Terminal Descent: The Manufacturing Delay and Disruption Cycle

Brent M. Johnstone

Lockheed Martin Aeronautics Company – Fort Worth

brent.m.johnstone@lmco.com

Terminal Descent: The Delay and Disruption Cycle in Manufacturing

Abstract (150-word limit): Another large, technically complex project comes in late to need and over cost. These failures are attributed to bad cost and schedule estimates, poor program management, or unforeseen circumstances. But is there a deeper explanation? This paper examines the cycle of delay and disruption that begins when changes from planned conditions are introduced. Large or frequent changes begin a cycle of rework and degraded performance that, once initiated, is difficult to escape. This cycle creates cost overruns and schedule delays which, if uncontrolled, can cause disaster. To illustrate the rework and performance cycle, a conceptual model is presented to apply these principles to a hypothetical aircraft development program. The model demonstrates that even relatively small changes in scope, performance of quality, or program funding can quickly push a project off the rails and result in late deliveries and higher costs.

Key words: Disruption, Learning Curves, Manufacturing, Scheduling, Methods, Modeling, Labor

Introduction

Cost and schedule overruns are typical of large projects that push the technological state of the art. Oxford business professor Bent Flyvbjerg's study of "megaprojects" – which he defines as "large-scale, complex ventures that typically cost more than 1 billion US dollars, take many years to build, involve multiple public and private stakeholders, are transformational, and impact millions of people" (Flyvbjerg, 2014) -- led him to develop the Iron Law of Megaprojects: "Over budget, over time, under benefits, over and over again." (Cohen, 2023). He found these megaprojects – think "high-speed rail lines, airports, seaports, the Olympics, high-energy particle accelerators, logistics for large supply chains, etc." (Flyvbjerg, 2014) – produced these dismal results:

- Only 47.9% are delivered on budget.
- Only 8.5% are delivered on budget and on time.
- Only 0.5% are delivered on budget, on time and with the projected benefits. (Cohen, 2023)

Christian Smart, in his book *Solving for Project Risk Management*, provides similar examples of large cost and schedule overruns:

- 80% of aerospace and defense projects have cost overruns while 90% have schedule delays. The average A&D development program exceeds budget by 50%, and one in six do so by a factor of two or more.
- Software and information technology projects average 1-2% growth in requirements each month. The average software/IT program experiences 43-56% cost growth in addition to exceeding the original schedule by 63-84%.
- Infrastructure build costs for the Olympic Games averaged 156% cost growth between 1968 and 2016, with the 1976 Montreal Olympics exceeding the original plan by eight times.
- Smart documented similar growth in cost and schedules for infrastructure projects such as dams, railroads, bridges and tunnels, and roads. Between 80-90% of these projects experienced cost overruns and 70-80% experienced schedule delays. (Smart, 2021)

The common denominator behind these projects is they are long-cycle, large dollar projects which either push the technological or logistic state of the art or are vulnerable to unanticipated design, engineering, manufacturing, or environmental issues.

Why are these types of projects so prone to cost overruns or schedule delays? Congress requires large Defense Department programs which exceed 50% growth above baseline cost estimates (a so-called "critical breach" of the Nunn-McCurdy Act) to undergo a root cause analysis. In a summary of 20 root cause analyses performed by the DoD Office of Performance Assessments and Root Cause Analyses (PARCA), the most frequently cited cause was poor performance by government or contractor program management, followed by unrealistic baseline cost and schedule estimates and changes in procurement quantities. (Bliss, 2015)

Outside of DoD, infrastructure researcher Matt Siemiatycki places the blame for cost growth more broadly. "The three main explanations for cost overruns and delays," he writes, "are technical challenges, over-optimism, and strategic misrepresentations" -- the latter being false descriptions of a project's affordability by its advocates to secure political and financial approval. (Siemiatycki, 2015)

And yet these explanations – poor program management, unrealistic estimates, and deliberate lies advanced by program advocates – are ultimately unsatisfying. For one thing, they do not explain why these situations never seem to improve over time, especially in the defense realm. Every decade since the 1970s has seen one or more Defense Department and Congressional initiatives to reform the acquisition process, many of them aimed precisely at these issues. And yet the title of J. Ronald Fox's book on the subject (*Defense Acquisition Reform 1960-2009: An Elusive Goal*) captures the frustration of military reformers looking back at the results. (Fox, 2011)

Mark Twain famously said, "Everyone complains about the weather, but nobody does anything about it." If cost and schedule overruns are caused by poor program management or unrealistic, compromised estimates, then how is the problem never fixed? How did initiatives to better train government and industry program managers or to produce truly independent cost estimates not improve the situation? Is it truly possible that we never learn the lessons from prior failures?

What is more, these explanations are not very helpful to the cost estimator who is trying to estimate the potential risk of cost overruns on a program, particularly in situations where there are deviations from the original program plan with impacts to cost and schedule. Asked to estimate the potential cost impact of delay and disruptions, how does an analyst price the impact of allegedly incompetent management? Or, if in fact original estimates were understated, can the analyst determine what should be the new cost of a project, considering its technical and programmatic challenges?

Or is it possible that the issue of cost and schedule growth stems from something more fundamental to the design and build process? And is it possible that understanding these internal dynamics might help the estimator provide a better, more defensible estimate of delay and disruption impacts?

This paper will attempt to answer these questions by examining the cycle of delay and disruption that begins when changes from planned conditions are introduced.

How Schedules Grow Over Time

We begin by noting the relationship between cost and schedule. From empirical observation, we know that cost and schedule growth are frequently interrelated. Figure 1 demonstrates Christian Smart's research on the relation between cost and schedule growth based on NASA project history. (Smart, 2021)

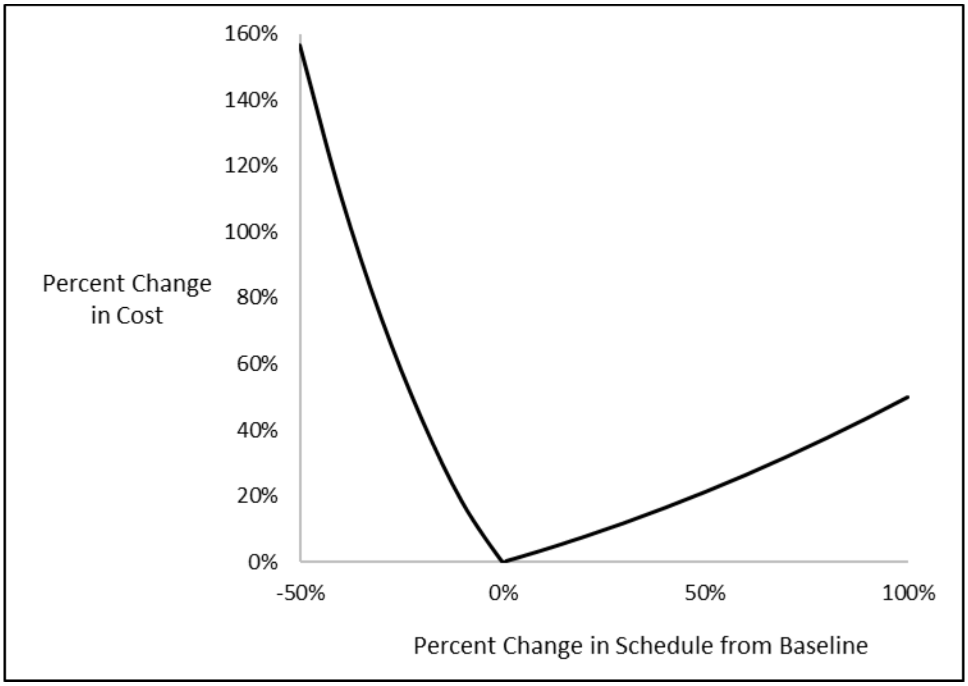

Figure 1. Cost and Schedule Growth. Reprinted with permission from Christian Smart.

This graph, however, does not answer the question of causality. Do schedule delays create cost overruns, or vice versa? Or is there a common factor that drives both to expand? To attempt to answer these questions, we will turn first to understand the reasons for schedule growth over time.

In its study of schedule growth, the Australian consulting firm Broadleaf Capital International argues that program managers and schedulers often fail to understand the true risks surrounding the program schedule. The typical schedule risk profile is like that shown in Figure 2 with an approximately normal distribution which shows the possibility of schedule growth approximately equal to the possibility of schedule contraction. In reality, the true schedule risk profile should recognize the strong possibility of a major schedule slip, i.e., the risk curve should have a right-hand tail on its probability distribution. The lack of a right-hand tail in a typical schedule profile is often justified on the erroneous assumption that any delays on the schedule critical path will be day for day. This assumption, argues Broadleaf, leads program schedulers to believe that “[t]here is a one-to-one linear relationship between slippage in a critical path task and extension of the project finish date. By itself, the linear relationship between activity durations and project duration cannot generate a long right-hand tail.” (Broadleaf, 2021)

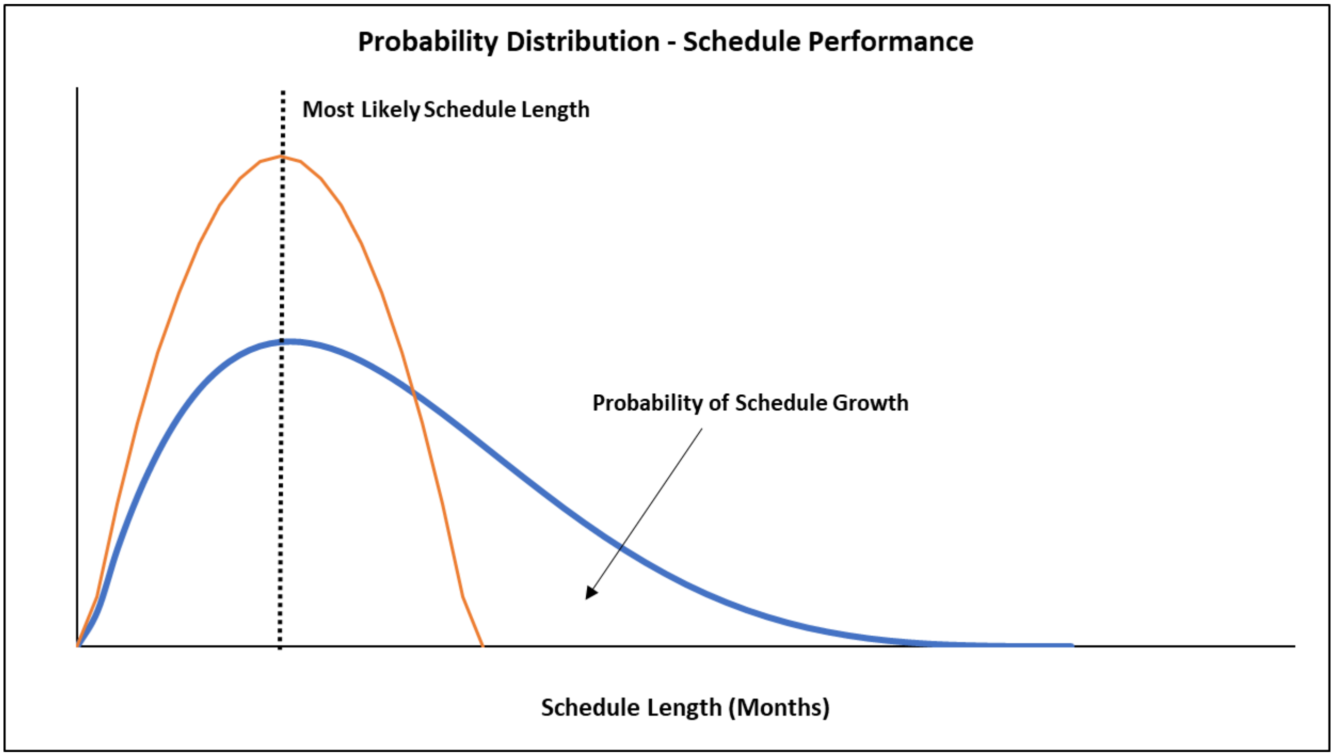

Figure 2. Probability Distribution – Expected Schedule Performance (Broadleaf, 2021)

So why is this a bad schedule assumption? Because it ignores “a simple and familiar non-linear mechanism that affects projects”:

> “When a schedule slips and looks set to slip further, pressure to regain control can lead to work being carried out in a different way from what had been planned…[I]t is not uncommon for formal controls to be deliberately relaxed, authority might be delegated closer to the work front…or effectively subverted to enable some progress, any progress, by any means possible. This is a state often referred to as firefighting.”

Furthermore,

> “Good planners aim to schedule general works in a sequence that allows each part to be completed as efficiently and realistically as practicable. It follows that, once work slips away from the plan, the project is no longer optimized. The new state will be suboptimal, and productivity will be lower….”

Unanticipated events create an initial schedule delay. Measures are taken to recover but result in diminished productivity. Losses in productivity in turn cause the program to fall further behind schedule resulting in further recovery measures that injure productivity. As shown in Figure 3, the program becomes trapped in a downward spiral – a terminal descent, if you will, which if uninterrupted can lead to a cost and schedule “blow-out.” (Broadleaf, 2021)

Figure 3. Delay Cycle and Schedule Impact. Reprinted with permission from Broadleaf Capital International.

This chain of events provides an explanation for schedule delays which does not invoke accusations of poor program management, overoptimistic schedule estimates, or deception by government or industry. Instead, schedule delays can be explained by “a systematic interaction…between the impact of recognized risks [and] firefighting behavior, causing a loss of productivity, stimulated by concern about delay.” (Broadleaf, 2021)

How does this delay in schedule relate to cost growth? We can identify three distinct impacts:

- Reduction in productivity, as illustrated in Figure 3. As productivity declines, more labor hours will be required to achieve the same end-result. Worse, productivity decreases will likely create further delays to the schedule. As we shall see later in this paper, the results can be devastating.
- Cost of the “marching army” – that is, the cost of overhead and fixed project support required during the length of a project. These costs are directly tied to the length of time it takes to complete the program. As schedules slip, these fixed costs increase.
- Independent of the reduction in productivity, many of the “expedient measures to recover” schedule have their own negative impacts to cost.

For example, in the aircraft industry, there are a series of tried-and-true techniques to attempt to recover behind schedule. Virtually all of them create higher costs:

- Larger than optimal crew sizes.
- Out of sequence work / out of station work.
- Overtime.
- Additional shifts.
- Hiring new employees or transferring employees from other programs.
- Continuing to charge to overhead or other accounts even if there is not productive work.
- Production gaps.
- Furloughs / voluntary layoffs.
- Movement of personnel into other areas where productive work is available.
- Resequencing of units.

- Utilizing In-house rework and repair facilities, as opposed to returning defective parts to vendors.

Delay and Disruption

The downward spiral illustrated in Figure 3 begins with "unanticipated events." What might those unanticipated events be? Broadly speaking, we can define these as changes from planned conditions. These changes can represent a wide range of possibilities, include design or scope changes, late customer decisions, inadequate specifications, diversion of key resources, or delays. They may be changes in standards, regulations, schedules, suppliers, or technology. All of these create disruption, which management consultant Kenneth Cooper defines as, "the change-induced impact on the cost of performing work that is not directly changed (so, productivity loss and increased rework)." (Cooper, 2022)

It is important to note that the cost of disruption does *not* include the direct cost of the change itself, but its follow-on impacts. For example, an engineering change will cause additional hours for engineering design and release. Those direct cost impacts are not considered disruption. But the cost of downstream impacts – the loss of manufacturing learning and productivity, the associated fluctuation in shop floor mechanics staffing, the negative delivery schedule impacts – all these follow-on impacts would be considered "disruption."

Common causes of delays and disruptions on an aircraft program are:

- Late-to-need engineering.
- Late-to-need tooling or equipment.
- Changes in customer requirements or work scope.
- Changes in customer funding.
- Part shortages and/or late delivery of materials.
- Changes in availability of long lead parts or materials.
- Engineering changes (particularly driven by changes in loads or weight issues).
- Suboptimal resourcing.

Delay and disruption can impact cost and schedule at any point in the product life cycle. But its pain is particularly acute in the early development phases of a program.

The S-curve pattern – marked by relatively limited cost improvement early in the program, followed by a steep improvement during initial production phases and an eventually flattening out as the program matures – is frequently observed in learning curve history. (Carr, 1946; Cochran, 1960, 1968; Engwall, 2001) The "bulge" observed in the early units (reference Figure 4) is typically attributed to the impact of engineering changes and the associated disruption. The most mature expression of this can be found in Cochran, et al. (1977), who attributes its existence to "the costs associated with design delays and changes."

> "[I]ntroduction of a new product causes additional cost above the normal improvement trend. The effect on direct labor hours is small for relatively stable designs and normal methods but much greater when the new product requires major methods changes with new tooling, extensive employee training and a protracted period of startup. This impact increase when the design is unstable and has to be debugged and changed during the production process, and again by design delays." (Cochran, et al., 1977)

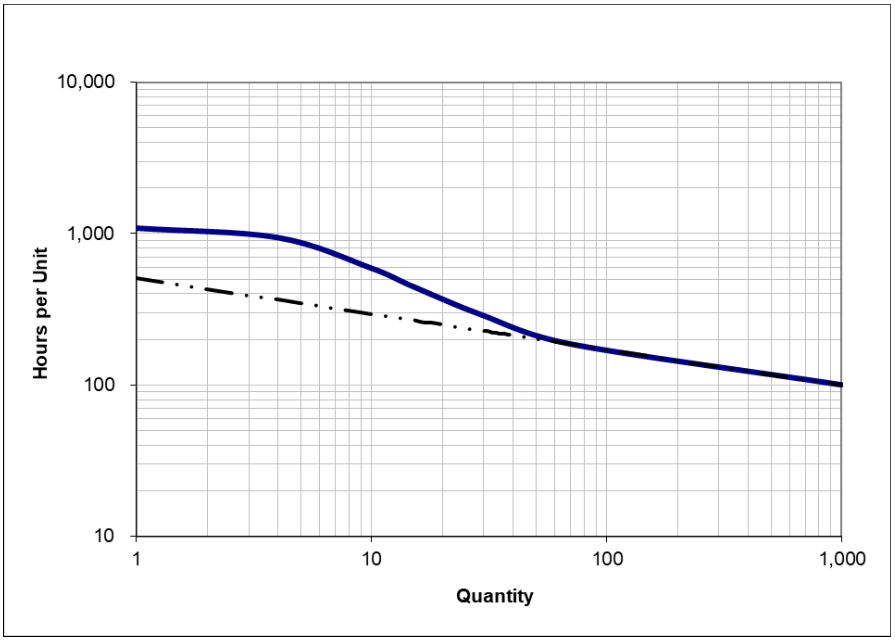

Figure 4. Learning Curve "Bulge" Observed in Early Units

These pressures have a direct correlation to schedule pressures, which Cochran introduces via the concept of "time compression":

> …[T]ime intensive demands are placed upon the entire organization, particularly when the product itself makes strong demands upon operational performance and precision manufacturing techniques. There are frequent design changes and production interruptions, production technology, and perhaps even new procedures of production planning and control must be developed. Time is always an actor, and the result Is a rather halting rate of cost reduction, in which design and manpower changes and production suspensions are an inherent element of the production process." (Cochran, 1968)

Accordingly, a better understanding of how the delay and disruption cycle impacts costs should be valuable to us in estimating early program hours, which is frequently one of the most difficult tasks facing an estimator who usually has little, if any, program-specific cost history to work from.

Identifying and Projecting Disruption

Yet having said that, we still have a difficult time projecting the cost of disruption in the future, or even identifying disruption costs which have occurred in the past.

The most convenient estimating approach would be to identify historical delay and disruption costs form prior programs and use this data as insight to predict the cost of future disruption. But it is typically almost impossible to separate out accounting charges related to disrupted, as opposed to baseline work. We may be able to see a loss of productivity, for example, but it is often not possible to attribute this to a singular cause because of the interrelatedness of the disruption cycle. And costs which can be charged to special accounts, e.g., rework and repair, are only part of the total cost picture. It is possible to broadly estimate cost impacts as a high level using a "similar-to" from an analogous program, but two disruption scenarios are rarely exactly alike, and substantial judgment must be used to compensate for the programmatic differences between the current program and its analogy.

If we cannot construct the cost of delay and disruption using actual cost or analogous history, how can we do it? Analysts have suggested that perhaps the disruption cycle can be modeled, and the resulting estimates calibrated at the top line to actual performance.

The initial suggestion for this idea came from two separate sources during the 1970's. One came from learning curve writer E. B. Cochran, who proposed:

> …the construction of an interactive computer model for the disruption process. The immediate goal was to predict disruption cost (in terms of production manhours) by quantifying their dependence on uncertainty in the development phase and the urgency of delivery dates. Several major benefits are gained by a successful effort: more realistic appraisal of the cost and delivery dates for a major program, both before go-ahead and during its conduct; improved time-cost tradeoff analysis; more effective early-warning of cost and delivery difficulties; and more objective appraisal of the facts which underlie contractual controversies involving disruption. An interesting result is that a model shows far greater ability to identify and measure disruption costs after they occur than is provided by any accounting system. (Cochran, 1978)

Prototypes of such a model were created (Wolfe, et al., 1980) but efforts apparently ceased after Cochran's death in 1980 and the details of the model have been lost.

A more lasting contribution comes from the legal field, where the problem of quantifying disruption becomes particularly acute when contractual disagreements between government and contractor over which party is responsible for the costs of delay and disruption advance to the courtroom.

There are a variety of methodologies to calculate disruption costs which are used in legal actions (measured mile, earned value, total cost, modified total cost, et al.) (Biser, 2017). But there is no agreed-upon best practice accepted by all parties in a legal dispute. One observer wrote "when one reviews contractor methods for calculating disruption claims, one is struck by their ad hoc nature, their lack of fundamental organizing principle, their inconsistency of approach from one case to the next, the use of arbitrary technique." (Cochran, 1978) It is not surprising that disruption claims are "typically underestimated, under-recovered, and involve a contentious recovery effort….Disruption is the most misunderstood, most debated, most abused, and hence least well-recovered aspect of cost damages." (Cooper, 2022)

The so-called "systems dynamics" approach to calculating disruption costs was developed in the 1970s as an attempt to address these issues. This methodology, pioneered by consultants at Pugh-Roberts Associates, attempts to identify disruption cost impacts by modeling the build process with a variety of feedback cycles which imitate delay and disruption events. This model can be run to simulate costs after the changes (the actual cost of the project) as well as to estimate costs without change (the "should cost" value). Figure 4 demonstrates the delta between actual cost and should cost. This difference represents a) the cost of the direct change itself, which can usually be discretely identified in the accounting records as well as b) the associated delay and disruption, which cannot be extracted discretely from accounting. (Cooper, 1980, 2022)

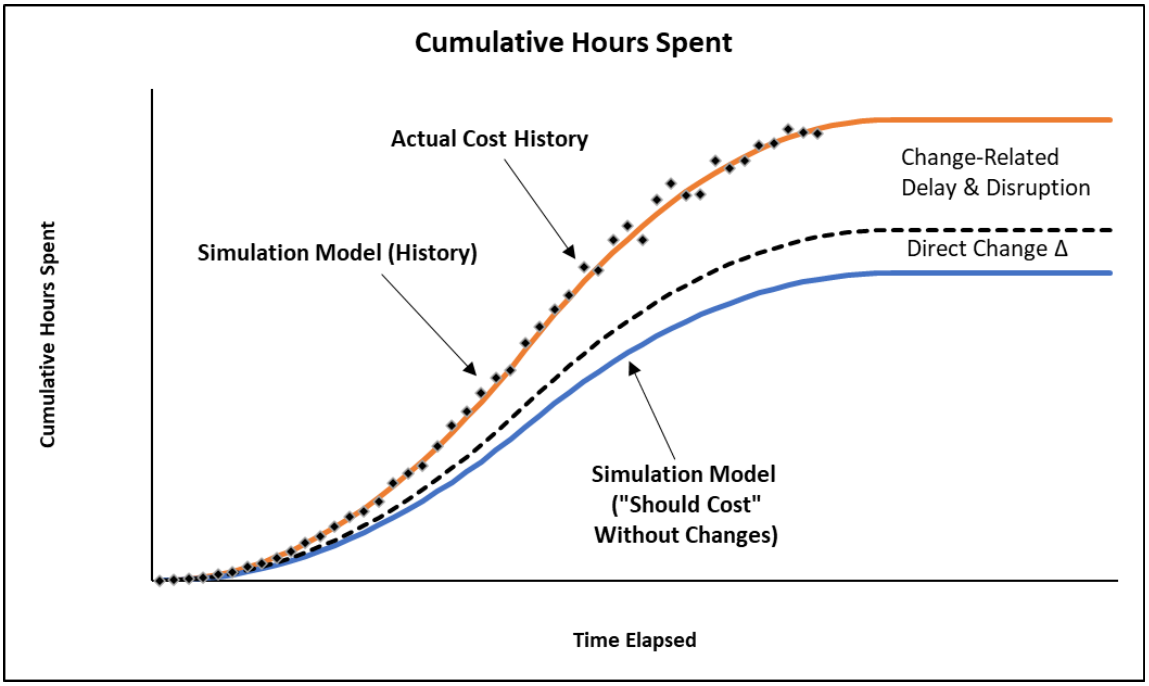

Figure 4. Systems Dynamics Modeling Approach to Delay & Disruption Estimates (Cooper, 2022)

Systems dynamics modeling was initially employed in a lawsuit between the U.S. Navy and Ingalls Shipbuilding. Ingalls sued the Navy claiming the Navy was responsible for $500 million of delays on the LHA amphibious assault ship and the DD-963 destroyer program. Kenneth Cooper led development of a simulation model which imitated the Ingalls shipbuilding process and displayed the cost impact of the Navy's directed changes. The Navy settled the claim out of court with Ingalls for $447 million, of which the model was responsible for $200-$300 million of the settlement. (Cooper, 1980) While systems dynamics has been used in other legal disputes as well as manufacturing simulations, its use in the Ingalls claim is still its most famous use-case.

The change cycle, as described by Cooper in a 2022 presentation, mirrors the delay cycle illustrated in Figure 3 previously. In this cycle:

- Customer adds and changes work, contractor responds by staffing up more.
- Contractor uses more overtime and must hire in tight markets.
- Less skilled hires are less productive and need more supervision.
- Rework causes more rework.
- Under schedule pressure, morale suffers, and work is less productive.
- Late and changing engineering hurt build. (Cooper, 2022)

This cycle is not unique to the shipbuilding business. Cooper argues it is the underlying pattern seen in literally hundreds of projects experiencing cost and schedule overruns. (Cooper, 2022)

The key insight into the systems dynamics model of disruption is the interrelationship between quality and the amount of work accomplished. Quality is defined as "the fraction of work that will *not* require rework." (Cooper, 1980). As Figure 5 demonstrates, work is accomplished over time at a rate determined by the people working the project and their productivity. As that work is performed over time, some fraction of it will require rework. However, this rework is often not recognized immediately. It is only discovered downstream in the design or build process. For example, an engineering release may contain an error which is not discovered until the mechanics attempt to install parts on the aircraft. Only then is the error discovered and the need for rework identified. Once rework is discovered, it is added to the existing backlog of work still to be accomplished.

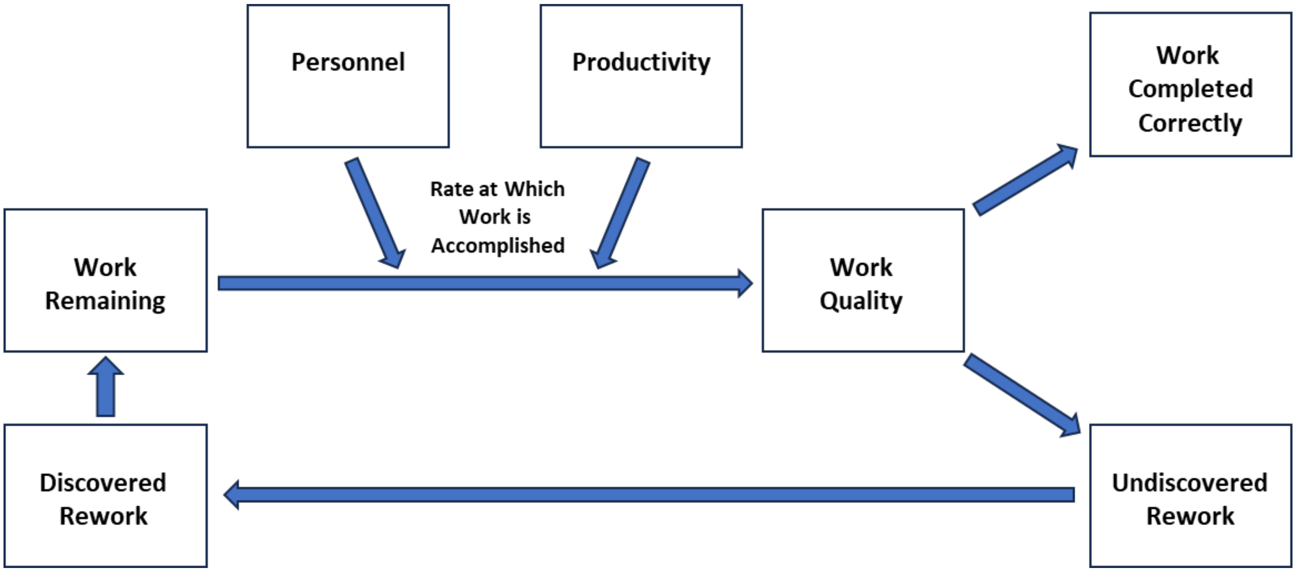

Figure 5. The Change Cycle. (Cooper, 1980)

Disruption Modeling: A Proof-of-Concept Approach

In the next section of this paper, we will transition from the theory of disruption to a practical demonstration of how cost and schedule can be disrupted on a program. To do this, a proof-of-concept model was built to illustrate the dynamics of the change cycle and show some of the sensitivities that create cost and schedule growth.

The requirements of the model are as follows:

- The model should provide a demonstration of potential disruption on a notional aircraft development program from design to fabrication to component assembly to mate/final assembly to flight line.
- It should be high-level and be modeled at the major task area (fabrication, component assembly, mate and final, flight line). It should not go further into the individual components (forward fuselage, center fuselage, wing, aft fuselage, etc.) nor go down to individual build station.
- It should not be based on company-proprietary data nor be based on any current or planned aircraft program.
- It should be Microsoft Excel-based and not require commercial simulation software which in turn have expensive fees and require significant training to use.

- Its output should be directional only, intended to instruct, and not suitable for proposal estimating or litigation purposes.

There is no publicly available simulation model for an aircraft build. However, a simulation model for Navy shipbuilding was described in detail by McCue (1997). This model was constructed to simulate and compare program performance at U.S. defense shipyards. It was in turn based on the work of Kenneth Cooper and Pugh Associates for Ingalls Shipbuilding as well as subsequent systems dynamics modeling efforts. The McCue model – in particular its estimating logic as well as the cost and schedule relationships behind it – was used extensively as a guidepost to construct this aircraft simulation model. For our purposes, it is not important if the specific cost and schedule estimating relationships are directly applicable to the aircraft industry. The intent is to demonstrate the logical structure of a delay and disruption model and how it might operate.

We begin by imagining a notional aircraft development program, not one based on any current or planned aircraft program, but hypothetical only. It has the following characteristics as seen in Figure 6:

- The program under study, including detail engineering design release, occurs over 48 months. This period excludes the extensive offsite flight test program typical of a development program.
- The year-long detail engineering design phase begins six months before the start of fabrication. This period excludes conceptual and preliminary engineering design.
- Eight (8) fighter aircraft will be built with first delivery occurring 29 months after the start of fabrication with follow-on deliveries occurring every two months following.

Figure 6 shows the typical sequence of aircraft build. Before build can start, engineering must begin the release of detailed design models to the shop floor. Detail part build begins in fabrication. Once the major structural parts have been fabricated, they can be loaded into jigs to begin component assembly. When the first aircraft components are completed, the individual pieces of the first aircraft can be mated and the landing gear attached. After mate, the aircraft moves on wheels to final assembly when the empennage is attached and avionics and vehicle systems and installed and tested. When final assembly is complete, the aircraft rolls out of the assembly building and to the flight line, where ground testing is performed prior to the initial checkout flights. Our model ends when the tested aircraft is either turned over to the customer or ferried out to the flight test site for further development testing, which typically continues for several more years.

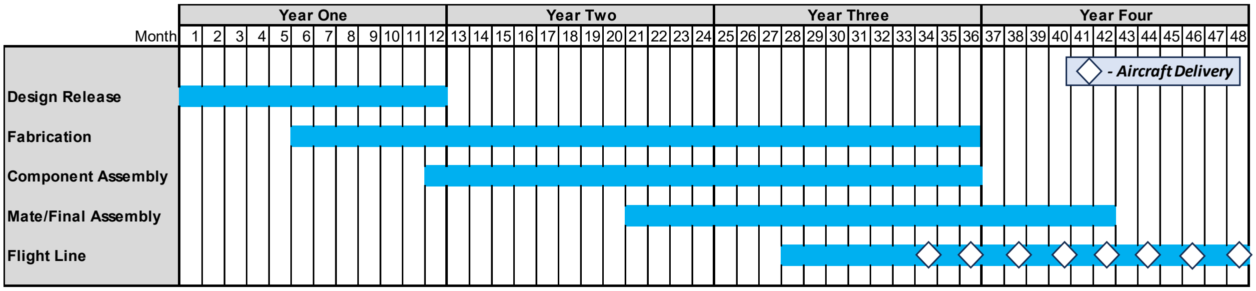

Figure 6. Notional Program Schedule.

The model is separated into five modules, each one representing a major task area (design, fabrication, component assembly, mate/final assembly, and flight line). Each area has a planned headcount based on productivity and quality assumptions around the program schedule. It also has an area for actual performance, where all these variables are allowed to deviate from the plan, allowing us to compare cost and schedule performance under different scenarios. Design release and build is modeled by work week. This allows for more discrete adjustments to headcounts as schedule pressures force deviations from the plan.

The major workflows in each module are illustrated below.

Work Accomplishment.

Figure 7 shows the work accomplishment cycle. Industrial engineered standard hours are used as the common denominator to measure progress and productivity.[1] A major task area begins with a set number of standard work to be accomplished. Standard hours are reduced over time as work is accomplished. The gross completion rate measures how many standard hours are earned (i.e., how much work is accomplished) in a week. The gross completion rate is in turn influenced by the actual productivity rate. (Productivity is described in more detail below in the "Impacts on Productivity.") Work that is completed reduces the work remaining at the beginning of the next week, and the cycle begins anew.

---

[1] Standard time is defined as "the time required for a fully trained operator to complete a defined element of work to known specifications following a prescribed method utilizing specific equipment, tools, material and workplace layout, including average personal, fatigue and delayed time allowance. Standard time is theoretically achievable when rework and scrap are eliminated, all production rate tooling is available, engineering requirements and manufacturing processes have stabilized, machine down time is eliminated, labor is perfectly matched to schedule requirements, parts/material shortage problems have been eliminated, workers become familiar with all aspects of the job and work has been repeated in a stable environment many times, e.g., hundred to thousands. All of these conditions are seldom, if ever, achieved except on high volume, short cycle, mature and stable product operations." (Engwall, 2001)

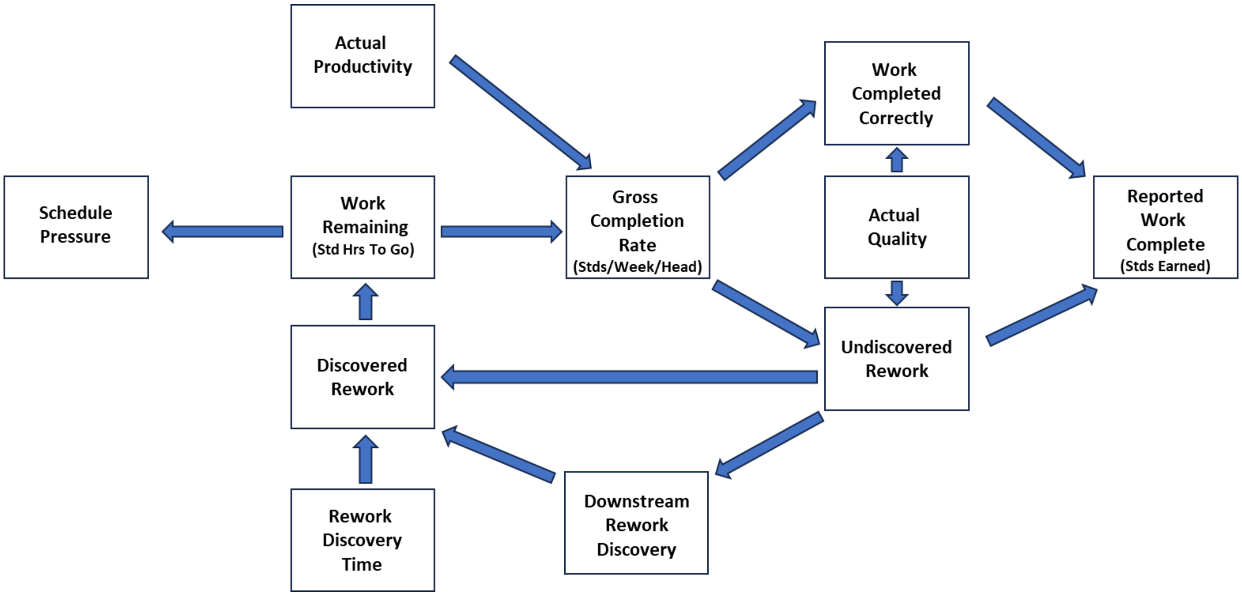

Figure 7. Work Accomplishment Cycle (McCue, 1997).

However, work that is completed is still subject to the rework cycle. As discussed earlier, quality measures the percentage of work that is correctly performed. Therefore, work is completed correctly, or it becomes undiscovered rework. Importantly, this undiscovered rework is not recognized immediately. It is subject to a time delay which varies by major task area. Depending on the area, rework is not discovered until several weeks later (6-week delay in design, 5 for fab, 3 for component assembly and mate/final, and 2 in flight.)

In addition, rework may be discovered in the current task area or downstream in a subsequent task area. (Reference Figure 8.) For example, an error in an engineering release may not be discovered until final assembly or even the flight line. The discovery time accelerates as a build area completes its work. The closer a build area is to finishing, the faster an error is discovered. Once it is discovered, it is added back into the backlog of standard work to be accomplished.

		Rework Discovered In:				
Rework Generated By:		Design Release	Fabrication	Component Assembly	Mate/Final	Flight Line
	Design Release	▨	▨	▨	▨	▨
	Fabrication		▨	▨	▨	▨
	Component Assembly			▨	▨	▨
	Mate/Final				▨	▨
	Flight Line					▨

Figure 8. Downstream Discovery Process

Note that the downstream discovery process puts additional cost and schedule pressure on the later segments of the build, particularly mate/final assembly, and flight line. These areas are discovering rework generated from design, fabrication, and component assembly as well as rework generated internally to their build areas.

Impacts on Productivity

Productivity is measured in terms of standard hours per 40-hour employee, i.e., a productivity factor of 4 means a mechanic earns 4 standard hours during each 40-hour week. At the beginning of the program, the productivity will be low, but as workers gain experience, this value will increase. This mirrors the traditional learning curve effect. The model assumes a baseline learning curve slope. Based on the number of cumulative units built, productivity will be adjusted based on that learning curve.

Each task area has a “normal” level of productivity consistent with the position on the learning curve. As Figure 9 shows, however, learning is not the only factor that influences productivity. While learning improves productivity over time, it can be degraded by morale, fatigue, and the entrance of new workers. All three of these are related by schedule pressure. As a program falls behind schedule, there is increasing pressure to try to recover it. This is typically done by increasing overtime or hiring additional workers, or both. However, increased overtime creates fatigued workers who perform less efficiently. Hiring new workers over and above what has already been planned creates learning loss and potential disruption. Lastly, as the program falls behind schedule, employee morale is negatively impacted with deleterious effects on productivity. The specific morale and fatigue degradation factors are taken from McCue (1997); the new worker factor is derived from the author’s prior research (Johnstone, 2021).

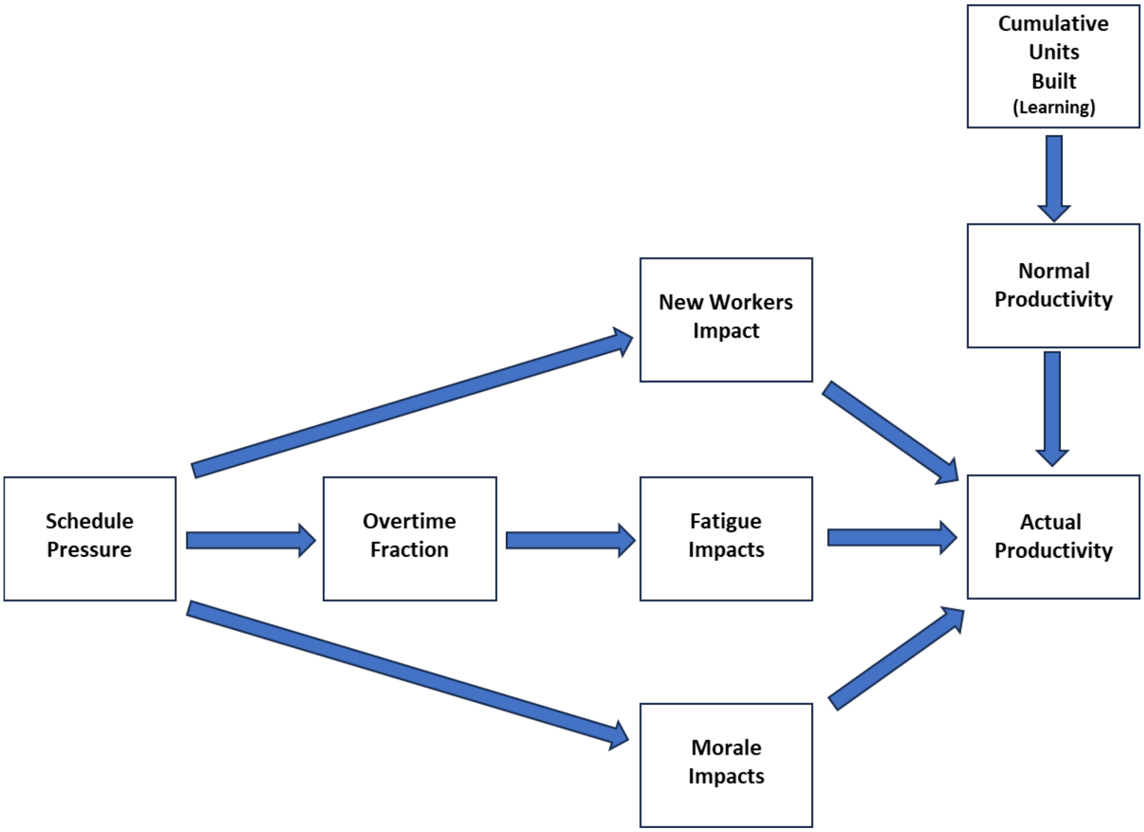

Figure 9. Productivity Impact Cycle (McCue, 1997).

Impacts on Quality

Similarly, each task area has an assumption about the “normal” level of quality. The model assumes a baseline of 85% quality, mirroring the assumption from McCue (2007). Like productivity, quality can be degraded by morale, fatigue and the impact of new workers introduced due to schedule pressures. Figure 10. shows these factors can degrade quality. The specific morale and fatigue degradation factors are taken from McCue (1997); the new worker factor is derived from the author’s prior research (Johnstone, 2021).

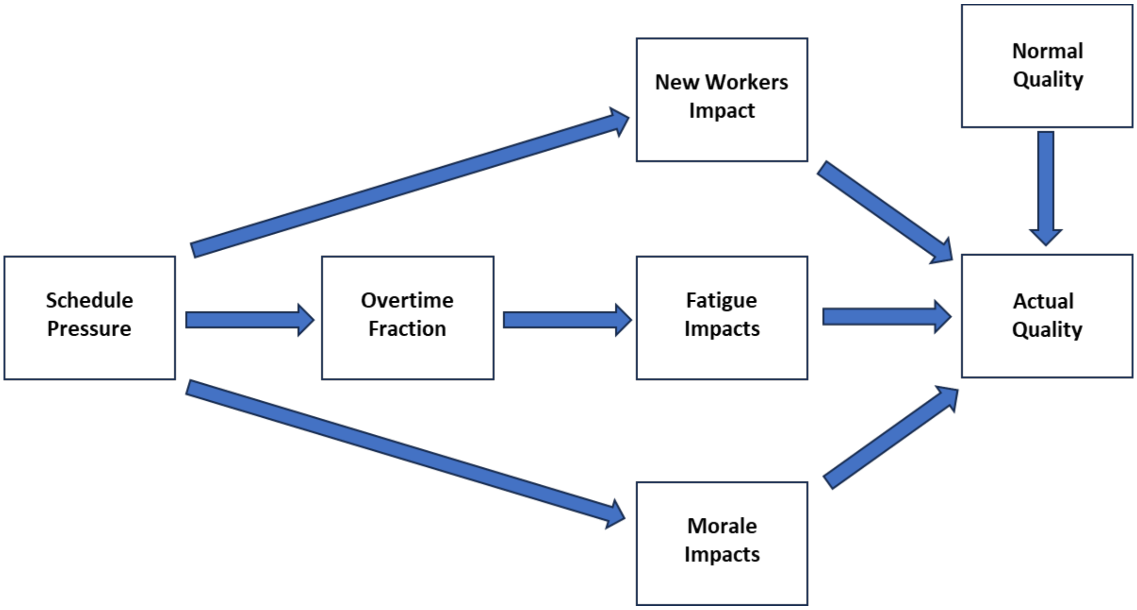

Figure 10. Quality Impact Cycle (McCue, 1997).

Labor Adjustment

The project plan has an assumed headcount over time sufficient to accomplish the work on schedule – provided there are no unexpected disruptions. If productivity degrades or schedule pressure is introduced, the desired headcount may increase as the program attempts to recover schedule. However, as Figure 11 demonstrates, there are two major limitations to adding personnel. First, a certain amount of time to adjust labor is required – new employees cannot be added instantaneously. There is a delay of 2-4 weeks before those new heads can be brought on the program. Second, there is a maximum weekly labor adjustment – only so many new employees can be on-boarded to the program at a certain time. Labor cannot increase by more than 50 heads per week. Model values for the time adjustment factor as well as the maximum weekly adjustment is derived from McCue's shipyard model (1997).

The consequence of these two factors is that the actual number of employees is frequently less than the desired number at a given point in time. Therefore, schedule problems cannot be immediately addressed by adding additional workers. The number of actual labor hours expended for the week is calculated by multiplying the number of actual employees by 40 hours per week plus any overtime.

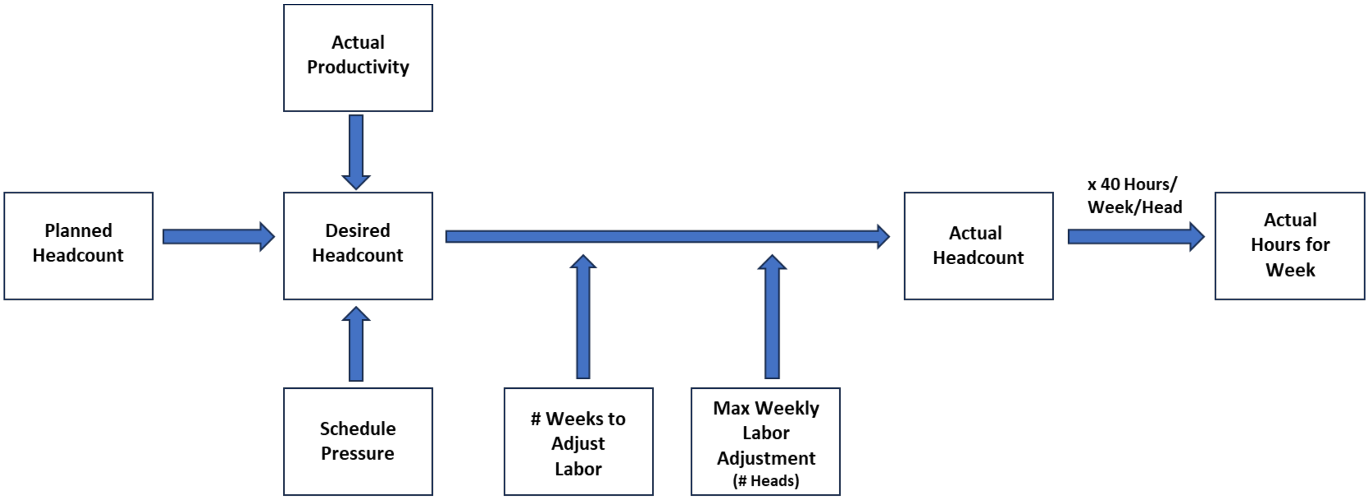

Figure 11. Labor Adjustment Cycle (McCue, 1997)

Case Studies

As an illustration of how the delay and disruption cycle can derail cost and schedule, two case studies have been developed using the model.

Case One – 80% Quality

In the first case, quality is assumed to be worse than expected (80% quality versus planned baseline of 85%). This will generate more rework than planned, and theoretically require additional resources if we try and maintain the original program schedule.

This in fact is what happens according to the model. Figure 11 shows the hours per week for the planned versus the actual build:

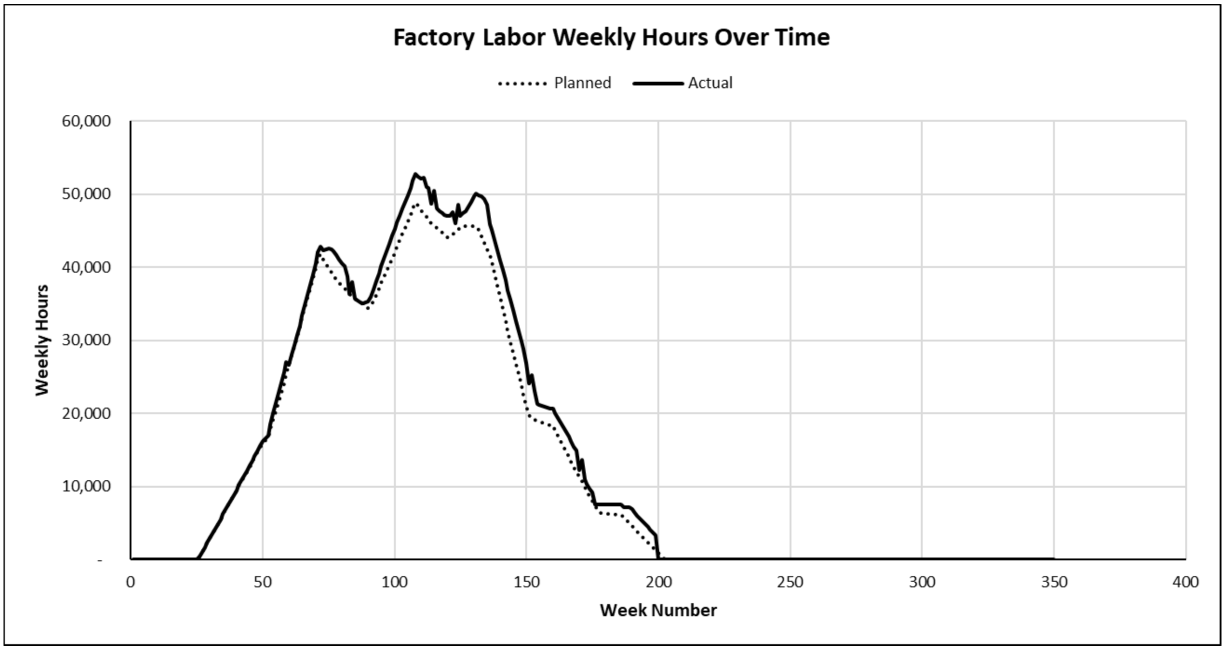

Figure 11. Case 1 – Factory Hours per Week (Planned & Actual) Over Time.

Overall, the model demonstrates an 8% overrun of actual hours to planned (Figure 12). By increasing headcounts, however, the program can maintain the program schedule and complete development on time. Note, however, that the program progresses through each build area, the overrun grows. This is due to the downstream discovery of rework, which creates more disruption as errors created in design or fabrication are uncovered later.

Area	Hours		
	Planned	Actual	Variance
Fabrication	1.00	1.04	4.4%
Component Assembly	1.00	1.08	7.5%
Mate/Final	1.00	1.12	11.7%
Flight Line	1.00	1.16	15.9%
Total Factory	1.00	1.08	8.5%

Figure 12. Case 1 -- Results by Task Area.

It is also important to understand the compounding effort of rework as it cycles through the work accomplishment cycle in Figure 12 above. Although the model uses 85% quality to develop the planning baseline, the rework hours is equivalent to almost 20% of the standard hour work. This is because the discovered rework task is itself subject to potential errors and may require subsequent rework as well. When the model inputs are adjusted to 80% quality, rework hours now compose 28% of the standard hour work – almost a 42% growth in rework activity.

Case Two – 75% Quality

In the second case, quality is assumed to be slightly worse (75% quality versus planned baseline of 85%). We might expect therefore a proportional increase in the number of hours spent.

In fact, the overrun is significantly worse than before – 44% over budget in total. Furthermore, we cannot maintain schedule. The rework requirements have grown faster than the ability of our headcount to keep with them, putting the program behind schedule. The critical path schedule was grown by 23% over the baseline, as shown in Figures 13 and 14:

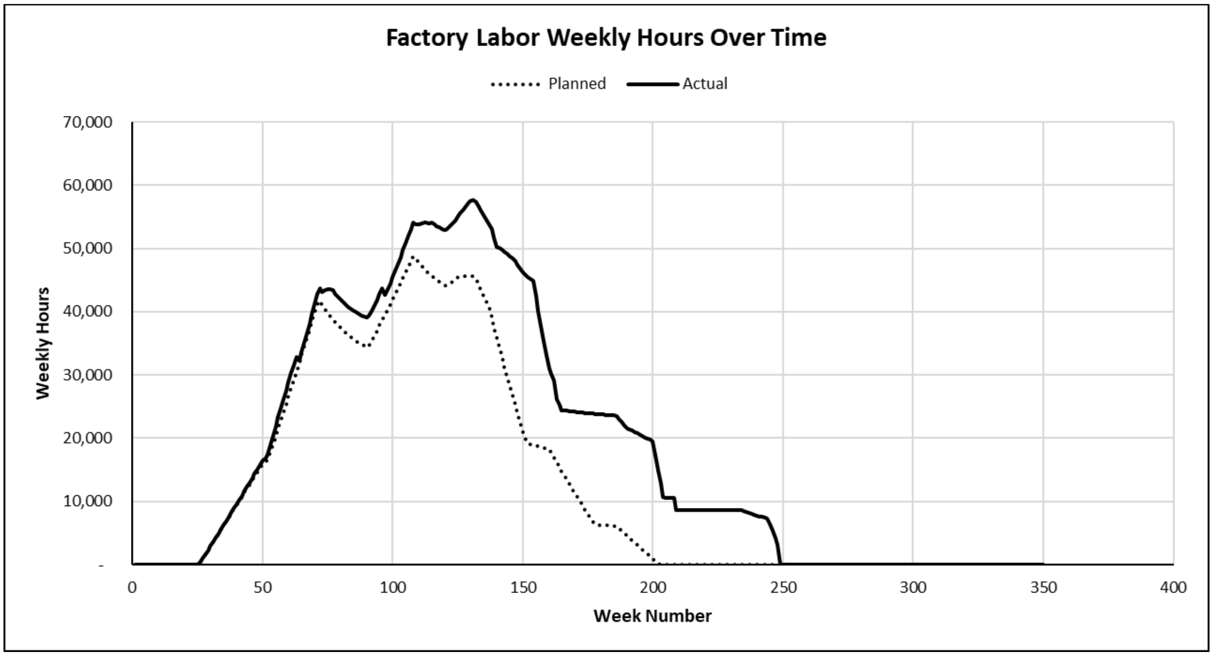

Figure 13. Case 2 – Factory Hours per Week (Planned & Actual) Over Time.

	Hours			Schedule (Weeks)		
Area	Planned	Actual	Variance	Planned	Actual	Variance
Fabrication	1.00	1.12	11.5%	1.000	1.03	3.2%
Component Assembly	1.00	1.24	23.7%	1.000	1.11	10.9%
Mate/Final	1.00	1.77	76.6%	1.000	1.35	35.2%
Flight Line	1.00	2.43	142.6%	1.000	1.55	55.4%
Total Factory	1.00	1.44	44.5%	1.000	1.23	22.8%

Figure 14. Case 2 -- Results by Task Area.

The reason for this substantial growth is the change from 85% to 75% quality has generated a substantial increase in rework activity. As noted earlier, in the baseline scenario, the rework hours are equivalent to 20% of the standard hour content. In the second scenario, rework is now equal to 46% of the standard hours. This represents a growth of 134% over the baseline. It is not surprising, therefore, that cost and schedule are so negatively impacted. This result suggests there is a "tipping point" in the amount of rework – at some point, it becomes so great that it overcomes the resources on the program and forces both a cost and schedule impact.

An examination of the Norden-Rayleigh curve by Alan Jones (2019) provides an interesting cross-check to these results. A Norden-Rayleigh curve represents cumulative resource spend over time – Norden (1970) argued a well-managed development program tends to expend resources over time on a Rayleigh distribution. In instances where the program baseline is underestimated, Jones argues that a "percentage increase in schedule generates an increase in cost equal to the square of the slippage increase." (Reference Figure 15.) Use of this rule of thumb would generate a 51% cost impact against a 23% schedule slippage ($1.228^2 = 1.508$). In fact, the model generated a 45% cost impact. This suggests even our simple model might provide valid answers where actual quality or productivity is less than planned with the concomitant results on cost and schedule.

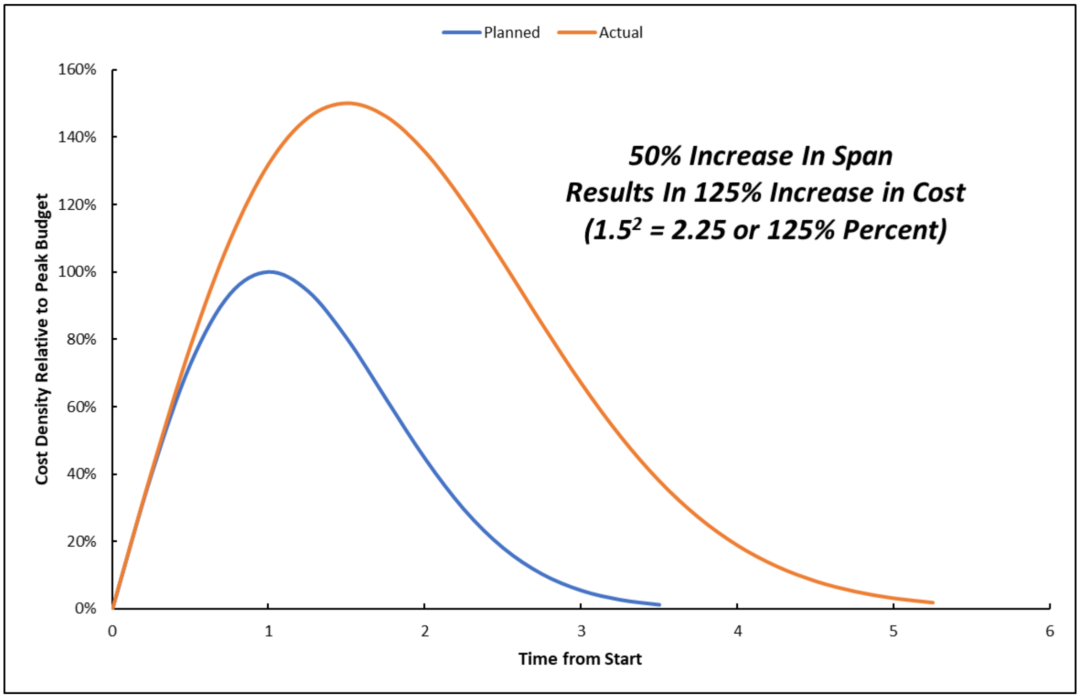

Figure 15. Norden-Rayleigh Curve – Jones' Square Root Rule (Jones, 2019).

Conclusions.

We began by noting the commonly advanced reasons for cost and schedule overruns, and how unhelpful they are to the estimator either *ex ante* or *ex post* when grabbling with an estimate of delay and disruption impacts. Using Cooper's concept of the change cycle and its contribution to the dynamic of disruption, we constructed a conceptual model to utilize this concept and produce an estimate of disruption costs. Through this model, we demonstrated that even a relatively small change in the disruption drivers can – under the right circumstances – plunge the program into a downward spiral as it struggles to overcome a wave of rework as productivity sinks, resulting in the "terminal descent" of this paper's title.

The disruption model shown here is not intended to be more than a proof of concept. Substantially more work would be required to develop a tool useful for real-world, in-depth analysis and use in the program planning process. Some of the necessary improvements include:

- Many of the cost estimating relationships used in this model were taken from McCue (1997). It is not certain that relationships developed for shipyard production are appropriate for aircraft production. Therefore, additional industry and/or company-specific research on the key influencers on rework, productivity and quality would be required.

- Calibration to historical cases of disruption is essential to test the validity of the cost and schedule relationships in the model.
- The model as described here works with limited constraints. The introduction of constraints around facilities, equipment, tooling, and other limiting factors would provide more realistic answers.
- The model does not have a workable means to automatically rephase staffing profiles once schedules are extended beyond the plan. Currently this effort must be done manually.
- The model does not consider the impact of customer driven design changes, which are clearly an important source of increased work scope.
- The model does not consider the impact of part shortages, which are another source of potential disruption.
- This model does not attempt to deal with the impacts of a delay in ramping up manpower, e.g., program funding limitations independent of worse-than-expected quality or productivity.
- Further refinement including a deeper modeling of the manufacturing process, possibly pushing it down to a station level, would be beneficial.
- An *ad hoc* Microsoft Excel spreadsheet model is not optimal for simulation, and leaves much to be desired. The freeware SimQuick Excel-based software package might prove a viable alternative, although it would be difficult to incorporate learning curve impacts. (Hartvigsen, 2016) Otherwise, the model could be created and run using conventional licensed simulation software such as VenSim.

Estimating the cost impacts of delay and disruption is difficult, and typically based on estimating judgement or empirical rules of thumb. These rules of thumb often underestimate the compounding impact of excessive rework and quality issues. A disruption model like that outlined here would better clarify the causes of disruption and its relationship to rework, quality and productivity while producing better estimates. It could aid in a more realistic assessment of potential program cost and schedule risks prior to estimate submittal and program go-ahead. Furthermore, it might prove useful to program managers and planners assessing alternate recovery strategies to known disruptions.

References

Biser, S. & D'Onofrio, R. (2017, October). How Do Courts Measure Damages Related to Project Disruption. Retrieved from https://www.foxrothschild.com/publications/how-do-courts-measure-damages-related-to-project-disruption.

Bliss, G. R. (2015, March). *Report to Congress on Performance Assessments and Root Cause Analyses*. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington DC. Retrieved from https://www.acq.osd.mil/asda/dpc/api/docs/arc/2015-parca-report-to-congress.pdf.

Broadleaf Capital International. (2021, May). Extreme Project Schedule Over Run. Retrieved from https://broadleaf.com.au/resource-material/extreme-project-schedule-over-run/

Carr, G. W. (1946, April) Peacetime Cost Estimating Requires New Learning Curves. *Aviation*, 76-77.

Cochran, E. B. (1960, July-August) New Concepts of the Learning Curve. *The Journal of Industrial Engineering*, 317-327.

Cochran, E. B. (1968) *Planning Production Costs: Using the Improvement Curve.* San Francisco, CA: Chandler Publishing Company.

Cochran, E. B. (1977) The Sources of Disruption to Project Cost and Delivery Performance. *Proceedings of the 6th Annual DoD/FAI Acquisition Research Symposium.*

Cochran, E. B. (1978) Measuring and Predicting Production Disruption Costs Due to Design Uncertainty and Delivery Urgency. *Proceedings of the 7th Annual DoD/FAI Acquisition Research Symposium.*

Cohn, B. (2023, February 2). 99% of Big Projects Fail. His Fix Starts With Legos. *The Wall Street Journal*. Retrieved from https://www.wsj.com/arts-culture/books/lego-megaprojects-bent-flyvbjerg-big-things-11675280517.

Cooper, K. G. (1980, December). Naval Ship Production: A Claim Settled and a Framework Built. *Interfaces*, 10(6), 20-36.

Cooper, K. G. & Reichelt, K. S. (2002). *Quantifying Disruption: Rigorous Analysis of a Challenging Problem*. Paper presented at Project Management Institute Annual Seminars & Symposium, San Antonio, TX. Newtown Square, PA: Project Management Institute. Retrieved from https://www.pmi.org/learning/library/quantifying-disruption-rigorous-analysis-problem-13.

Cooper, K. G. (2022, March). The Dark Side of Projects: Disputes, Contract Claims & Disruption. Retrieved from https://systemdynamics.org/wp-content/uploads/2022/04/The-Dark-Side-of-Projects-Kenneth-Cooper-Slides-March-2022.pdf.

Engwall, R. L. (2001). Learning Curves. In Kjell B. Zandin (Ed.), *Maynard's Industrial Engineering Handbook,* 5th ed (pp. 17.83-104). New York, NY: McGraw Hill.

Flyvbjerg B. (2014, April/May) What You Should Know about Megaprojects and Why: An Overview. *Project Management Journal*, 45(2). Retrieved from https://www.pmi.org/-/media/pmi/documents/public/pdf/research/research-summaries/flyvbjerg_megaprojects.pdf

Fox, J. R. (2011). *Defense Acquisition Reform 1960-2009: An Elusive Goal.* Washington, DC: Center of Military History, United States Army.

Hartvigsen, D. (2016). *SimQuick: Process Simulation with Excel*, 3rd ed. Charleston, SC: CreateSpace.

Johnstone, B. M. (2021, October). How Green Was My Labor: The Cost Impacts of Manufacturing Personnel Changes. *Journal of Cost Analysis and Parametrics*, 10(1), 5-18.

Jones, A. R. (2019). *Risk, Opportunity, Uncertainty and Other Random Models*. London, UK: Routledge.

McCue, T. P. (1997). *The Dynamics of Naval Shipbuilding. A Systems Approach* (Master's thesis). Office of Naval Research, Arlington VA. Retrieved from https://apps.dtic.mil/sti/citations/ADA326883

Norden, P. V. (1970) "Useful Tools for Project Management. In M. K. Starr (Ed.), *Management of Production* (pp. 71-101). Baltimore, MD: Penguin Books.

Siemiatycki, M. (2015) Cost Overruns on Infrastructure Projects: Patterns, Causes, and Cures. IMFG Perspectives, 11. Institute on Municipal Finance & Governance, University of Toronto, Ontario, Canada. Retrieved from https://tspace.library.utoronto.ca/bitstream/1807/82852/1/imfg_perspectives_no11_costoverruns_siemiatycki_2016.pdf

Smart, C. B. (2021). *Solving for Project Risk Management: Understanding the Critical Role of Uncertainty in Project Management*. New York, NY: McGraw Hill.

Wolfe, P.M., Cochran, E. B., and Thompson, W.J. (1980). A GERTS-Based Interactive Computer System for Analyzing Project Networks Incorporating Improvement Curve Concepts. *AIIE Transactions*, 12(1), 70-79.

Biography

Brent Johnstone is a Lockheed Martin Fellow and production air vehicle cost estimator at Lockheed Martin Aeronautics Company in Fort Worth, Texas. He has 37 years' experience in the military aircraft industry, including 34 years as a cost estimator. He has worked on the F-16 program and Advanced Development Programs and has been for 28 years the lead Production Operations cost estimator for the F-35 program. He has a Master of Science from Texas A&M University and a Bachelor of Arts from the University of Texas at Austin.