Modeling Design Rework In A Product Development Process

by

Matthew F. Bromberg
Bachelor of Arts in Physics, University of California, Berkeley, 1992

Submitted to the Sloan School of Management and the Department of Mechanical Engineering in partial fulfillment of the requirements for the degrees of

Masters of Science in Management
and
Masters of Science in Mechanical Engineering

In conjunction with the Leaders for Manufacturing Program
at the Massachusetts Institute of Technology
May 2000
June 2000

Signature of Author ____________
Sloan School of Management
Department of Mechanical Engineering

Certified by ____________
Stephen C. Graves
Abraham J. Siegel Professor of Management Science and Engineering Systems
Thesis Supervisor

Certified by ____________
Steven D. Eppinger
Associate Professor of Management Science and Engineering Systems
Thesis Supervisor

Accepted by ____________
Margaret Andrews
Director of Master's Program
Sloan School of Management

Accepted by ____________
Ain A. Sonin
Chairman, Committee on Graduate Students
Department of Mechanical Engineering

Modeling Design Rework In A Product Development Process

by

Matthew F. Bromberg

Submitted to the Sloan School of Management and the Department of Mechanical Engineering in partial fulfillment of the requirements for the degrees of Masters of Science in Management and Masters of Science in Mechanical Engineering

ABSTRACT

Managing the product development process is of vital concern to corporations. A critical aspect of product development that negatively impacts program cost and timing is rework. Unfortunately, in large organizations with successive development cycles, the product, process and organizational complexity preclude simple solutions. Even given sufficient data, many organizations do not understand what constitutes good and bad performance relative to rework. Through research at General Motors Truck Product Group, a model was developed that forecasts expected total rework. The model assumes rework is a function of: 1) The product portfolio and timing; 2) The complexity of each product program; 3) The pattern of rework over time for product programs; 4) The "lifecycle age" of each product program. The model has four potential uses: A) To aid in portfolio/project planning; B) To provide a rework performance baseline for management; C) To evaluate initiatives with regards to their impact on design rework; D) To identify leverage targets for management attention and improvement.

Thesis Supervisor: Stephen C. Graves
Title: Abraham J. Siegel Professor of Management Science and Engineering Systems

Thesis Supervisor: Steven D. Eppinger
Title: Associate Professor of Management Science and Engineering Systems

Introduction

Competition for faster product life-cycles has motivated companies to "optimize" product development processes. In the automotive industry, efforts to accelerate product development are complicated by product complexity (a typical production vehicle may have more than 5000 parts and subassemblies), and by the successive product development cycles (auto manufacturers are striving towards fourteen product launches per year). A significant roadblock to efficient product development is design rework.

Rework has many definitions, but it usually refers to situations in which work is repeated due to a design error or a product change. In contrast, within the development process, there are often planned periods of work repetition. For instance, marketing and engineering will go back and forth reconciling product requirements with engineering capabilities and cost. Because this is an essential product development technique, we consider this planned work repetition good for the product and worth the time and cost. We distinguish it from rework by using the term *design iteration*. This paper is concerned with rework, which we define as any change to a part/configuration or as any adjustment affecting a supplier or assembly process that occurs after a release date.[1] We purposely kept our definition of rework broad so that it encompasses as much information as possible. We also observed that any attempt to narrow the definition, say by excluding supplier changes, potentially underestimates rework and fails to deliver a clear performance picture. Figure 1 shows a schematic representation of the product development process and illustrates the difference between design iteration cycles and design rework situations.

[1] The release date is a management tool used to schedule the design process. It is determined by the portfolio planning group and is based on process bottlenecks and marketing concerns.

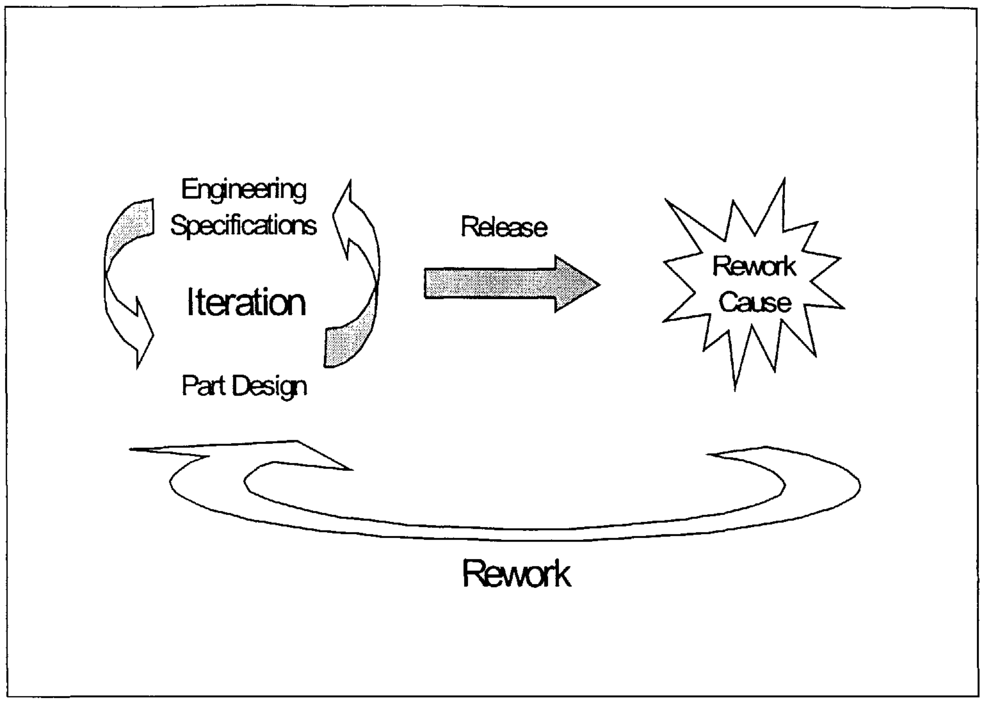

Figure 1 Schematic Representation of Design Iteration and Rework

There are many *causes* of rework. In fact, as product and process complexity increases beyond that of a simple object, the number of potential causes increases exponentially. Unfortunately, sifting through a myriad of causes may be untenable. However, a more tractable way of looking at rework is to identify *factors* that greatly influence the amount of rework for a given product. We distinguish causes from factors such that causes are tactical while factors are strategic. In other terms, factors are at a higher level of abstraction than causes.[2] For instance, a rework cause might be that a design problem was not caught with computer-aided analytical tests, but it was discovered in a later stage prototype. On the other hand, a rework factor may be that the timing of destructive testing relative to prototype build cycles determines overall product rework. Another way to the understand the difference is to observe that factors are more systemic and suitable for an analytical predictive model while causes are less suitable for simulation because they often evoke debate and controversy. In the course of this research,

[2] The concept of varying the level of abstraction to provide better insight is championed by Shoji Shiba in A New American TQM. See references.

numerous causes were uncovered through data analysis and interviews. A more detailed discussion of rework causes is included in **Appendix A**.

Our research uncovered four factors that explained a majority of observed rework even when looking at different vehicle models and platforms. While each factor by itself is not descriptive enough for management decisions, combining them together may provide an accurate and useful model. The first factor is the number and timing of product development programs (known as a product portfolio). The second factor is the relationship between product complexity and product rework. The third factor is the distribution of rework over time; we found that this distribution is determined by the management process and is independent of the product type. The last factor is the age of the product platform. In other words, a vehicle built on a mature platform will exhibit less rework than one built on a new platform. Since effective product development requires effective management of rework for the whole organization, it is necessary to know when the organization is performing well and when it is performing poorly. The rework model developed in this paper provides such insight. The remainder of this paper is organized as follows:

Section 1 motivates the problem, provides definitions and surveys related literature.

Section 2 presents factors that impact rework in a product development process.

Section 3 motivates and then describes the rework management model.

Section 4 offers suggestions for extending the model.

Section 5 concludes, discusses the project's potential impact, and offers final suggestions for minimizing rework based on the model.

Appendices amplify the research, the data and the model.

1.1 Rework in the Vehicle Development Process

At the time of this research, General Motors Company employed a 24-month vehicle development process (VDP). The VDP is under constant refinement including attempts to compress individual steps, to eliminate steps through employment of computer aided tools, and to overlap steps such that development is accelerated. A simplified representation of the VDP is provided in Figure 2. The design iterations and rework detailed in Figure 1 occur during the later half of the VDP.

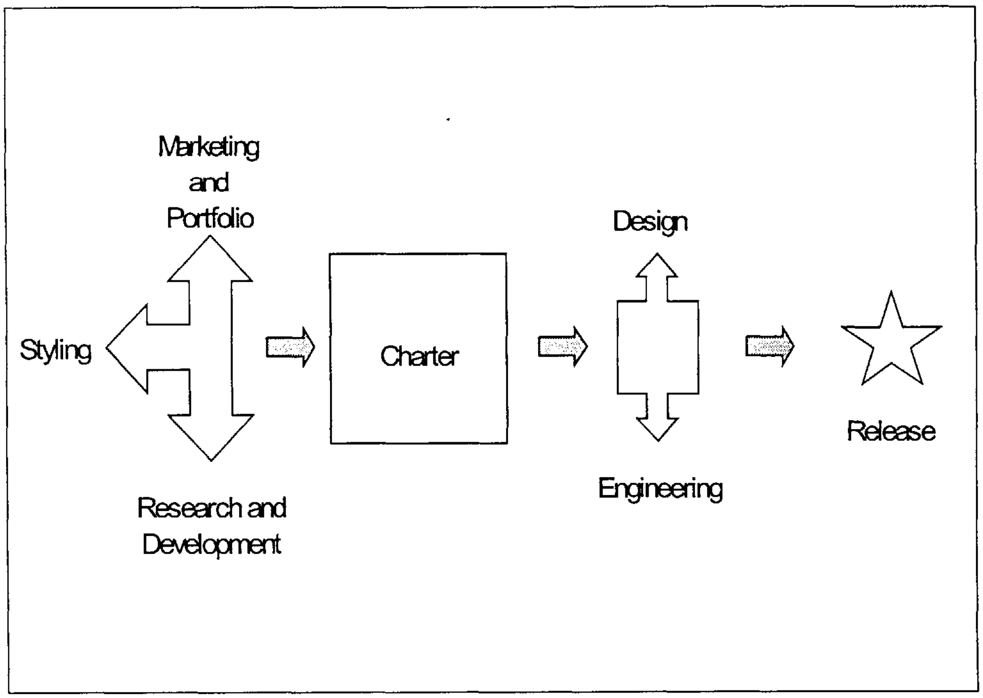

Figure 2 Schematic Representation of the Vehicle Development Process

During the period of design iterations, engineering and design learning occurs, and design, engineering and assembly limitations are reconciled. After this iterative development, parts are *released* for production. In other words, construction of production tools is started. In practice, parts are released on a staggered basis depending on prototyping and production lead-time requirements. Any changes after this part release milestone are considered rework, and are handled through a process and document called an Engineering Work Order (EWO).[3] These documents have a myriad of purposes including but not limited to tracking work progress, obtaining approval, and notifying affected departments within the organization. Lastly, because the VDP is in a constant state of flux, no two programs follow exactly the same process. While this supports continuous improvement, the effect of so much change on organizational learning and progress is unknown.

In tackling the rework problem, the first question we asked was how much rework is occurring? While a simple question, there is no simple answer. Tracking different metrics can yield entirely different results. For instance, if we analyze the number of EWOs (the number of documents), we see a fairly constant trend over the past few years, whereas counting the number of reworked parts shows statistically significant variation. As an example, Figure 3 shows a graph of reworked parts over time.

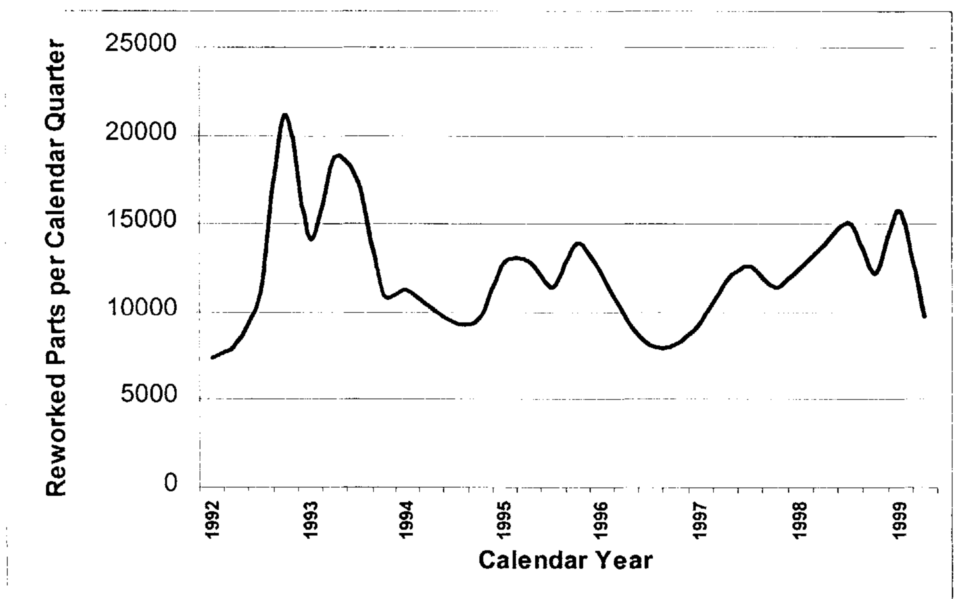

Figure 3 GM Truck Group Engineering Rework

[3] Other organizations refer to these as Engineering Change Notices (ECN).

However, looking at this graph, we asked a more important question: how are we doing? Unfortunately, looking at this graph, we cannot ascertain whether we are doing well or poorly.[4]

1.2 Rework Metrics

These questions first led to an investigation of rework metrics. While we held the premise that process management relies on metrics, identifying the appropriate ones is easier said than done. In an ideal product development scenario, one would track the root cause, the direct costs (such as part/tooling and design hours), the indirect costs (such as inventory depletion), and the impact costs (such as quality and time effects, and the potential to cause the rework of additional parts). We may also want to know the impact on designers, engineers, planners, human resources, and other functions. Just as important, the metrics must reflect the nature of the product and the process. For instance, do products with more parts take more design effort? Unfortunately, such metrics are seldom available. See **Appendix B** for a discussion of the difficulties of defining and maintaining effective metrics. Our objective in this work was to identify what currently collected data (or synthesis of currently collected data) could be used as an effective metric for total system rework. Based on the discussion of **Appendix B**, we identified three basic requirements for our metric:

(a) The metric must be related to the process and to the product.

(b) The metric must be easy to understand.

(c) The metric must be easy to collect, and the collection process must be disciplined.[5]

[4] Data Collection in a complex process is difficult. Throughout the study, we were careful to maintain representative data while at the same time eliminate sets that were grossly inconsistent due to anomalies with the database or with the vehicle program. For instance, data from a vehicle program that was cancelled would have been eliminated from the study because it is not representative of a typical product nor of the process. Likewise, changes to the data collection systems required some adjustments. Lastly, the database used for the research starts in 1992. This results in the initial ramp in rework shown in Figure 3. The ramp is an artifact of the data collection process, not a performance indicator.

[5] A disciplined collection process is one that not only has minimal statistical variance, but also has the understanding and support of the engineers. This support helps to minimize erroneous data entry.

1.3 A Useful Rework Model

Attempting to evaluate how well or poorly the organization is performing with respect to rework motivates the development of a rework model. Ideally, the model would provide insight to the following three questions about rework:

(a) How much rework will occur for a given product?

(b) When will this rework occur during the product development process?

(c) How much will this rework impact the organization in terms of cost, effort and time?

All will be addressed in **Section 3**. Next, we review the relevant research literature.

1.4 Literature Review

There are four key research areas related to this paper. As a starting point, product development and design engineering publications analyze the foundation of engineering design management. By examining these underlying principles and methodologies, we may better understand why current problems exist. The second key area is, of course, design iterations. While the majority of literature focuses on planned design iterations, these papers still provide insight into causes and factors of design rework. Third, there is a growing interest in engineering change management. A couple papers were found to provide useful introductions to terms and approaches although practical solutions are scarce. Finally, there is a wealth of literature about complex system modeling. While many of the best references come from studies conducted outside product development, looking for similarities in complex systems, and then adopting appropriate techniques can shed insight.

Literature on product development and engineering design is extensive and often fairly general, but we found two texts particularly useful. An excellent starting point is Joseph E. Shigley's seminal text, Engineering Design. Although somewhat dated as it was published in 1963, this text succinctly defines the design problem in section six. Shigley is one of the first to propose design optimization—the process of planning and automating development steps to arrive at the optimum solution with regards to product performance, speed and cost. Shigley believed that further advances in computers and optimization routines would continue to benefit engineering design. Extending from this belief, engineers and managers have spent the past four decades striving for this design utopia in which design iterations, both planned and unplanned, are minimized. A more modern review of product development is Product Design and Development by Ulrich and Eppinger. In addition to presenting a tractable development methodology, the text summarizes much of what has been done in the field of product development, explains current technologies and includes an extensive reference section. Another overview can be found in the recent article, "A Survey of Design Philosophies, Models, Methods and Systems," by Evbuomwan, Sivaloganathan and Jebb. The authors summarize the past four decades of design methods and theories, and then define terms and chart developments in the field. Lastly, the product development consulting firm Pittiglio Rabin Todd & McGrath (PRTM), has numerous resources and references available at no charge on their web site (http://www.prtm.com). Notably, they complete an annual benchmarking study (PRTM 1997 Product Development Benchmarking Study) which surveys industry accomplishments and trends.

Design iteration is a fairly recent topic for product development research, and most of the work has focused on planned iterations as opposed to unplanned iterations. While the work on planned iterations is not exactly the topic of interest, reviewing these papers provides insight into engineering design, complex systems modeling, and change management. Furthermore, some authors conjecture that better management of planned iteration cycles will lead to fewer unplanned iterations. Kulkarni completed an excellent summary of iteration literature in his report, "Controlling Rework in the Vehicle Development Process." He finds that product development research has identified three key areas for rework management: task scheduling and batching, task sequencing, and change management. In the area of task sequencing, considerable work has been done developing design structure matrices (DSM). These tools help identify and plan key development iteration areas to improve process management. There are several papers that develop and build DSM techniques, but a key introduction to DSM methodologies can be found in the paper by Smith and Eppinger: "Identifying Controlling Features of Engineering Design Iteration." As a next step, the paper, "A Model-Based Method for Organizing Tasks in Product Development," by Eppinger, Whitney, Smith and Gebala uses the DSM tool to streamline a development program and then create a management strategy.

Engineering change management is becoming as complex as the underlying product development processes. Literature topics range from process overviews to clarifying appropriate metrics. In particular, Mitchell Fleischer's short paper, "Where are engineering metrics?" codifies the difficulties in accurately measuring aspects of engineering design and change management. A more thorough description of the problem along with some suggestions was presented by C. Terwiesch and C. Loch in "Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development." Their paper analyzes the engineering change systems with respect to strategies for lead time reduction. Lastly, Robert Kaplan and Robin Cooper present a thorough introduction to integrated metrics in their book, Cost & Effect: Using Integrated Cost Systems to Drive Profitability and Performance. While the book is not about product development or change management in particular, they identify several important principles in establishing administrative and managerial systems.

There is exhaustive literature on parametric modeling of complex processes. While there is much written in product development, we found it beneficial to look outside mechanical engineering for generic modeling methodologies. J. Sterman has conducted excellent work in the field of System Dynamics. His recent book, Business Dynamics: Systems Thinking and Modeling for a Complex World, is a comprehensive introduction to systems modeling, and includes numerous case study illustrations. Complex system modeling is also explored in Sterman's key paper, "Learning In and About Complex-Systems." In this paper, the author emphasizes that models, in order to be effective, must be built from a box of tools that not only teaches us about complex systems, but also allows us to model those systems in a manner that improves the decision making process.

2.1 Rework Factors

A key to developing the rework model presented in **Section 3** was to identify quantifiable relationships between various *factors* and rework. Numerous product-related, process-driven and organizationally determined factors influence rework. Conducting a literature review, analyzing data, and performing a series of interviews will yield an endless list. Based on our research, we hypothesized the existence of many such factors. However, four factors stood the test of data and common sense, and we focused on them.[6] The four factors presented as hypotheses and later supported by data are:

1. The number and timing of product programs is a key determinant of rework.
2. Product complexity drives rework.
3. Rework occurs according to a consistent distribution independent of product type.
4. The "lifecycle age" of a product platform affects the amount of rework.

Appendix C presents a discussion of other factors.

2.2 Metric and Data Used in Analysis

In this study, the metric for rework was Part Count. This refers to the number of parts listed on the EWO and includes all parts affected by a specific rework case. At General Motors, any post-release change to the product, configuration, supply chain or assembly process is initiated through an EWO, and is quantified by part count in addition to other metrics. A couple examples will help illustrate the metric. If a frame requires redesign, an EWO is issued that details all affected frames; part count refers to this quantity. As another example, consider an EWO initiated in response to a supplier that cannot build a part as initially contracted. Due to the supplier's changes, General Motors is forced to alter several adjoining assemblies. Part count quantifies all affected parts regardless of the cause or complexity of changes.

Up front, we want to stress that the metric choice was not trivial, and in fact was based on numerous hypotheses and trials. There were many possibilities, but part count was the

[6] Throughout the paper, the terms factor and driver are used interchangeably. Some of the factors are clearly rework drivers, while others are merely observed behaviors. Using factors, drivers and observed behaviors together is one of the unique attributes of this methodology, and it is not our intent to distort definitions or classifications.

one metric that fulfilled our requirements and yielded the insights of this research. To justify this metric, we looked at data records of all EWOs from the past six years. Recalling the discussion in **Section 1.2** and in **Appendix B**, this metric has the following advantages: it is simple to understand and directly related to the end product, and it is already in use for other purposes. While all organizations differ, most development processes track individual components, so this, if not some variation, may likely be used. Unfortunately, the metric does not reflect the nature or cost of the change, nor does it reflect the type of part affected. In the following sections, we introduce the four factors hypothesized above. All four factors are quantified with the metric part count.

2.3 The Product Portfolio Drives Rework

The portfolio mix is one of the key determinants of overall engineering rework. The portfolio mix refers to the sum total of vehicles under development. For instance, the group might be working on three model years of two small vehicles, three model years of two medium vehicles, and four model years of large vehicles. While the portfolio consists of only three distinct programs, the portfolio is made up of ten different vehicles. Programs and model years are chosen to fulfill marketing strategies and engineering development needs as determined by the corporation's strategy board. Recall that the demands of the strategy board are tempered with development process constraints such as prototyping and testing capacity, or engineer and designer availability. Figure 4 is an example of a portfolio matrix. Each platform or program is listed on the left, along with a brief description of the vehicle type. The rest of the matrix shows the different model years and their respective launch dates. This matrix will be explained further later in the paper.

Program		Complexity	1995	1996	1997	1998	1999	2000	2001
			Model Year						
			Launch Date						
1	truck	Medium	Aug-94	Aug-95	Aug-96	Sep-97	Sep-98	Jul-99	Oct-00
2	truck	Small	Aug-94	Aug-95	Aug-96	Aug-97	Aug-98	Aug-99	Aug-00
3	truck	Small	Aug-94	Aug-95	Aug-96	Aug-97	Aug-98	Aug-99	Aug-00
4	truck	Small	Aug-94	Aug-95	Aug-96	Aug-97	Aug-98	Aug-99	Jan-01
5	truck	Medium	Aug-94	Aug-95	Aug-96	Aug-97	Jun-98	Aug-99	Aug-00
6	truck	Medium					Aug-98	Aug-99	Aug-00
9	truck	Medium	Dec-94	Aug-95	Aug-96	Aug-97	Aug-98	Aug-99	Aug-00
23	truck	Large	Aug-94	May-95	Aug-96	Aug-97	Aug-98	Aug-99	Aug-00
24	truck	Large	Aug-94	Aug-95	Aug-96	Aug-97	Aug-98	Aug-99	Nov-00
25	truck	Large	Aug-94	Aug-95	Aug-96	Aug-97	Aug-98	Aug-99	Nov-00

Figure 4 Sample GM Truck Group Portfolio Matrix

2.4 Complexity Drives Rework

Complexity occurs on many levels ranging from the sheer size of a product to the functional interrelations of components. It is necessary to simplify these issues and identify a clear definition. Based on interviews and on a study of the internal organization, we identified three vehicle classifications of complexity. While this is a simplification, and inevitably some vehicles may be inappropriately grouped, it provides a tractable framework. More importantly, it is a starting point. For each vehicle, the number of *production parts* was tallied, and then each vehicle was grouped according to three easily differentiated categories: high (or large) complexity, medium complexity and low (or small) complexity.[7] Production parts are defined as the number of parts actually used in a final vehicle. Figure 5 illustrates the different part categories and their relative quantities in a vehicle development process. While the figure is qualitative, it shows the relative quantities of parts that occur during development. This variance motivated us to focus on one category throughout the research. Based on current improvement initiatives and data collection processes, we focused on *production parts*. For each complexity group, an average and a range were determined.

[7] The high complexity group is comprised of GM's medium duty trucks; the medium complexity group is equivalent to GM's full size trucks, and the low complexity group is equivalent to GM's mid-size trucks.

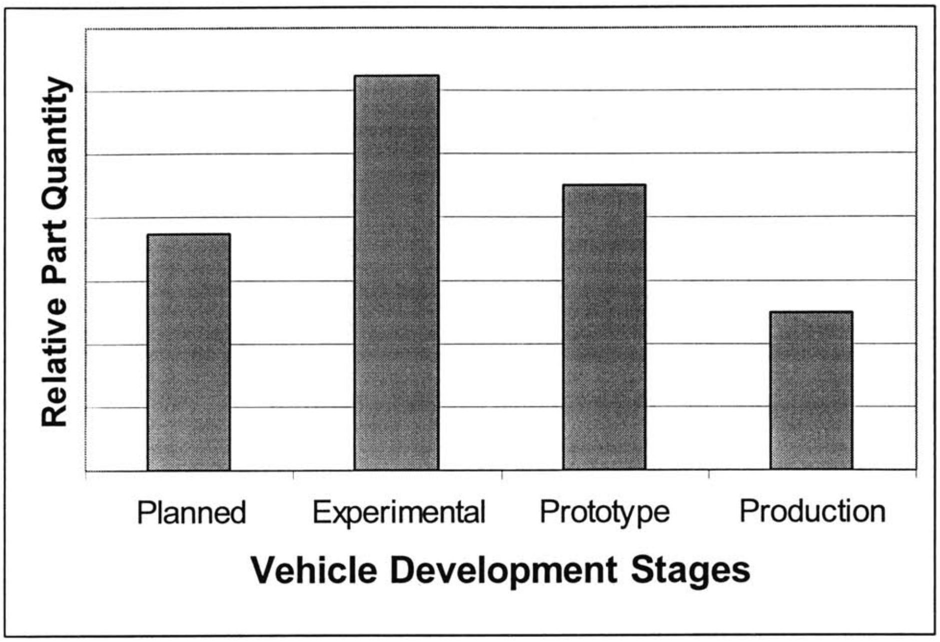

Figure 5 Comparison of Part Quantities During Product Development

Theoretically, a key driver in rework should be product complexity. While this is intuitively obvious, representing complexity with a metric is not trivial. Moreover, showing a consistent relationship between complexity and rework is inherently difficult due to data variance. In Figure 6, we show a graph of rework as a function of production parts. From this, we observe two phenomena: first, on average, each part will be designed, released and then reworked once, if not more, during the remainder of vehicle development; second, rework increases more than linearly with an increase in product complexity. The data used to generate this graph was the set of four platforms for which we found new platform year data. We could not use later development years because this data is influenced by other factors.

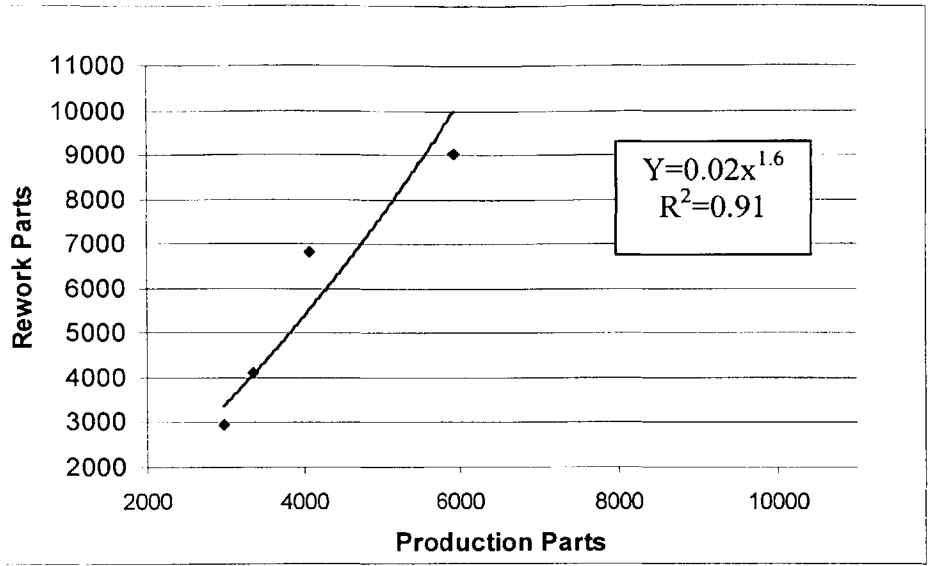

Figure 6 Relationship Between Complexity and Rework[8]

2.5 Experience Reduces Rework

Many applications of learning curves have been identified in manufacturing environments. Notable is Boeing's use of the learning curve to price airplanes. The learning curve gives a relationship between marginal effort and cumulative effort. For production, the cost of each unit decreases as cumulative production increases. So, if the first plane cost $250 Million, the hundredth plane may cost $175 Million. In Boeing's case, they may competitively price the planes at $200 Million. While there are many sources of this reduction, when the effect is aggregated, it is difficult and unnecessary to drill down the detail.

Automotive manufacturing uses platforms to improve development costs and quality. A platform is a basic vehicle architecture that is used as a skeleton for several different products. Extending the learning curve phenomena to product development, we conjecture that the design rework should decrease as the platform matures. If we measure platform maturity by looking at successive development years, we find a consistent 15% annual reduction in the number of reworked parts. The decrease in

[8] Complexity is measured by Production Parts. The graph is based on the first development year of four separate platforms. The equation is the best fit approximation for the data.

rework is caused by numerous factors including learning on behalf of the engineer and the designer, reduction of parts through design refinement, and an increase in the ratio of carryover parts to total parts.[9] The graph in Figure 7 shows this behavior for six platforms, and we observe that the 15% annual reduction is consistent in the long-run even though in the short-term, individual models might experience an increase in rework over the previous year. This annual variation is due to specific model year changes.

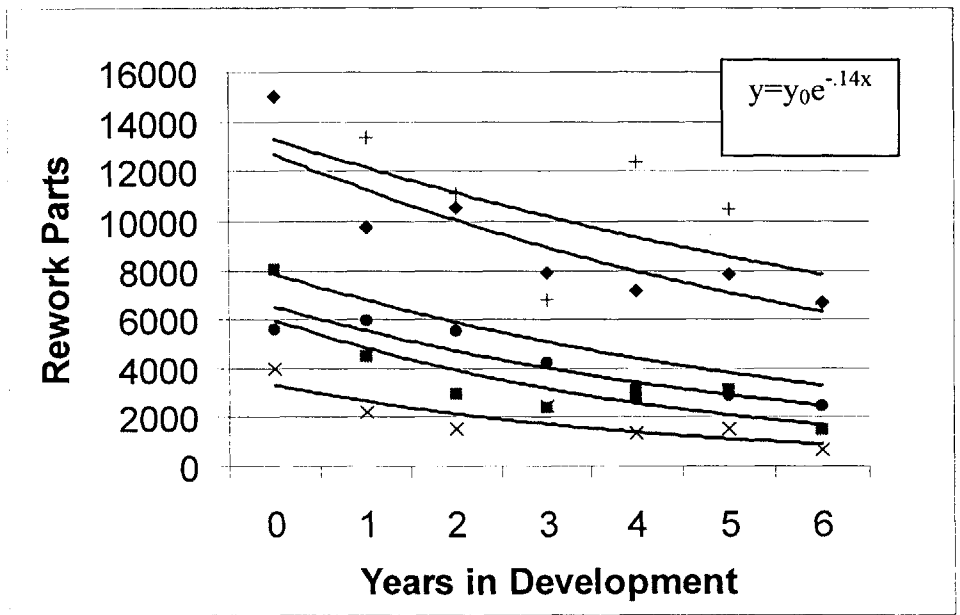

Figure 7 Six GM Truck Programs in which Experience Reduces Rework[10]

The starting point for each curve in Figure 7 depends on the respective platform's complexity. The trend lines clearly indicate the similarity of their behavior.

2.6 Rework Occurs According to a Distribution

Accepting that rework occurring during the vehicle development process is distributed in a manner that depends on the product and the process, we can speculate over the shape of the distribution and whether the bulk of rework occurs early or late in the development

---

[9] Carryover parts are parts that are reused from model year to model year. The metric, considered a measure of part robustness, is commonly used to compare automobile companies.

[10] The data samples are from six unique platforms. The equation is a best fit regression calculated with a simple non-linear optimization routine. The y_0 is the initial level of rework (the first year a platform is in use), the x term represents the number of years the platform is in development (x=0 is the first year, x=1 is the second year, etc...).

process.[11] In the study at General Motors, the distribution shown in Figure 8 was uncovered.[12]

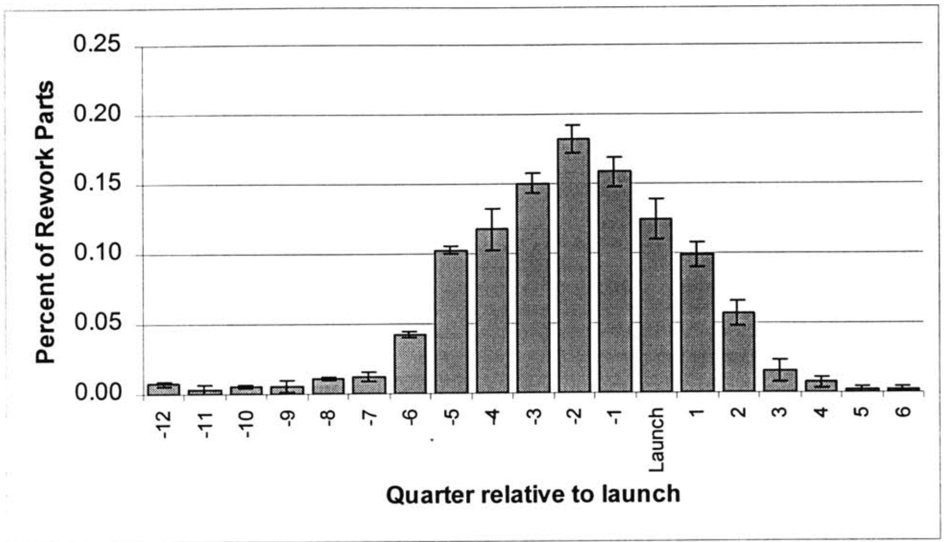

Figure 8 Distribution of Rework Relative to Vehicle Launch Quarter

In addition to the consistency of this distribution, it is noteworthy that nearly 30% of the rework occurs after product launch during a period lasting one year.

[11] See "Frontloading: Shortening Development Time at Toyota Through Intensive Upfront Effort," by Tadaaki Jagawa.

[12] To generate this distribution, over thirty completed vehicle programs were sampled. Each vehicle's rework was plotted over time relative to its launch date. To normalize the samples, each was then converted to a percentage of rework. The distribution was calculated as the average of all the trials. It is very consistent.

3.1 Engineering Rework Model

So far, we have introduced four factors that greatly influence the rework occurring during General Motors Truck vehicle development process. Each factor has been hypothesized, explained, and insofar as possible, supported with evidence. In what follows, the four factors will be combined into the rework model. As purported in the introduction, using the model's objective output, we can do four things:

A. Assist the portfolio planning group in scheduling and gating vehicle development.

B. Evaluate the performance of the organization with respect to rework by providing a baseline of performance.

C. Evaluate group initiatives. Considerable resources are expended in improvement activities intended to reduce rework. Unfortunately, there is seldom an objective way to evaluate progress. Now we can identify when a program performs better than expected, and if it does, we may use that knowledge to make changes across the organization.

D. Identify leverage areas. The debate will no longer reside with the metrics, and will instead focus on the problem.

3.2 Model Inputs

The key input to the model is the product portfolio matrix. This matrix includes every product either under development or planned. Recall, to keep the model simple, the products were classified into three complexity groups: high, medium and low. It is also essential to know the product platform's history. For instance, if the product is based on a three-year old platform, we need to include the previous products so that we capture the experience effect. We also want to know each product's launch date. This allows us to overlap distributions for multiple products based on the launch calendar. An example matrix was shown in Figure 4. In this product portfolio matrix, each vehicle is listed with its complexity group, launch date and platform age. In this example, all vehicles except for program six were based on platforms launched in calendar year 1995.[13] To clarify,

[13] The distinction between model year and calendar year can be confusing. In order to build an accurate model, the distinction must be clear.

look at 2000 model year vehicles: program six trucks are based on a two-year old platform, while program four trucks are built on an older, six-year platform.

Also included in the portfolio information is a description of annual product changes. For instance, sometimes a platform might undergo major redesign, while in others, it might experience minor, non-structural changes. There are numerous degrees of change, so a simplification is in order. We classify vehicles as being based on either a new platform, a platform undergoing minor change, or a platform undergoing major change. Based on these categories, two factors were calculated from historical data. For major changes, the increase factor is approximately 1.4, while for minor changes, the factor is approximately 1.0. This factor is easily incorporated into the model as will be shown later. To model the effect of organizational experience, we use a constant annual reduction term. Based on Figure 7, the reduction is approximately 15%.[14] The final key input is the rework distribution shown in Figure 8.

3.3 Model Algorithm and Formulation

The model relies on a simple algorithm. First, for each product type, the complexity is determined based on our complexity groupings. For instance, if the vehicle is highly complex, the vehicle is assumed to have 6500 production parts. Second, based on complexity, the amount of rework is determined for an all-new version of the product. Third, if the product is based on an aged platform, the amount of rework is reduced by the experience factor one time for each year of platform age. Fourth, depending on the nature of the model-year change (i.e., minor versus major), the amount of rework is adjusted by the change multiple (1.0 and 1.4 respectively). Fifth, the overall amount of rework is then distributed in time according to the rework distribution. Finally, the contributions of each program are summed. A simple example will help clarify the model's algorithm.

The example is shown graphically in Figure 9. Say we have a single product—a medium sized, multi-optioned truck. We assume that this vehicle is built on a two-year-old

[14] A simple non-linear program was used to determine the variable that best fits all curves in **Figure 7**.

platform, and we further assume that the vehicle only has minor changes from the previous year. We want to estimate the associated rework. The first step is to classify the product's complexity, so assume the vehicle is of medium complexity. Based on this, we approximate the final production parts at 4500. The second step is to determine the amount of rework that will likely occur. Given the exponential relationship between complexity and rework, we use the curve of Figure 6 to estimate that this 4500 part vehicle will have 6500 reworked parts during its development. But recall that this vehicle is built on two-year-old platform. The experience curves tells us that the organization improves 15% per year, so the total amount of rework is reduced to 5525. Now, to determine the impact of minor versus major changes, we simply multiply by the appropriate adjustment factor. In this case, given that the changes are minor, the factor is 1.0, and the amount of rework is unchanged. Lastly, we want to find out when the rework occurs. Using the distribution curve of Figure 8, we can determine what percentage of rework occurs in each quarter relative to the vehicle's launch date. For the launch quarter, the distribution leads us to expect 12% of the total rework. For our example, that results in 663 reworked parts.

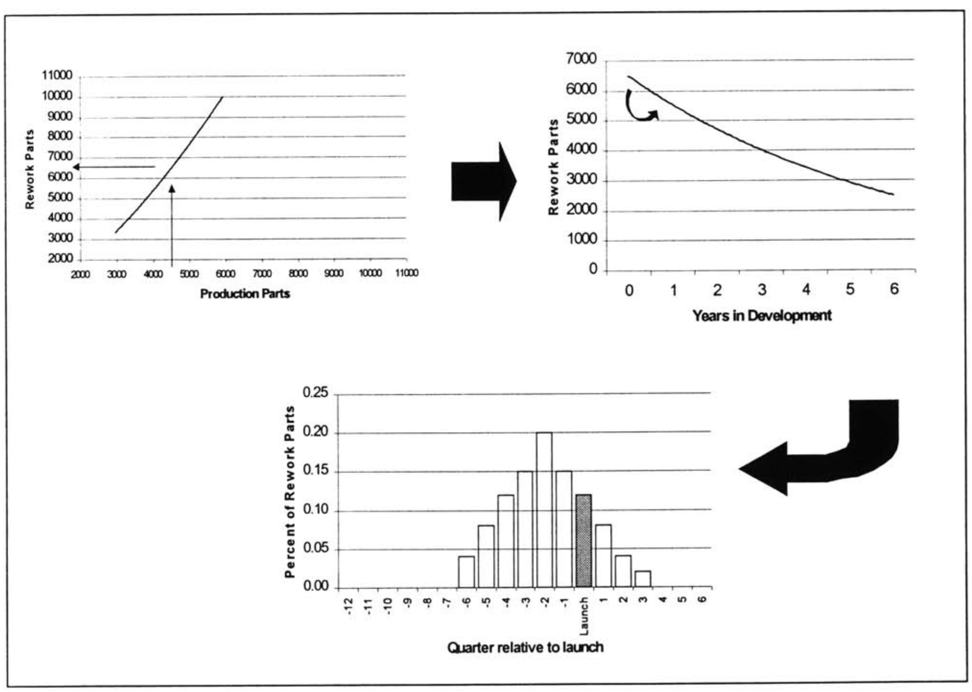

Figure 9 Illustration of Rework Model

This single vehicle example illustrates how the model functions. From here, it is trivial to extend the calculation to a multi-product scenario. The algorithm calculates each product in the portfolio mix separately as above. Then the model aggregates the rework by adding the rework distributions according to calendar date.

3.4 Model Output

The model output is a graph showing rework over time for the entire portfolio. Rework, of course, is measured by part count. In Figure 10 we see the model's prediction of rework given the actual product portfolio worked at GM Truck Group.

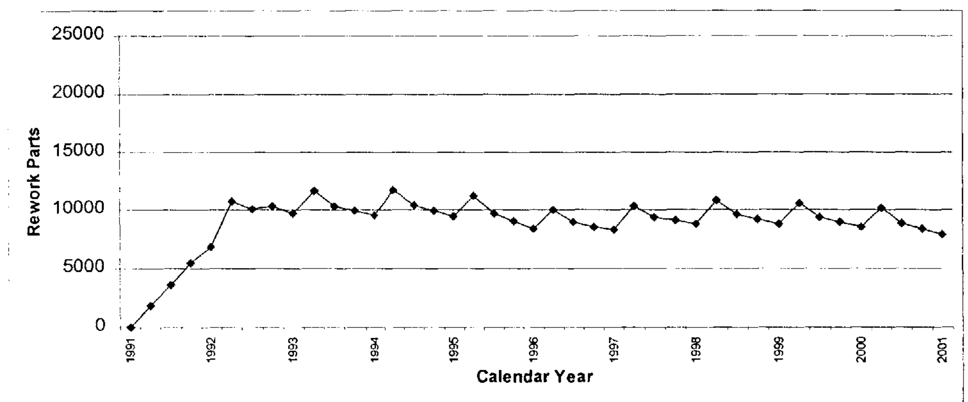

Figure 10 Sample Output of Rework Model—Entire GM Truck Group Portfolio

The seasonal trend seen in the output graph is a manifestation of using identical distributions for each product and of using common vehicle launch dates. In **Section 4.1**, model extensions are discussed—some of which will likely smooth the output. After running the model, we used it to compare predicted rework to actual rework. Recall the graph of Figure 3 in which we see the organizational rework for the past eight years. We could not determine if the performance was good or bad because we could not decipher the peaks and valleys. Using the model, we revisited the graph by superimposing the model's output on top of the actual data. This is shown Figure 11.

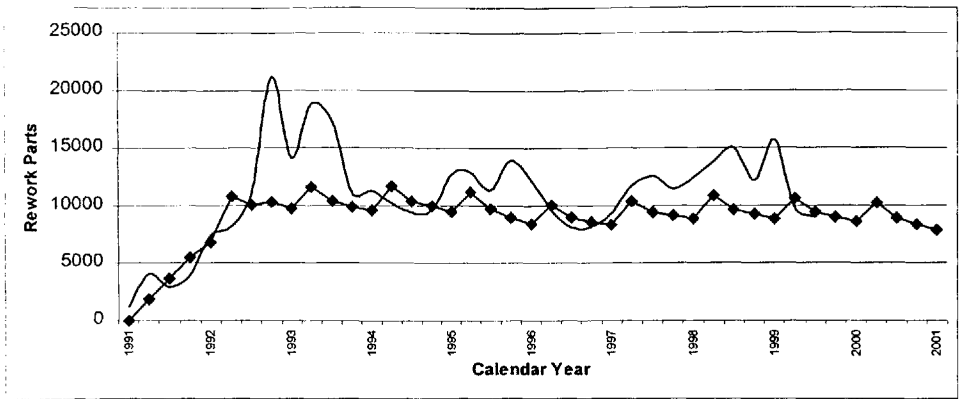

Figure 11 Comparison of GM Truck Group Rework to Rework Model

The graph shows that the model's prediction fluctuates around the low points of the actual data. More importantly, there are three major peaks: 1992-1993, 1995-1996 and 1998-1999. Armed with this information, we investigated the peaks. Without hesitation, and without much disagreement, managers and engineers explained the 1992-1993 and 1998-1999 peaks. In 1992, there were new emission regulations that required significant rework of all vehicles under development. In 1998, GMC developed an all-new truck platform for the first time in over 10 years. While there was no consensus on the 1995-1996 peak, the following section on uncertainty explains how we handled it.

If we want to know the rework prediction for a specific vehicle, we adjust the portfolio matrix. For instance, allowing the user to toggle vehicle programs individually or by groups allows us to look at various levels of detail. We can then compare the subset to actual data. In Figure 12, we see a comparision of model output to actual data looking only at medium complexity vehicles.

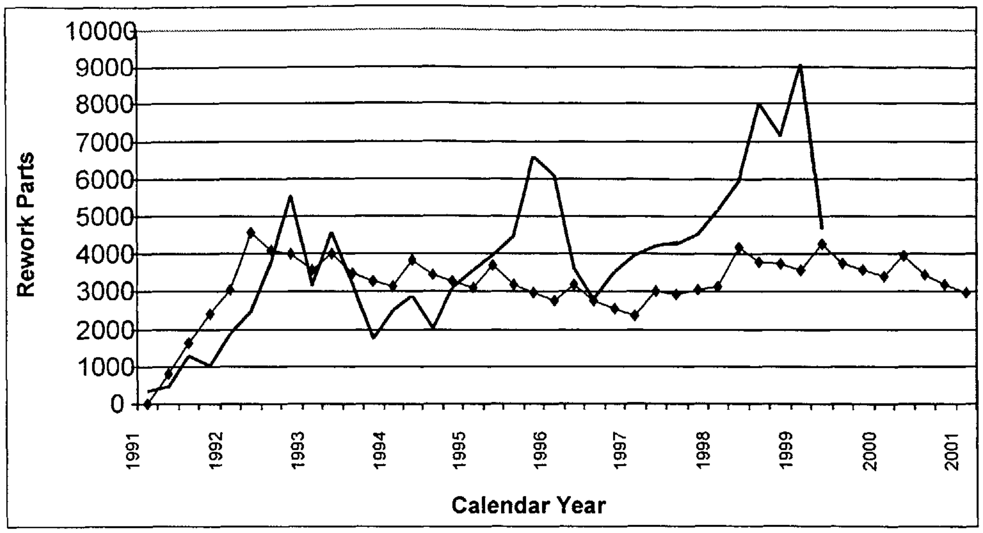

Figure 12 GM Truck Group Rework for Medium Complexity Vehicles Compared to Rework Model's Prediction

From this graph, the impact of the 1998-1999 new truck model can clearly be seen. The 1995-1996 peak seems to indicate a higher than normal level of rework. From this, we may deduce that the aggregate rework peak for the same period may be in large part determined by these medium complexity vehicles. The next steps would be to look at the other vehicle groupings to verify this assumption, and then to start interviewing managers with regards to this particular vehicle program. In this manner, insight can be gained through employment of the rework model.

3.5 Model Validation

After comparing actual data to the model's output, it was necessary to confirm our belief in the model's validity. Clearly, the data fit was most convincing; using four relatively simple inputs, the mere fact that our model not only resulted in the correct magnitude, but also showed variation linked to portfolio changes and platform characteristics gave us confidence in the model's validity. Extending this to program or vehicle specific charts further reinforced our confidence because we could assign observed variations to program changes and externalities.

Secondarily, management review provided a strong vote of confidence. The model's simplicity allows everyone to understand it, and at several reviews, the comments were all positive. While this is not scientific, management's intuition should not be underestimated. Lastly, we believe final validation will occur with time. The model's input to decisions is not exclusive, but rather it is additive, so we feel that watching the model over the next two to four years will confirm its validity while exposing the organization to no risk.

3.6 Model Uncertainty

Although the model provides an accurate prediction of rework, many of its inputs are averages, approximations and distributions. Consequently, we felt it was necessary to add sensitivity bands based on statistical formula. There are several methods for doing this including sensitivity analysis and simulation. Looking at the inputs and the algorithm gave us an indication of which method to employ.

One of the key inputs to the model is vehicle complexity. Recall that we used three complexity groupings for all vehicles in the portfolio. These groups have average complexity levels, and this average is subject to uncertainty, especially if it used to approximate future models. The second key input is the experience curve. While the 15% annual reduction is fairly consistent, we observed that individual programs experienced anywhere from a 20% to a 10% reduction. The third key input is the annual change factor: minor or major. This factor is subject to uncertainty and should be simulated or evaluated for sensitivity. The final key factor, the rework distribution, already incorporates a degree of variability. For the other factors, it would be prudent to add a degree of uncertainty. An easy way to factor in uncertainty is to employ a simulation program like Crystal Ball or @Risk. The goal in using these simulation programs is to generate control bands similar to those used in a statistical process control run chart. These bands prevent us from hastily concluding that a program's rework differed statistically from predicted levels.

4.1 Model Extensions

The primary purpose of the model is to assist product and process improvement efforts. The model provides insight through its holistic view of the development process. However, if process and product improvements are made, the organization status quo has changed, and the model would no longer accurately reflect performance. For instance, we may have accelerated our program learning by adopting knowledge transfer tools. Consequently, the model should be reformulated periodically. At a minimum, the parameters for the model may have to be adjusted. It is unlikely that the factors will change as they should be independent of these changes. On the other hand, there may be exceptions. For instance, if the organization decides to use an existing platform for 10 years, it might be hasty to assume annual improvements occur at the same rate indefinitely. It is more likely that improvement diminishes after a period of time, and the experience factor will have to be adjusted.

There are numerous ways to extend the model. A probable first extension would be to determine if the model needs finer granularity. For instance, perhaps more complexity grouping categories would improve the output. Alternatively, rather than group product changes as new, major and minor, perhaps a few more categories will provide better insight. In addition, it may help to shorten the analysis time period from quarters to months, weeks or days. Further insight may be gleaned by trying to capture more phases of the development process. One possibility is to refine the correlation between parts and rework. As Figure 5 shows, there is a relationship between the number of planned, experimental and production parts. Our model simplifies this relationship by looking only at production parts. It may be interesting to look at both experimental and planned parts, and input those relationships. Another extension could focus on the experience factor. For instance, we may factor the carryover ratio into the algorithm. This ratio refers to the fraction of parts that are reused in a given vehicle. It might help explain some of the reduction in rework, and this may give a more accurate representation of not only the benefits of carryover parts, but also the impact of changing the carryover ratio from year to year.

5.1 Conclusion and Impact

As stated before, the real value in the model is the methodology, which can be used to look at a variety of complex processes. We have already begun to explore how holistic views of other processes can be obtained through simple metrics. However, the model has also had a direct impact on the Truck Group. The output is being used as additional input to aid in portfolio planning. By using the model, rework efforts can be analyzed for dependence on product type and launch sequence.

Additionally, value will hopefully be realized in the analysis of improvement initiatives. General Motors Truck Group spends a significant amount of money on vehicle design. GM generated revenues of over $160 Billion last year, and expenses were around $150 Billion. There are dozens of initiatives on the table for improving vehicle development. They range from adjusting the number of prototype stages to reordering the manner in which parts are released for production. While many of these initiatives have merit, unfortunately it is difficult to evaluate them. Because the model is scalable, we hope to evaluate such initiatives on whatever level they are implemented. Predicting what should happen based on the status quo, and then comparing that to what actually happens under the influence of one of these initiatives should give us an objective appraisal of success. Doing that before ramping these initiatives out to the entire organization could lead to tremendous savings.

5.1 Lessons

As stated in the beginning, the most important finding of this research was the methodology: dissecting a highly complex process into easily understandable and quantifiable factors. The second lesson of this research was defining the specific rework factors. We hope the observations about the rework experience curves and the rework distribution spark additional research in this area. Third, the difficulty in identifying appropriate metrics for product development should not be underestimated. The general belief of management is that metrics yield solutions. As we have shown, metrics can be counterproductive and often misunderstood, and consequently should be chosen with care, evaluated frequently, and used cautiously in decision making. Finally, the scant amount of research in the field of product development management taught us that the field is ripe for creative research.

5.2 Rework Reduction Suggestions

Based on our research and the insights generated by the model, we offer five simple suggestions for reducing rework:

1. Use existing platforms and assemblies for as long as possible. The benefit of experience with a given platform has such a positive impact on the reduction of rework, that any suggestion to move to a new platform should be weighed carefully.
2. Minimize part count. The relationship between complexity and rework is fundamental. Modules and sub-assemblies are a move in the correct direction only as long as specifications and requirements treat the module as a single part in function and in space.
3. Encourage engineering ownership. By assigning ownership to parts, the move down the experience curve might be hastened.
4. Factor the rework distribution into scheduling product launches so that the overall workload remains manageable.
5. Stagger new platform launches as much as possible. The workload impact of multiple new platforms would likely be unmanageable.

Acknowledgements

Many General Motors people were involved in this project ranging from fellow researches to engineers and supervisors. Everyone was professional and enthusiastic, and it is by no accident that GM designs and manufactures such wonderful vehicles. I thank them all. In particular, I would like to acknowledge:

My supervisor and friend David S. Kim
My advisors Steve Graves and Steve Eppinger
My mentors Ruth Zarger and Dave Vander Veen
My data expert Butch Younkin
My co-researchers Paul Gutwald and Jennifer McClean
My coworkers Julie Irwin, Bernice Jung, Susan Owen, Alex Lin, CC Hsu, Eileen Mutiso and Dan VanWingerdan

Finally, thank you Christine—you are my inspiration.

References

Balcerak, K.J. and Dale, B.G., "Engineering Change Administration—The Key Issues," Computer Integrated Manufacturing Systems 1992, Vol. 5, Issue 2, pp. 125-132.

Browning, T., "Use of Dependency Structure Matrices for Product Development Cycle Time Reduction," Proceedings of the Fifth ISPE International Conference on Concurrent Engineering: Research and Applications, Tokyo (July 1998).

Carascosa M., Eppinger S.D. and Whitney, D.E., "Using the Design Structure Matrix to Estimate Product Development Time," Proceedings of DETC'98, 1998 ASME Design Engineering Technical Conferences, Atlanta (September 1998).

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

Czernel G.W., "Developing and Using Benefit Measurements for Integrated Product Development (IPD)," SAE Reprints 962186.

Eppinger S.D., Whitney, D.E., Smith R.P. and Gebala, D.A., "A Model-Based Method for Organizing Tasks in Product Development," Research in Engineering Design—Theory Applications and Concurrent Engineering (1994) Vol. 6, Issue 1, 1-13.

Eppinger S.D., Nukala, M.V. and Whitney, D.E., "Generalised Models of Design Iteration Using Signal Flow Graphs," Research in Engineering Design (1997) 9: 112-123.

Evbuomwan, N., Sivaloganathan S. and Jebb A., 'A Survey of Design Philosophies, Models, Methods and Systems," Proceedings of the Institution of Mechanical Engineers Part B—Journal of Engineering Manufacture (1996) 210: (4) 301-320.

Fine, Charles H. 1998. Clockspeed: Winning Industry Control in the Age of Temporary Advantage. Reading, Massachusetts: Perseus Books

Fleischer, M., "Where Are Engineering Metrics?" Automotive Manufacturing & Production, July 1999.

Jagawa, T., "Front-loading: Shortening Development Time at Toyota Through Intensive Up-front Effort," International Body Engineering Conference, Detroit, Michigan (September 1995).

Kulkarni, D., "Controlling Rework in the Vehicle Development Process," General Motors R&D Contract Report CR-99/01/ESL, April 1999.

Loch C.H. and Terwiesch, C., "Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development," Working Paper forthcoming in Journal of Product Innovation Management, 1998.

Loch C.H. and Terwiesch, C., "Accelerating the Process of Engineering Change Orders: Capacity and Congestion Effects," Working Paper forthcoming in Journal of Product Innovation Management, 1998.

Goldratt, Eliyahu M. 1984. The Goal: A Process of Ongoing Improvement. Great Barrington, Massachusetts: North River Press

Goldratt, Eliyahu M. 1997. Critical Chain. Great Barrington, Massachusetts: North River Press

Hopp, W.J., M.L. Spearman. 1996. Factory Physics: Foundations of Manufacturing Management. New York: McGraw Hill

Kaplan, Robert S., Robin Cooper. 1998. Cost & Effect: Using Integrated Cost Systems to Drive Profitability and Performance. Boston: Harvard Business School Press

Nahmias, S. 1997. Production and Operations Analysis. New York: Irwin

Pittiglio, Rabin, Todd and McGrath, "1997 Product Development Study," World Wide Web (http://www.prtm.com/prtm/newscontacts/95/pdbs.html) online October 01, 1999.

Sharke, P., "ME Metrics," Mechanical Engineering, 121(9): 14-16, September 1999.

Shigley, J. E. 1963. Engineering Design. New York: McGraw-Hill.

Smith R. P. and Eppinger S.D., "Identifying Controlling Features of Engineering Design Iteration," Management Science (1997) Vol. 43, No. 3, 276-293.

Smith R. P. and Eppinger S.D., A Predictive Model of Sequential Iteration in Engineering Design, Management Science (1997) Vol. 43, No. 8, 1104-1120.

Smith R. P. and Eppinger S.D., Deciding between Sequential and Concurrent Tasks in Engineering Design, Concurrent Engineering: Research and Applications, Vol. 6, No. 1, March 1998, 15-25.

Sobek, D.K., Liker, J.K. and Ward, A.C., "Another Look at How Toyota Integrates Product Development", Harvard Business Review, July-August 1998.

Sterman, JD, "Learning In and About Complex-Systems," System Dynamics Review (1994) 10: (2-3) 291-330 SUM-FAL.

Shiba, Shoji, Alan Graham, David Walden. 1993. A New American TQM: Four Practical Revolutions in Management. Portland: Productivity Press

Sterman, J.D. 2000. Business Dynamics: Systems Thinking and Modeling for a Complex World. New York: McGraw Hill.

Ulrich, K.T. and Eppinger, S.D. 1995. Product Design and Development. New York: McGraw Hill.

Vining, G. 1998. Statistical Methods for Engineers. Pacific Grove: Duxbury Press.

Ward, A., Liker, J., Cristiano, J.J., and Sobek, D.K., "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster," Sloan Management Review, Spring 1995.

Womack, J.P., D.T. Jones, D. Roos. 1990. The Machine that Changed the World. New York: Harper Perennial.

Appendix A: Discussion of Rework Causes

The following list is not exhaustive, but it does represent some of the more common product development issues that may adversely affect rework. We have aggregated problems to a high level in an attempt to keep the list tractable yet insightful. In some cases, we have included a brief description of a solution, although many suggestions are based on hypothesis and intuition.

1. Program change. Usually initiated at the executive level, often motivated by cost or marketing. Unfortunately, these decisions are rarely communicated clearly and timely throughout the organization, and consequently they are the most frustrating to the engineers and designers. The impact on rework can be monumental. For instance, a recent truck model was set back 12 months in development due a late decision to offer a different engine option. Reconciling the decisions of high-level mangers with the functional organizations is the biggest challenge of modern management. It is further complicated by the sheer size of the organization and the related communication costs. We believe that a pseudo-cost function may help align executive incentives. The cost function would charge programs for engineering changes even though the exact cost of such changes is difficult to measure.
2. Supplier conflict. Problems range from suppliers going out of business to suppliers informing the group that they cannot complete the part as specified. Again, the impact of such changes is difficult to measure. In some cases, the purchasing organization is able to develop a speedy workaround, while in others, the vehicle is held up for redesign and testing. Moving towards concurrent engineering is likely to minimize the occurrence and the impact of supplier conflicts.
3. Part failure. Failure is a broad term that encompasses engineering-designing conflicts, material failures and weight-cost conflicts. These may be identified through physical or analytical testing. Attempts to improve math-based analytical testing will likely eliminate late discoveries. However, a simple technique for improving first-time design is to assign part ownership. In modern organizations, promotion and rotation is frequent, and many individuals work many parts over the course of one design or several years. By trying to align designs with individuals, we feel that ownership will improve the ability of designers to preempt problems.

4. Assembly failure. Problems of this type arise when the assembly process cannot complete the product as prescribed and the assembly process cannot be changed. In addition to the methods of ownership and concurrent engineering introduced above, improving communications between manufacturing and engineering may help reduce these problems. Unfortunately, cross-functional communications can be expensive in terms of time and resources. One suggestion for long-term improvement is to rotate managers and supervisors through both organizations. While the training costs may be expensive, the benefits in terms of speed and first-run quality will likely compensate.

Appendix B: Discussion of Effective Metrics

There are many problems with conventional metrics that hinder effective product development management. The first problem we uncovered was the lack of consistent definitions and the lack of common understanding of metrics. To some, rework was a measure of product quality, and since our goal was of reducing rework, final product quality should increase. A converse argument is that development rework is essential to resolving product problems, and product quality should improve with rework. To others, rework is a throughput metric. As such, a major source of development delay and cost is rework, so if we reduce rework, we should improve speed to market and cost to market. All interpretations are valid; it is important to clarify definitions and goals.

Another common problem is to correctly encompass the problem with a data collection time scale. Many organizations use financial time periods such as fiscal years, quarters and months to measure and evaluate engineering projects. In addition, the amount of data is often inadequate. In the case of vehicle development, looking at three to six months of data is too short relative to the multi-year development process. It is necessary to go back as far as possible to identify problems and develop changes. A third problem stems from incorrectly "focusing the magnifying glass." Most companies operate is some form of a matrix organization. Moreover, each functional area may have responsibilities that defy a two-dimensional chart. So while it may be easy to focus on a group's or a project's metrics, doing so may sub-optimize performance elsewhere. Metrics for vehicles, which take 24 months to develop, should directly reflect the product and the time frame. Doing otherwise makes it difficult to show the link between behavior and results.

Finally is the issue of discipline. General Motors has numerous data systems supporting vehicle development. These systems range in purpose from data collecting and benchmarking to administrative systems that process information in support of development. Many thousands of individuals are involved in data entry and data usage. Unfortunately, the discipline in system use is poor. Entries are often inconsistent, and relatively few people understand the purpose and impact of their role. For example, EWOs contain reason code fields. This field is intended to record the cause of the change in order to first expedite approval, and second to learn from the change and to prevent the change from being required in the future. Unfortunately, not only are the codes too generic to provide insight, but engineers select the same code, product improvement, 80% of the time. This lack of discipline undermines efforts to understand and improve the process, and it also makes quantifying many aspects of development and rework unsuitable for a model.

A costly solution would be to manually collect data, evaluate it and then automate or institutionalize those metrics that overcome these problems. Of course, there is always a risk that the data collected does not yield any insight. Alternatively, if we can identify and use existing metrics that avoid the aforementioned problems, perhaps we can gain management insight. Many organizations track costs, man-hours, part numbers, supplier contracts, engineering changes, etc.... The problem is to determine which metric(s) best represent the product and the process.

Appendix C: Discussion of Other Rework Factors

There are many factors affecting rework. Recall that factors are distinct from causes in that they universally apply to products and organizations. While many of factors provide process and product insight, not all of them can be quantified to a degree suitable for a predictive model. Others may be quantifiable, but data collection or consistency issues preclude their inclusion. The following list describes other factors that we hypothesized influence rework, and explains why they were not included in the model.

1. Timing Differences. General Motors EWO system is designed to resolve changes within 90 days. Data confirms that, on average, changes are resolved in this time frame. We hypothesized changes that occur faster or slower than the average may be related to notably refined or notably cumbersome systems respectively. We further hypothesized that factoring in these system and product differences into the rework model may add valuable information. However, through further analysis and interviews, we discovered that change timing is influence by too many intangible factors. For instance, a simple change might sit on an engineer's desk due to numerous factors including preferences, program volumes and program focus. Consequently, EWO process times were not included in the rework model.
2. Political Factors. Political interests of different organizations complicate developing vehicles. For instance, the safety and regulatory requirements associated with automobiles sometimes come into conflict with unrestricted engineering and development. While an understanding of their play in the product development arena may make all parties more sensitive to the potential impact, there is no way to eliminate these factors, nor can we measure their impact on the organization. Moreover, including these factors into the model is untenable.
3. Other Metrics. Clearly there are dozens of metrics related to product development and to rework that may seem suitable. Most obvious would be to track EWO documents. Initially we set out to do just that with the goal of discovering a relationship between program complexity, timing and the number of change documents issued. However, it was discovered that EWOs were a tracked metric, and furthermore, many managers used the metric to evaluate performance. Engineers responded by doing everything possible to minimize the number of EWOs issued.

One common technique for this was to initiate an EWO, and to keep that change notice open while additional changes, sometimes completely unrelated to the primary purpose of the EWO, are added to the document. In this manner, the administrative functions of a change are assured, while the benefit of minimizing EWOs is realized. Other metrics were discovered to have similar problems, and that is why they were not factored into the model.

4. Prototypes. GM uses prototype to resolve engineering and design issues. In an attempt to accelerate the development process, additional prototyping stages have been added. The idea is that these additional prototypes will identify specific problems early in the process. In support of the numerous prototypes, release dates are pre-determined so that development efforts occur when needed. Two behaviors were observed. First, engineers did whatever is necessary to meet release dates, and often that included releasing parts that were nothing more than a place holder. Second, having so many prototypes has clogged the information systems, and the ability for the system to absorb lessons from a prototype into the next stage is questionable. For these reasons, and also because the data sample of programs to prototypes is small and inconsistent, the prototype factor was not included in the model. However, we believe that this area definitely deserves further investigation.
5. Organizational Change. General Motors is redefining itself on a major scale, and this is good. Many major processes have been recreated including the rapidly shrinking Vehicle Development Process, the state-of-the-art Validation Center, and the rotation of key managers. However, this change has impacted the organization's ability to learn about its processes. No vehicle program has been executed with exactly the same program, nor do many supervisors understand the processes. Likewise, initiatives for improvement are sprouting up in many places because ownership and resources have been tangled over the past few years. It is logical to look for associations between these process and program execution, but many of the major processes have changed too frequently to factor into the model.