). Consider a different function for the probability of review success. Suppose that there is a bound on the time needed for the design stage and the probability of review success equals 1 if this amount of time is spent. Moreover, probability of review success is a linear function of t . Let D be the upper bound, and the probability $p(t) = t/D$. Then, the expected total time $T(t) = (t + r)/p(t) = (t + r)D/t = D(1 + r/t)$, which reaches its minimum at time $t^* = D$, independent of the review time r . The optimal expected total time is $T(t^*) = D + r$, and there is no uncertainty about the value (since $p(D) = 1$). This shows that iteration is not helpful for this type of design scenario.

Risk aversion also affects the optimal policy, as the next section discusses.

4 Single Project: a Risk-averse Manager

Managers are notoriously risk averse. A risk averse manager worries more about the variability of project duration than the expected duration. If a design stage has a deadline, the risk-averse manager is most

concerned about maximizing the probability that the project is completed by a deadline. (Ideally, this probability will equal one.)

Let C be the deadline for the design stage to pass the review (this is given). Let $P(t)$ be the probability of meeting this deadline. A single iteration requires $t + r$ time units. Thus, the project can go through at most $k = \text{floor}(C/(t + r))$ iterations (including the last, successful one). (Recall that $\text{floor}(x)$ is the largest integer less than or equal to x .) Note that a short design stage means that the project could go through multiple iterations, increasing the probability of meeting the deadline above $p(t)$.

Let X be the number of iterations required to pass the review. This includes the iterations that failed the review (if any) and the last, successful iteration. X has a geometric distribution, with a probability of success equal to $p(t)$. Thus, the manager wishes to choose t to maximize the following probability:

$$P(t) = P\{X \leq k\} = 1 - (1 - p(t))^k$$

If $p(t) = 1 - e^{-at}$, for $t > 0$, then $P(t)$ can be expressed as follows:

$$P(t) = 1 - e^{-kat}$$

The function kat is discontinuous due to the jumps in the floor function of k . There is a set of critical points at which $C/(t + r)$ is an integer. Let $m = \text{floor}(C/r)$. Then, define t_1, t_2, \dots, t_m as follows:

$$t_i = C/i - r$$

(Note that these values get smaller as i increases.) The value t_i is the most time that can be spent in the design stage to accomplish i iterations before the deadline. If $t_{i+1} < t \leq t_i$, then $k = i$. Since i increases within that range (and we want kat large to maximize $P(t)$), t_1, t_2, \dots, t_m are the only values of t that need to be considered. If $t = t_i$, then $kat = ia(C/i - r) = a(C - ri)$. This decreases as i increases, so $t^* = t_1 = C - r$ maximizes $P(t)$:

$$P(t^*) = 1 - e^{-a(C-r)}$$

Thus, a deadline that produces risk aversion also implies that iteration is not helpful. Figure 4 illustrates this for a specific example with $C = 2.5$ time units.

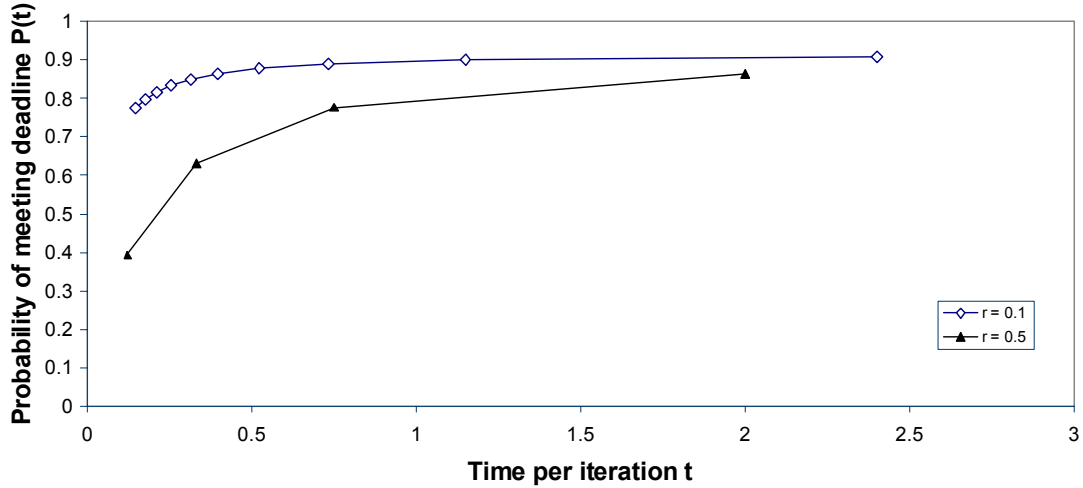


Figure 4. Probability of meeting deadline versus time per iteration for different values of r .

5 Multiple Projects: External Arrivals

Consider an organization in which new product development projects arrive with an arrival rate of λ projects per time unit. We assume that each project has its own resources, so the design stages of different projects can operate concurrently and independently. The design reviews, however, require the same unique resource (called the *reviewer*). This may be a test facility, a manager, or an executive review board.

A key consideration is the utilization of the reviewer. We assume that $\lambda r < 1$ so that the reviewer has enough capacity to review each project once. If all projects have the same constant a and perform the design stage for t time units, then the utilization u of the reviewer equals $\lambda r/p(t)$. Since u must be less than one, this provides a lower bound on $p(t)$ and on t . The system will be stable if and only if $p(t) > \lambda r$.

With multiple projects contending for the same resource, a project may have to wait for the reviewer after performing the design stage, due to variability in the times that projects are ready for reviews and in the review times. Let $q(t)$ be the average queue time for the reviewer. This depends upon the utilization of the reviewer, and we assume that the average queue time is proportional to $ur/(1-u)$. Such a term is common in queueing models. As the utilization approaches 0, this term approaches 0. As the utilization approaches 1, this term increases without bound. The constant k incorporates the degree of variability in the system.

$$q(t) = kur/(1-u) = k\lambda r^2/(p(t) - \lambda r).$$

Figure 5 illustrates the impact of time per iteration on the queue time.

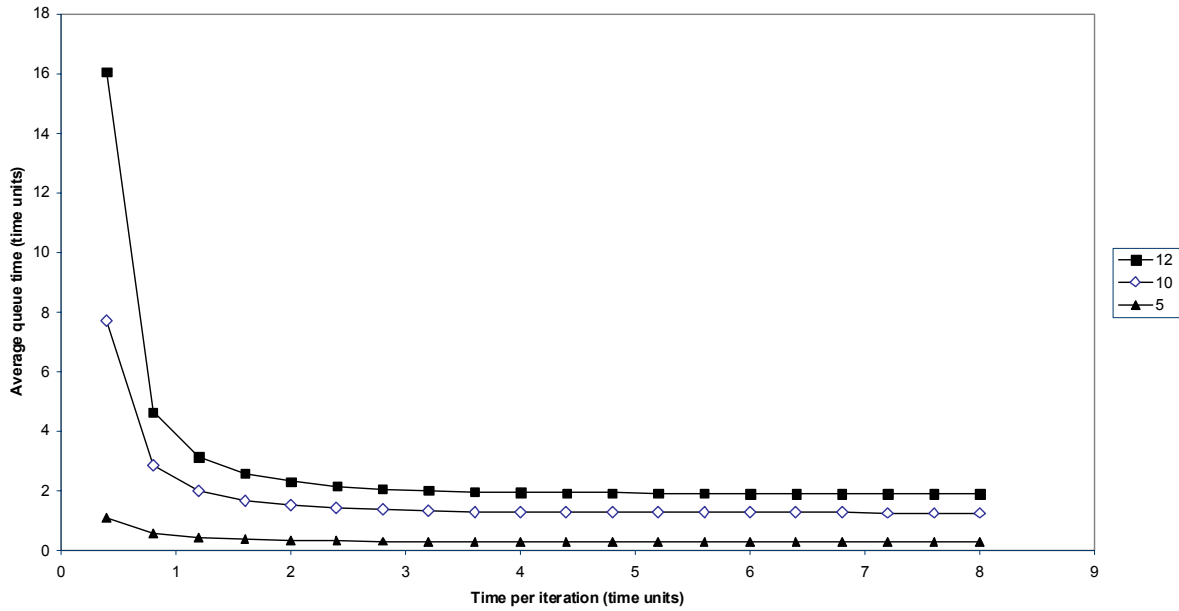


Figure 5. Queue time versus time per iteration for different values of r .

We assume that no more work is done while the project is waiting for the reviewer. The average length of each iteration is $t + q(t) + r$. The expected duration of a project is $T(t) = (t + q(t) + r)/p(t)$. Figure 6 illustrates the impact of time per iteration on $T(t)$. If t is too short, there is excessive queueing, so the expected duration is great. Increasing t too much increases the expected duration above its minimal value.

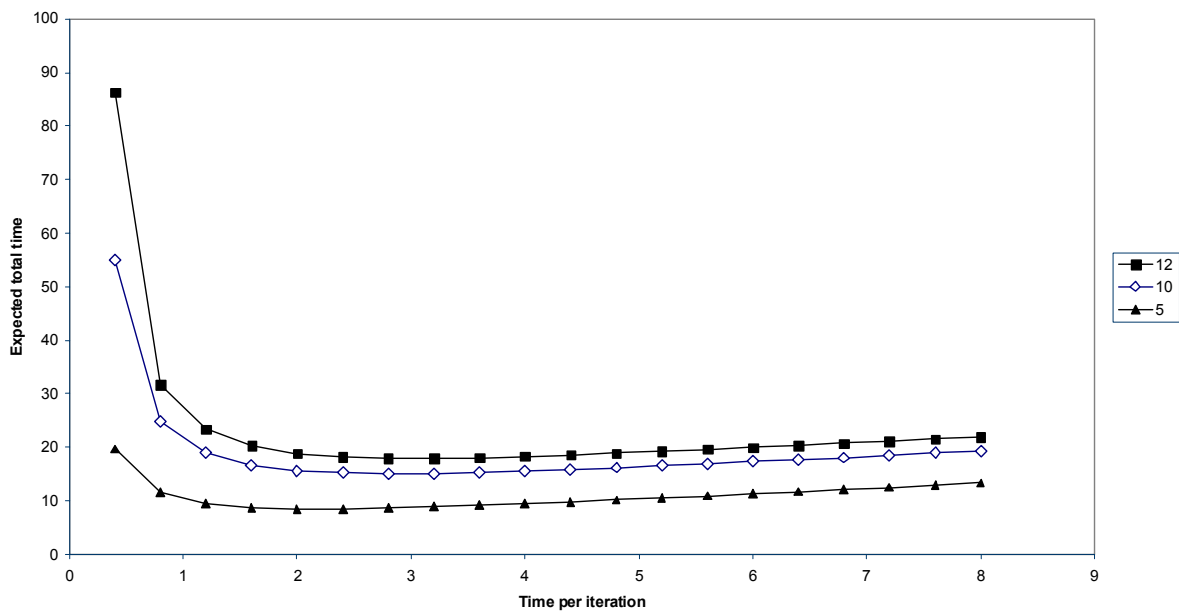


Figure 6. Expected total time versus time per iteration for different values of r .

6 Constant Number of Projects

We now consider the case where the product development organization has limited resources. Thus, it can conduct a limited number of projects simultaneously. If the firm has a backlog of project ideas, then a new project begins when and only when an existing one completes. Let N denote the number of projects that are active. The amount of time spent in each design stage affects not only the expected duration of a project but also the rate at which the organization completes projects (according to Little's Law). This is similar to a production line that operates under a CONWIP production control policy: for a given WIP limit, reducing the cycle time increases the throughput (see, for instance, Hopp and Spearman, 2001).

If the review time is constant, then, between two reviews of the same project, the other $N-1$ projects will be reviewed, which requires $(N-1)r$ amount of time. If $t < (N-1)r$, then each project will have to wait for $q(N) = (N-1)r - t$ time units. The length of each iteration equals Nr . The expected duration of a project is $T = Nr/p(t)$, and the expected completion rate is $N/T = p(t)/r$ projects per time unit.

Since $p(t)$ is an increasing function, t should be at least $(N-1)r$. If $t = (N-1)r$, then the reviewer utilization $u = 1$ and the length of each iteration equals Nr . The expected duration of a project is $T = Nr/p((N-1)r)$, and the expected completion rate is $N/T = p((N-1)r)/r$ projects per time unit.

If t is larger than $(N-1)r$, then all of the other projects have had time to be reviewed, so the project doesn't have to wait for the reviewer. The length of each iteration equals $t + r$. The reviewer's utilization $u = Nr/(t + r)$. The expected duration of a project is $T(t) = (t + r)/p(t)$, and the expected completion rate is $N/T(t) = Np(t)/(t + r)$ projects per time unit.

Note that the above model will be useful if the review time is random but has little variability. If the review time is random and has significant variability, then we can analyze this system as a single-server queueing system with a finite calling population (Hall, 1991). The following analysis is exact if the review time is exponentially distributed.

Let the state of the system be n , the number of projects undergoing or waiting for review. n can range from 0 to N . If we let h denote the intensity, then $h = Nr/t$. Let $P(n)$ denote the steady-state probability of the system being in state n . Let $z(n)$ equal the following quantity:

$$z(n) = N!/(N-n)! (r/t)^n$$

$$P(0) = (1 + h + z(2) + \dots + z(N))^{-1}$$

$$P(1) = h P(0)$$

$$P(n) = z(n) P(0), \text{ for } n = 2, \dots, N.$$

The reviewer's utilization $u = 1 - P(0)$.

From these state probabilities, we can calculate the expected queue length L_q as follows:

$$L_q = P(2) + 2P(3) + \dots + (N-1)P(N).$$

The average rate at which the reviewer completes reviews is the reviewer utilization multiplied by the service rate: $(1 - P(0))(1/r)$. The rate at which projects complete equals the rate of successful reviews,

which is $p(t)(1-P(0))(1/r)$. Figure 7 shows the how the review time variability affects the project completion rate when $r = 0.5$, $N = 6$, and $a = 1$.

By Little's Law, the expected time in queue $q(N) = rL_q/(1-P(0))$. The expected length of each iteration equals $t + q(N) + r$. The expected duration of a project is $T(t) = (t + q(N) + r)/p(t)$.

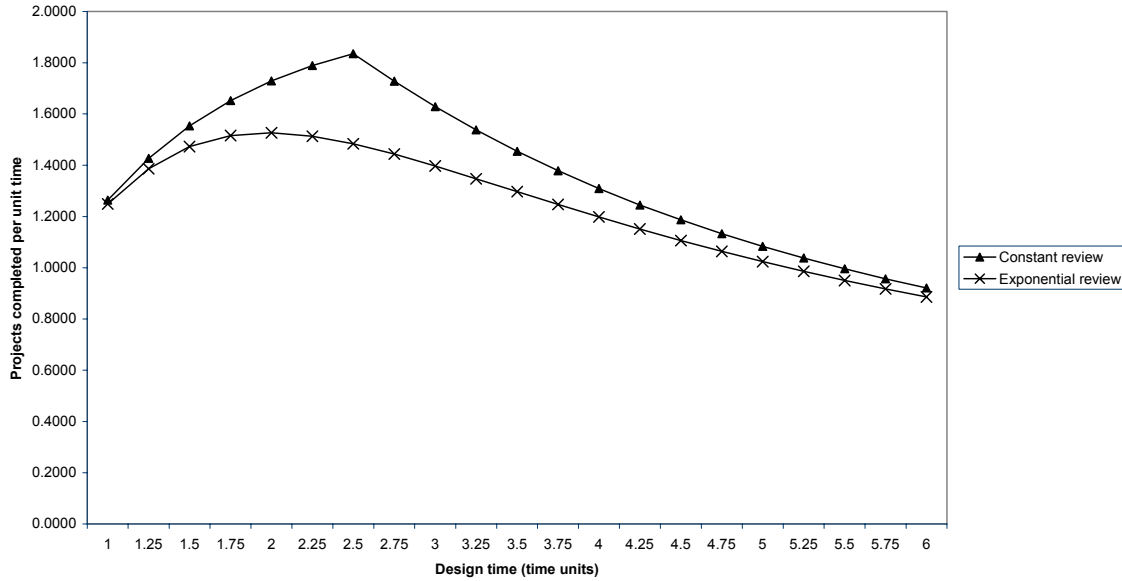


Figure 7. Project completion rate versus time per iteration for constant and exponentially distributed review times.

7 Summary and Conclusions

This paper presents a deliberately simple model of a product development process with a design review in order to obtain some useful insights into the issue of controlling iteration in product development. In particular, to minimize the expected duration of a project that involves creativity and more technical risks, managers should accept that a successful review cannot be guaranteed. Performing more iterations can reduce the expected total time of the project. Also, reducing the time needed to perform a review (or test) has a significant impact on the expect duration. Not only is there less review time per iteration, reducing the review time makes shorter design stages (more frequent iteration) desirable.

However, if the design problem is routine, then iteration does not reduce the expected total time of the project. The team should finish the design stage to get a design that will certainly pass. Reducing the time needed to perform a review (or test) has less impact on the expect duration in this case.

Iterations do add variability to the project duration. In a creative design process, a risk-averse manager who wishes to meet a deadline should avoid iteration, since it does not increase the probability of meeting the deadline. Also, reducing the time needed to perform a review (or test) has a significant impact on this probability, since it means more time to perform the design stage.

The models of systems with multiple projects show how the time spent on the design stage affects the overall system performance. The manager must carefully select the design time in order to reduce the expected total time (for systems with external arrivals) or the rate of project completion (for systems with a constant number of active projects).

There are many more issues regarding iteration and design reviews that are not considered in this model, and the model could be extended in many ways. For instance, different projects may have different probability-time curves. The probability-time curves may vary based on the number of iterations already completed. Research on methods for eliciting the probability-time curves from managers and engineers and data about previous projects would be a useful step towards providing a tool for helping managers make a more informed decision.

Acknowledgements

The author appreciates the insights and help provided by Linda Schmidt, Joseph Donndelinger, my students, and many other collaborators. This material is based upon work supported by the National Science Foundation under grant number 0225863.

Bibliography

- Costa, Ramon, and Durward K. Sobek II, "Iteration in engineering design: inherent and unavoidable or product of choices made?" DETC2003/DTM-48662, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Illinois, September 2-6, 2003.
- Ha, Albert Y., and Evan L. Porteus, "Optimal timing of reviews in concurrent design for manufacturability," *Management Science*, Volume 41, Number 9, pages 1431-1447, 1995.
- Hall, Randolph W., *Queueing Methods: for Services and Manufacturing*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- Herrmann, Jeffrey W., and Linda C. Schmidt, "Viewing Product Development as a Decision Production System," DETC2002/DTM-34030, Proceedings of the 14th International Conference on Design Theory and Methodology Conference, ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Canada, September 29 - October 2, 2002.
- Hopp, Wallace J., and Mark L. Spearman, *Factory Physics*, Irwin/McGraw-Hill, Boston, 2001.
- Krishnan, V., and Karl T. Ulrich, "Product development decisions: a review of the literature," *Management Science*, Volume 47, Number 1, pages 1-21, 2001.
- Pich, Michael T., Christoph H. Loch, and Arnoud De Meyer, "On uncertainty, ambiguity, and complexity in project management," *Management Science*, Volume 48 Number 8, pages 1008-1023, 2002.
- Roser, Christoph, David Kazmer, and James Rinderle, "An economic design change method," *Journal of Mechanical Design*, Volume 125, pages 233-239, 2003.
- Smith, Robert P., and Steven D. Eppinger, "Identifying controlling features of engineering design iteration," *Management Science*, Volume 43, Number 3, pages 276-293, 1997.
- Thomke, Stefan, and David E. Bell, "Sequential testing in product development," *Management Science*, Volume 47, Number 2, pages 308-323, 2001.
- Thomke, Stefan, *Experimentation Matters*, Harvard Business School Press, Boston, 2003.