Using System Dynamics to Analyze the Effect of Funding Fluctuations on Software Development

Citation: Trammell, Travis, Stuart Madnick, and Allen Moulton. "Using System Dynamics to Analyze the Effect of Funding Fluctuations on Software Development." 34th International Annual Conference of the American Society for Engineering Management 2013 (ASEM 2013).

As Published: http://www.proceedings.com/21551.html

Publisher: American Society for Engineering Management (ASEM)

Persistent URL: http://hdl.handle.net/1721.1/98916

Version: Original manuscript: author's manuscript prior to formal peer review

Using System Dynamics to Analyze the Effect of Funding Fluctuation on Software Development

Travis Trammell
Stuart Madnick
Allen Moulton

Working Paper CISL# 2013-06

March 2013

Composite Information Systems Laboratory (CISL)
Sloan School of Management, Room E62-422
Massachusetts Institute of Technology
Cambridge, MA 02142

USING SYSTEM DYNAMICS TO ANALYZE THE EFFECT OF FUNDING FLUCTUATIONS ON SOFTWARE DEVELOPMENT

Abstract ID: 2

MAJ Travis Trammell, Academic Instructor
U.S. Military Academy and MIT Sloan School of Management
West Point, NY <Travis.Trammell at usma.edu>

Stuart Madnick, Professor
MIT Sloan School of Management and Engineering Systems Division
Cambridge, MA <smadnick at mit.edu>

Allen Moulton, Research Scientist
MIT Sloan School of Management and Engineering Systems Division
Cambridge, MA <amoulton at mit.edu>

Abstract
Almost everyone understands that budget fluctuations have an impact on software development, but it is difficult to estimate the magnitude of the impact and all the causes. This paper uses System Dynamics modeling to examine how gaps in funding affect software development productivity and product delivery delay. The results provide decision makers with an improved sense of the negative impacts of budget fluctuations. Two key insights include the "ramp up tax" that slows development and the "gap tax" due to the loss of project-related skill and familiarity when employees are transferred off of a project and then return. The model experiments also compare the different impacts of temporarily stopping a project versus stretching out a project by temporarily reducing the funding level.

Keywords
Software development, project management, funding effects, system dynamics.

Introduction
Project managers faced with funding gaps and erratic funding fluctuations have a good sense of the negative effects on software development productivity and schedule. Nevertheless, in large organizations senior management often employs simplistic linear mental models of those effects. Although it is well understood that budget fluctuations have an impact, it has been difficult to understand the magnitude of the impact and the causes and little research has been conducted to investigate the effects of funding gaps and fluctuations on the software development projects.

We examine ways in which the budget processes and contracting constraints can lead to project funding gaps and fluctuations and some of negative effects of delays in production that often accompany funding fluctuations. These include: first mover advantage, macroeconomic risk, and the inability to meet requirements. Having established some of the negative implications of product delay, a System Dynamics model is developed to examine the impact of funding fluctuations on software development. Next, we briefly describe our model and the results from experiments with the model. The insights gathered from these results provide evidence for software development project managers and program sponsors in large organization to demonstrate to senior decision makers the extent of the negative impacts of budget fluctuations.

Factors Contributing to Funding Uncertainty
Funding gaps and fluctuations have many causes. Although this phenomenon can occur in any organization, we will specifically look at government funded software development efforts. In the U.S. government, funding, and the withholding of funding, is a key control mechanism. High level managers also regularly use funding as a lever to exercise oversight and redirection of ongoing programs. It is not unusual for programs to be temporarily halted for reevaluation before being allowed to proceed. Funding for programs that are generally favored can also be held hostage for political leverage in areas where there is disagreement.

In the U.S. government, funding orginates with Congessional appropriations and is administered by the Executive. Even after appropriations are passed, Congress and the Executive may stop funding with rescissions or sequesters. Since the government runs on an annual funding cycle, the government may need to cease expenditures if disagreements or other delays hold up the passage appropriations laws beyond the October 1 start of the fiscal year. To prevent the government from shutting down entirely, Congress often passes "continuing resolutions" which continue the prior year's appropriations within constraints, such as barring new contracts. Those constraints can lead to funding gaps at the project level. Funding decisions also become much more difficult and uncertain when cutbacks are required to reduce the overall level of the budget. During active military operations in Iraq and Afghanistan, the Defense budget expanded by hundreds of billions of dollars per year. When major conflicts ends, defense budgets fall rapidly, as is presently in progress and anticipated to accelerate over the next ten years.

When the top line budget of an agency is cut, an agency first tries to meet its fixed obligations, including permanent staff. Since software development projects are almost always variable costs outsourced to contractors, they are vulnerable to greater percentage cuts. Higher level agencies also use a mechanism described as an "internal tax" which holds back a percentage (15-30%) of annual funding appropriated for a lower level organization's programs to see how events develop during the year. The lower level organization must then make the case to be given back the funds to proceed with its programs. Often, a significant portion of this "tax" is returned to the agency later in the fiscal year for spending within that year. Unfortunately, this is neither guaranteed nor easy to predict, so there is often need for adjustments during the year depending upon whether and when funding is restored.

Contract turnover can also lead to funding gaps. Government contracts are awarded for periods of performance, most commonly one year. Gaps in funding can develop across different periods of performance or when a contract lapses. During these gaps, work cannot be conducted on the project. Contracting regulations also create problems. Contracts are limited to five years (an initial year plus four optional years). A new bidding process must be conducted if work beyond five years is required on a specific project. This rebidding process can be complicated and contentious, which can also result in a funding gap. The rebidding process can also result in a complete turnover in personnel on the project that leads to repaying the "ramp up tax" discussed below.

Impact of Product Delivery Delay

Funding fluctuations, gaps and reductions inevitably affect the completion date of a software development project. But, determining the magnitude of the impact can be difficult as noted in reports such as Kadish et al (2006). Delays due to funding gaps have important consequences in both the private and public sector for multiple reasons, including time to market, first mover advantage, macroeconomic risk, project cancelation, and inability to meet customer needs.

Time to Market. The length of time it takes for a software product to reach the market has several implications for the developer. The first is that the overall cost of the program is increased. As work stretches out, labor and development costs continue. Additionally, the old system must be maintained for a longer period of time. This maintenance is often more expensive because it occurs as the end of the useful life of the out-of-date system. Finally, while the new system is under development, the organization is not receiving the expected benefits from the system inducing a productivity loss on the organization because it is forced to operate with less capable systems.

First Mover Advantage. First mover advantage refers to the considerable advantages to being first to market with a new product. Lieberman and Montgomery (1988) describe three primary drivers of first mover advantage: technological leadership, preemption of assets, and buyer switching costs. Of these three sources of advantage for the first mover, when government work is examined, buyer switching costs appear to carry significant weight. It is common parlance in government circles that the most difficult thing to accomplish is to end an established program. While the causes of this so called "refusal to die" are not the focus of this research, we accept this staying power as given and consider the resulting implications for buyer switching costs. For example, there are the costs of purchasing the new system, retraining the workforce, productivity loss during the transition, and failure risk. In the government arena, all of these factors matter greatly but are also increased because of the organizational inertia previously discussed.

We encountered an example of the effect of governmental first mover advantage regarding software systems in the competition between the Defense Intelligence Agency's (DIA) Human Intelligence (HUMINT) Online Tasking and Reporting (HOT-R) and the Combined Information Data Network Exchange (CIDNE). CIDNE was developed as a specialized tool to deal with the frustrations of commanders in Iraq about the inability to share operational information or gain a more complete understanding of the situation on the ground. By government comparison, CIDNE was created and launched in record time to generally positive responses. Most importantly

CIDNE garnered significant support from senior decision makers in the U.S. Central Command, most notably General David Petraeus. While CIDNE performed well as an operational tool and was valuable in maintaining intelligence archives in theater, it was not designed as a complete HUMINT reporting system. The DIA, acting as the lead executive agent within the U.S. government for HUMINT, was tasked with standardizing HUMINT reporting across the intelligence community. In pursuit of this objective, DIA was developing HOT-R. This system focused solely on HUMINT reporting and did not replicate any of the CIDNE functionality in the operational realm. However, HOT-R encountered significant resistance within U.S. Central Command because of the perception that it duplicates or is meant to replace CIDNE in some way. Outside of Central Command, the HUMINT community has seen HOT-R as superior product upgrade to procedures previously in use. The key issue is the order of system delivery. It appears plausible that had HOT-R been delivered first, all the community of users would have accepted the system. However, CIDNE had the first mover advantage in Central Command. Such switching costs can result in inconsistent systems in different parts of the government and increase problems in integrating information.

Macroeconomic Risk. A second major incentive for the government to get products to market and in the hands of customers faster is the issue of risk. Because of macroeconomic and business cycle effects, it is advantageous to complete projects as expeditiously as possible. Stretching a project out over longer periods will make budgeting for the project much more difficult and expose the project to more economic risk. Government revenue is directly affected by the business cycle and government projects are generally less stable during economic downturns.

Project Cancelation. Delays in major Defense projects can lead to additional scrutiny and potential cancelation. A delay of six months to a year must be reported as a "significant change." A delay of over a year must be reported a "critical change." Both types must be reported to Congress. When a "critical change" is reported, a complete re-estimation and re-evaluation of the project must be performed and certified. Work on the project must stop unless the re-evaluation proves the value within 60 days. If the project is approved for continuation, it may be restarted after a gap.

Inability to Meet Customer Needs. Finally, the longer a project takes to reach the market, the greater the likelihood that the original requirements for the product have changed. The pace of technological advancement continues to increase along with the introduction of disruptive technologies. This risk continues to increase as newer and more innovative products are developed. A similar pace of change has been seen within the security community. During the Cold War, the focus of the national security and intelligence community did not change greatly, allowing systems to have a much longer development time. Given the dynamic security environment that exists now and likely in the future, products that take too long to develop will not arrive in time to meet the threat that was the impetus for the original development.

System Dynamics Modeling Of Software Development

The System Dynamics (SD) approach is based on identifying individual causalities and how they combine to create, often non-linear, feedback loops that can be the causes of counter-intuitive outcomes (*viz.* Sterman, 2000). This approach leverages deep, but often isolated and fragmented, knowledge. The core of the SD modeling strategy is representation of system structure in terms of stocks, flows, and the causal mechanisms that govern their rates of change. In this connection, feedback loops are the building blocks for articulating the causality represented in these models. The interaction among the various feedback loops in a model can represent and explain system behavior. SD models improve management understanding by explicitly capturing the complex interactions among many feedback loops, rejecting notions of linear unidirectional cause-and-effect, and allowing the analyst to view a complete system of relationships whereby the 'cause' might also be affected by the 'effect' and enables analysts to uncover 'hidden' dynamics. SD also facilitates understanding long term effects of short term policies or behaviors.

Additional characteristrics of SD include: (1) *Objective input*: Ability to utilize data to determine estimates for parameters affecting the causality of individual cause-and-effect relationships, (2) *Subjective (expert) judgment*: Ability to represent and model cause-and-effect relationships, based on expert judgment, even when detailed data does not exist, (3) *Intentions Analysis*: Ability to identify the long-term unintended consequences of policy choices or actions taken in the short term, and (4) *Tipping point analysis*: Ability to identify and analyze "tipping points" – where further incremental changes lead to significant impacts, and (5) *Transparency*: Ability to explain the reasoning behind predictions and outputs of the SD model.

System Dynamics has been used to model project management (Lyneis & Ford, 2007) and a diverse array of situations, including crisis in the world oil market, dynamics of arms races, threats to sustainability, combat

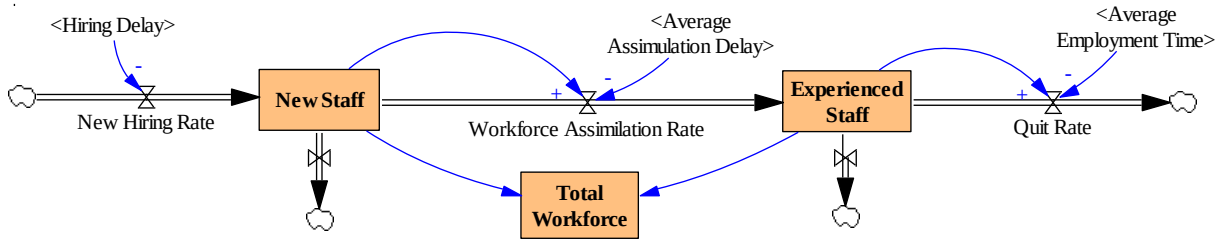
vehicle accidents (in non-combat settings), impact and usage of social media on uprisings, and stability and instability of countries. An example of the latter can be found in (Choucri, et al., 2006).

System Dynamics Model of Government Funded Software Development Performance (GFSDPM)

The model developed in this study relies on prior research applying SD to the factors affecting software development productivity, particularly the Software Development Performance Model (SDPM) in Abdel-Hamid and Madnick (1991). Although management actions that led to variations in total project cost were studied in SDPM, changes in budget constraints was not a central focus of that work.

Early on in the examination of this issue, it became clear that the effect of budget fluctuations would be most directly seen in staffing adjustments. Exhibit 1 shows a section of the SDPM model related to staffing. The values for many of the model parameters used here were based on those developed in the research for the SDPM model and confirmed through discussions with subject matter experts involved in government software development.

Exhibit 1. SDPM Staffing Model Sector from Abdel-Hamid and Madnick (1991), p.64.



The three rectangles in Exhibit 1 represent stocks, in this case quantities of staffing resources assigned to a software development project. The double lines represent flows, which transfer material between the stocks (or the outside world, represented by little clouds). The valve symbols along the flow lines represent rates, which control the flows. The single line arrows represent information transfers from one element of the model to another.

While staff are individual people, we abstract staffing resources to continuous variables measured in units of full time equivalent people. The Total Workforce represents the complete staff assigned to the project. In order to capture the learning effect of experience on the project, Total Workforce is broken down into two sub-stocks: New Staff and Experienced Staff. New Staff represents the fraction of the total staff that are new to the project. Over time, New Staff learn about the project and become assimilated into Experienced Staff. The model depicts this effect as a flow controlled by the Workforce Assimilation Rate, which is modeled as a first order exponential delay with an Average Assimilation Delay. One point that is important to note is that "experience" as used in the model is project-related experience, not career experience or general software development skills. It is also important to note that individual workers do not move between New Staff and Experienced Staff. As an individual's level of experience on the project grows, the Experienced Staff stock will increase and the New Staff stock will decrease. Total Workforce will remain unchanged from the assimilation effect.

Model Overview

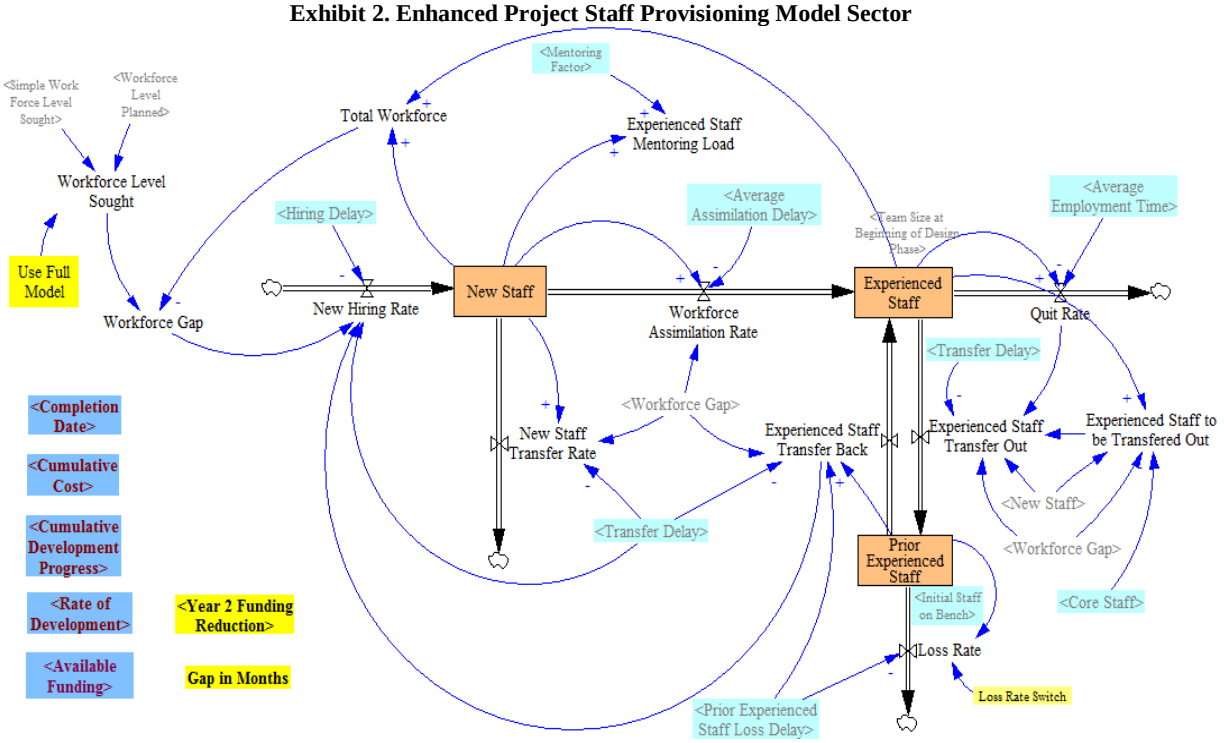
The GFSDPM model has four major sectors: (1) Project Staff Provisioning, (2) Project Staff Planning, (3) Funding Process and Expenditure Accounting, and (4) Software Development. Because staffing levels were the critical connection between funding and performance, the Project Staff Provisioning is the most significant. The other sectors of the model were simplified in order to isolate the effects of budget fluctuations. The Software Development sector begins with a required amount of work to be accomplished. It then compares that with the amount of work already accomplished to determine the remained amount of work to be completed. In the Project Staff Planning sector, based on the due date of the project, the model determines the required amount of staff to complete the project on time. It then checks if funding is available for the *Workforce Level Sought*. Funding determines the workforce ceiling, which constrains the *Workforce Level Sought* amount sent to Project Staff Provisioning sector, which models the hiring and training process discussed next. This is a dynamic model, so these steps are repeated at each time interval, also illustrated later.

Enhanced Project Staff Provisioning Model Sector

Our enhanced Project Staff Provisioning model sector is shown in Exhibit 2. The key addition to the original SDPM model, which has been incorporated into the Project Staff Provisioning sector, is the *Prior Experienced Staff* stock.

Discussions with Agency personnel were the impetus for the creation of this stock. The most glaring factor that emerged was the length of time required to get new personnel cleared to work on a project. This clearance process consisted of two major steps. The first was the standard U.S. government background investigation. This process consists of an extensive background investigation and is mandatory for all personnel with access to Secret or Top Secret material. The background investigation can take up to a year and a half depending on outside factors. Furthermore, for this agency, all software development is conducted within secure areas that require additional special clearance to access, beyond a standard Secret or Top Secret clearance. This special clearance is administered directly by the Agency and requires an additional background check. The seemingly redundant Agency clearance is designed to provide additional security, but adds a further delay to bringing new personnel on board. Of note to our research, contractors, who do most of the software development work, have a lower priority for investigations than do military personnel or government employees. Lower priority can result in even further delays as contractor investigations are pushed to the bottom of the queue. The investigation process is also greatly affected by funding considerations. For example, in May of 2011, all investigations were placed on hold due to funding uncertainty during Congressional budget negotiations. Once an employee finally receives both a U.S. government security clearance and the special DIA clearance, the process is still not complete as he or she must still receive building access, referred to as getting "badged," to the agency's facilities. Getting badged is rarely a lengthy process but it remains another level of complexity in an already tedious process.

Exhibit 2. Enhanced Project Staff Provisioning Model Sector



The net result is that personnel who have completed the process described above and received a "badge" become extremely valuable to the organization. Great efforts are taken to avoid the loss of these "badged" personnel. One of the most common methods to avoid the loss of badged personnel is to rotate them among different projects when funding for one project if reduced or exhausted. This method is referred to as placing employees "on the bench." If employees were not placed on the bench, they would lose their badge for building access. Bringing them back onto a project would be greatly delayed by redoing some of the clearance procedures. The group of employees placed on the bench during periods of reduced funding is modeled as the *Prior Experienced Staff* stock shown in Exhibit 2.

While on the bench, these employees begin to experience a loss of familiarity and knowledge with respect to the specific project. For purposes of our experiments with the model, the *Loss Rate* of project-specific knowledge for the *Prior Experienced Staff* is modeled as an exponential decay with an average time of four months. This "de-assimilation" time period was suggested by subject matter experts (SMEs) as being analogous to the average time as for *New Staff* to assimilate and become *Experienced Staff* as described above.

When staff reductions are needed, the model first reduces *New Staff* before cutting into *Experienced Staff*. Similarly, when funding is increased, employees are drawn first from the *Prior Experienced Staff* before the model attempts to hire *New Staff*. The number of people on the bench is also reduced by a small number of employees quitting the organization while on the bench, shown as the P*rior Experienced Staff Loss Rate*.

Funding Process and Expenditure Accounting

One element that was absent from the original SDPM model is the possibility of funding fluctuations. The SDPM model was primarily geared toward commercial software development, without consideration of budget fluctuations. The assumption driving the SDPM model is that as long as a commercial project is meeting expectations, it will remain funded unless the project is modified for some other reason. In our enhanced model, the Funding Process and Expenditure Accounting portion of the model is designed to produce some of the funding scenarios actually experienced by developer teams. The scenarios identified for testing include: (1) continuous funding, (2) funding gaps of varying lengths, and (3) reductions in funding that does not produce a funding gap.

The Project Staff Planning portion of the model is deliberately simplified to estimate the staffing level necessary to complete the project based a number of work units required for the project and the desired completion date. The model attempts to maintain this staffing level as long as funding is available.

The experience level of the total staff comes into play in the Software Development sector. Based on research underlying SDPM, productivity is increased by the project-related experience level of the staff (modeled by the *Experienced Staff* stock). Additionally, the model addresses the reality that the amount of production work conducted by experienced staff is reduced by the need to train and assimilate new staff. The relative productivity of new staff and the ratio of new staff needed for training are controlled by parameters which can be varied. Values used are based on the research done for the SDPM model and confirmed by the subject matter experts in this study.

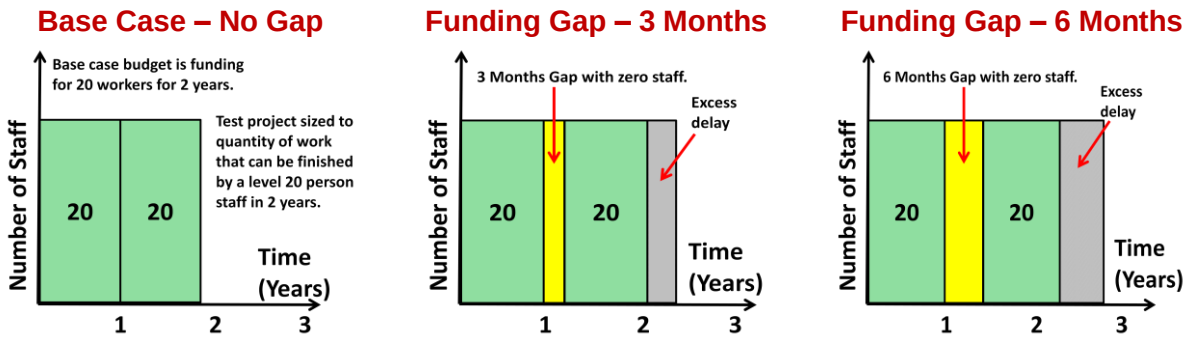
Experiment Design

As previously mentioned, funding levels directly impact the amount of staff assigned to the project. The base case for our experiments was set to a staff number of 20 workers. The level of work required was set to be completed in exactly two years, 480 working days in the model (20 work days per months). For this experiment, we assumed that the funding adjustments occurred during Year 2 only. In Year 3, the Year 1 staffing level of 20 workers is resumed. All of the funding test cases are compared based on the amount of spillover of work into the third year that results from funding modifications. The overall experimental design is shown in the diagrams in Exhibits 3 and 4.

Funding Gap Test Cases: No Funding Gaps, Three Month and Six Month Funding Gap

The No Gap case, shown on the left in Exhibit 3, is the baseline test in which sufficient funding is provided to complete the project with 20 workers within two years. The two additional test cases consider a three month or six month funding gap following the end of the first year of the project, as shown in the center and right of Exhibit 3. During the gap, all experienced staff are transferred to the bench and any new staff are released. This results in no work being conducted on the project during the funding gap period.

Exhibit 3. Funding Gaps Experiment Design

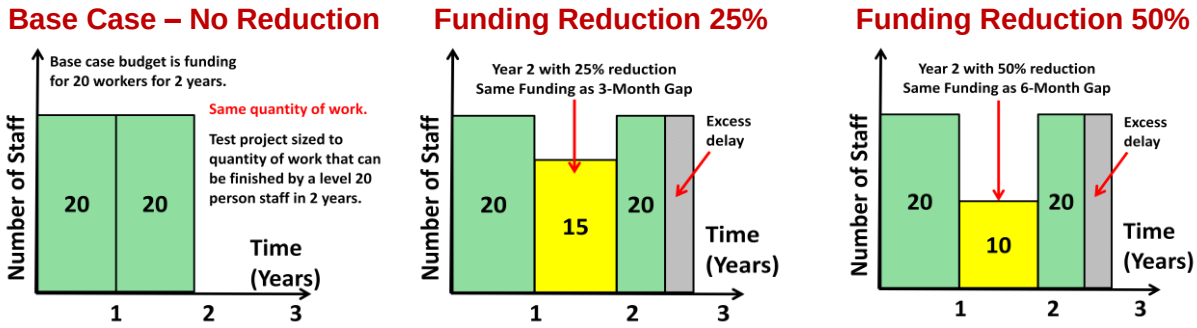


Funding Reduction Test Cases: Reduction of Funding by 25% and 50% in the Second Year

We also evaluated an additional type of funding fluctuation, percentage reductions in funding. The cases chosen for modeling were a 25% or a 50% reduction of funding in the second year of the contract. For example, if the first year funding was $5 million, for the 25% reduction case, the second year funding is reduced to $3.75 Million. The

reduction of 25% was chosen in order to compare the results to a three month funding gap. Assuming constant funding over the entire year, a three month funding gap reduces spending by 25% for that year. The funding 25% reduction results in a commiserate 25% reduction in workforce (15) during the second year as illustrated in the center case of Exhibit 4. The third case, shown on the right side of Exhibit 4, models a 50% reduction of funding in the second year of the contract, second year funding is $2.5 million, and this reduction will be compared with the six month gap. The reduction reduces the workforce to 10 employees in the second year.

Exhibit 4. Three Funding Reduction cases: No Reduction, 25% Reduction, 50% Reduction



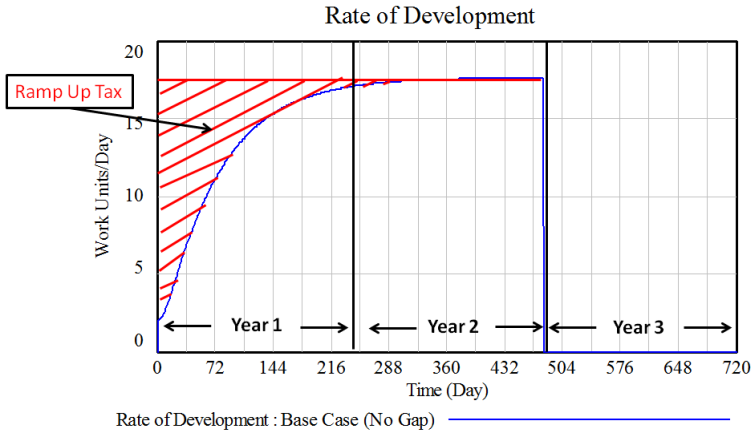
Test Results

Two notable insights were gleaned by developing and running our model for these various cases. These insights include the analysis of what we refer to as the "Ramp Up" tax and the "Gap Tax," both described in the following.

Ramp Up Tax

The developers and development managers, who provided much of the information and structure for our model, noted that the rate of development on every project took time to ramp up and then stabilize. In this study,we refer to this effect as the "ramp up tax." The ramp up tax is a result of the time taken to hire new staff combined with the time taken for these new staff members to assimilate on to the project. The SMEs consulted in this study suggested a shape illustrated in Exhibit 5.

Exhibit 5. The "Ramp Up" Tax Effect on Rate of Development



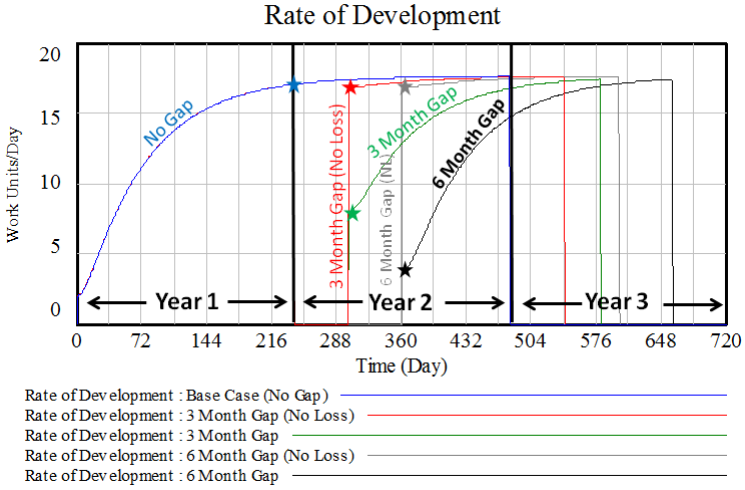
Assimilation Effect

Assimilation is a vital important concept for explaining the operation of our model. Assimilation describes the time it takes for an employee who is newly assigned to a project to become familiar with that project. We assume that the new hire is a trained professional but is new to this particular project – thus some time must be invested to learn the goals of the project, the current status of the project, project specific knowledge (e.g., software standards and libraries to be used), his/her specific responsibilities, etc. Assimilation exacerbates the ramp up tax phenomenon because new staff are less effective than experienced staff due to lack of project-specific knowledge. Furthermore, there is a documented need for experienced staff to spend significant time helping to assimilate new staff, which in turn reduces the productivity of the experienced staff. The chart shown in Exhibit 5 illustrates the ramp up tax in the no funding gap base case. The rate of development is measured in work units per day plotted over the life of the project.

Due to the delays in hiring staff and the assimilation process, it takes about a year for the team to attain its full stable level of productivity, as shown in Exhibit 5. The upper left corner area, shaded in red, is what we refer to as the "Ramp Up Tax." In the ideal case, with a team of 20 people, we would expect the steady state productivity level to be 20 work units per day, but even in steady state, because of natural turnover, a number of employees leave

the project and new hires are added to replace them – which continuously reduces the productivity somewhat to a level of about 17 work units/day.

Exhibit 6. Effect of Funding Gap on Rate of Development



Development Productivity Drop from Experience Loss during Gap		
Test Case	**Productivity After Gap**	**% Loss in Productivity**
Base Case (No Gap)	**17.06**	**0%**
3-Month Gap	**8.08**	**53%**
6-Month Gap	**3.76**	**78%**

Project Experience Loss Effect

While on the bench, employees lose familiarity with the project, because while on the bench they are actually assigned and working on other projects. As a result, the focus and daily involvement that is required for employees to remain effective on complex projects does not occur while on the bench. The longer the employees remain on the bench the greater the loss of skills and familiarity with the project. This results in a reduced rate of development after a funding gap as shown in Exhibit 6.

Ignoring Experience Loss: The Rate of Development as indicated in Exhibit 6 drops only slightly when no experience loss is modeled. In that case, the drop in the Rate of Development following a gap only occurs because the quit rate still reduces the "Experienced Staff" stock, though only slightly.

Including Experience Loss: Including the experience loss effect, funding gaps significantly reduce productivity when work is resumed after the gap. The longer the funding gap, the larger the productivity loss following the gap. The longer an worker sits on the bench, the more experience loss they suffer. The decreases in productivity are shown in the lower green (3-month gap) and and the lowest black (6-month gap) lines in Exhibit 6. In the base case, where there is no funding gap, the team starts the second year with its steady state productivity of 17 work units per day. After a 3-month gap, initial productivity level for the second year drops 53% to 8. work units per day. After a 6-month gap initial productivity drops 78% to under 4 work units per day.

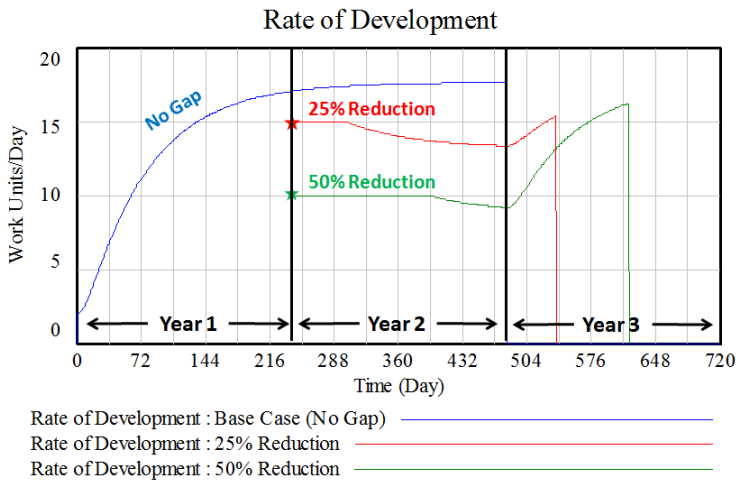
Effect of Funding Gap on Development Progress

The reduced rate of post-gap development productivity shown in Exhibit 6 leads to reductions in development progress compared to the no gap base case and delays in project completion beyond just the length of the actual funding gap time. This is in contrast to the naïve, though frequent assumption, that a 3 month funding gap merely delays the completion of a project by 3 months. We refer to this additional delay as the "Gap Tax." The model shows that a 3-month gap results in a 5-month delay and therefore a Gap Tax of 2 months in additional delay. For a 6-month gap, the total delay is 9 months and a Gap tax of 3 months in gap-adjusted completion delay.

Effect of Funding Reductions on Development Progress

A similar analysis can be conducted for the funding reduction scenarios in the experimental second year – accomplished by a percentage reduction in staff. Productivity in the base case, along with the cases of a 25% and 50% reduction in second year funding, are shown in Exhibit 7. The first significant difference between Exhibit 6 and Exhibit 7 is that productivity loss is much less in the case of funding reductions versus funding gaps. This occurs because there is essentially no "gap tax," i.e. no loss of knowledge, since none of the staff that are continuing the work into the second year needed to be put "on the bench."

Exhibit 7. Effect of Funding Reductions Development Progress



Development Productivity Drop from Experience Loss during Reduction		
Test Case	**Productivity After Reducton**	**% Loss in Productivity**
Base Case (No Gap)	**17.06**	**0%**
3-Month Gap	**14.98**	**12%**
6-Month Gap	**9.99**	**41%**

Effect of Funding Reductions on Development Progress

The reduction in funding for the second year, and the corresponding reduction in staffing and development productivity, does affect the rate of development progress and completion date, which like the funding gap, requires development efforts to continue into the third year, as shown in Exhibit 7. Model results indicate a non linear relationship between the size of the funding reduction and the resulting delay in the project. A 25% funding reduction leads to a 12% loss in productivity, whereas a 50% funding reduction causes a 41% loss. Again, this is impacted by the prioritization of New Staff when releasing personnel. However, once the funding reduction reaches a certain level, all New Staff have been released and Experienced Staff must then be released. This explains the non linear relationship as funding reduction and delay increase and move closer to a one to one relationship.

Comparison of the Effect of Funding Gaps and Funding Reductions on Total Delay

The effect of funding gaps and funding reductions are compared in Exhibit 8. These results suggest that, in general, funding reductions produce shorter delays. That is, given a choice between a funding gap and a funding reduction (of the same funding amount, i.e., 3 months vs. 25%, 6 months vs. 50%), it is better to have a funding reduction and keep a minimal amount of staff assigned to the project to keep work going in order to minimize the effect of the ramp up tax and the creation of a gap tax.

In particular, whereas a 3-month funding gap actually extends the completion date of the project by 5 months (the extra 2 months coming from the gap tax), a 25% reduction only extends the completion date by 3 months. This surprisingly good result comes from a short-term boost that the project gains by releasing the New Staff first. The reduced New Staff level enables the Experienced Staff to focus all their efforts on the development since there is less training of New Staff to be done. Of course, in a project of a much longer duration, there is a penalty to be paid for this boost since there are fewer trained Experienced Staff in the long run.

.

Exhibit 8. Comparison of Project Delay Effects of Funding Gaps vs. Reductions

Test Case (2nd Year Gap)	Total Project Delay
No Gap	**0**
3-Month Gap	**5 months**
6-Month Gap	**9 months**

Test Case (2nd Year Reduction)	Total Project Delay
None	**0**
25% reduction	**3 months**
50% reduction	**7 months**

Another interesting observation is that while there was no extension of the completion delay in the 25% reduction compared with a 3-month gap, a 50% reduction does result in a 7 month delay, which is more than the corresponding 6-month funding gap. This is primarily due to the limits on short-term boost that can be gained by releasing New Staff. Once all the New Staff have been released, (1) Experienced Staff must be released to get down

to a 50% funding reduction, which significantly reduces productivity and (2) a much higher ramp up tax must be paid when the project is returned to full staffing.

Conclusions And Suggestions For Future Research

Budget fluctuations are a fact of life for anyone working in the government and many other organizations. As noted earlier, there are many causes of budget fluctuations. The budget fluctuations that result are, however, not without cost. This research provides a starting point for investigating the effect of these fluctuations and methods for better understanding and attempting to minimize their cost.

The results match a general intuition that stopping and starting projects, i.e. creating funding gaps, is more costly than continuing work. But this paper makes the "intuition" more concrete. The existence of the ramp up tax, and consequent gap tax (essentially a re-ramp tax), provides the reasoning for this effect. While these negative effects are rarely quantified and measured, we have anecdotal reports that funding gaps have resulted in delays similar to what is predicted. These research results have also been used to good effect in one agency to forestall impending gaps from contracting problems.

Additionally, the effect of funding reductions can be minimized by the practice of releasing New Staff first. Again, this matches our intuition that any organization can, for a short time, focus all effort on production at the expense of training and maintenance. However, this will have negative long term consequences.

Overall this research provides a base for future investigation of this topic. The logical next step will be to broaden the funding scenarios. Additionally, more sophisticated production and development models should be developed to illustrate other organizational effects and development approaches, such as "agile" development. Finally, although the details of this research were based on a study of a government agency, private sector organizations can also encounter similar "start and stop" situations in software development – and this research provides a basis for further study of those cases.

Recommendations

Placing employees on the bench is an effective way to avoid the extensive security clearance delays that would result from trying to bring in new people from outside the agency. However, after a period on the bench, the skills and knowledge levels can decrease to a point where they can be essentially classified as new staff to the project and must be fully assimilated back on to the project.

As mentioned above, it is interesting to note that a funding reduction in the second year does not produce an equal reduction in funding for the overall project. This occurs because releasing New Staff first has only a modest impact since (a) they were not yet fully productive and (b) they reduced the productivity of the Experience Staff that were helping them to assimilate onto the project. However, failing to retain and train New Staff may have negative effects on future projects but this effect was not investigated with this model.

A key finding is that, given a choice, a reduced funding level is preferable to a funding gap. During periods of reduced funding, a core group of experienced staff are retained and production continues at a reduced level.

References

Abdel-Hamid, T. K., & Madnick, S. E. (1991). *Software Project Dynamics: An Integrated Approach.* Englewood Cliffs, NJ: Prentice Hall.

Choucri, N., Electris, D., Goldsmith, D., Mistree, D., Madnick, S., Morrison, J. B., . . . Sweitzer-Hamilton, M. (2006). Understanding & Modeling State Stability: Exploiting System Dynamics. *IEEE Aerospace Conference.* Big Sky, MT.

Hicks, K. (2011, Sept 14). Defense Strategy in a Time of Budget Austerity. *presentation at MIT by the Deputy Under Secretary of Defense for Strategy, Plans and Forces*.

Kadish, R., Abbott, G., Cappuccio, F., Hawley, R., Kern, P., & Kozlowski, D. (2006). *Defense Acquisition Performance Assessment Report.* https://acc.dau.mil/adl/en-US/18554/file/733/Defense%20Acquisition%20Performance%20Assessment%20Report%202006.pdf.

Lieberman, M., & Montgomery, D. (1988). First-Mover Advantages. *Strategic Management Journal*, 41–58.

Lyneis, J. M., & Ford, D. N. (2007). System Dynamics Applied to Project Management: A Survey, Assessment, and Directions for Future Research. *System Dynamics Review*, 157-189.

Sterman, J. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World.* Boston: McGraw-Hill/Irwin.

About the Author(s)

Travis Trammell, MBA MIT and MPA Harvard Kennedy School of Government. Major U.S. Army, Academic Instructor, United States Military Academy Department of Systems Engineering. Courses taught include Project Management, Computer Aided Systems Engineering, and Military Officership. Current research focused on evaluating the U.S. nuclear weapons counterproliferation effort using System Dynamics. He has served for ten years as a U.S. Army Officer in a variety of positions including: Platoon Leader, Aide de Camp, and Troop Commander. Deployed to Baghdad and Wasit Province in support of Operation Iraqi Freedom. His military awards include the Bronze Star Medal, Meritorious Service Medal, and Army Commendation Medal with Valor Device. He is certified as a Project Management Professional (PMP) by the Project Management Institute.

Stuart Madnick, PhD MIT, John Norris Maguire Professor of Information Technologies in the MIT Sloan School of Management and Professor of Engineering Systems in the MIT school of Engineering. He received his MBA and Ph.D. in Computer Science from MIT and has been on the MIT faculty since 1972 and head of IT Group more than twenty years. He is co-Director of the PROductivity From Information Technology Initiative and co-Heads the Total Data Quality Management research program. He is the author/co-author of over 300 books, articles, or reports on topics such as integrating information systems, data semantics, and strategic use of IT. He was a key designer of IBM's VM/370 operating system and Lockheed's DIALOG information retrieval system and has been the co-founder of high-tech firms, including Intercomp, Mitrol, iAggregate. Dr. Madnick has been a Visiting Professor at Harvard, Nanyang Technological University, University of Newcastle, Technion, and Victoria University

Allen Moulton, SB MIT. Research Scientist, MIT Sociotechnical Systems Research Center. Current research includes the use of system dynamics to problems of managing application of information technology to achieve enterprise value and to innovative methods for detecting low frequency events. Other research focuses on critical success factors in achieving economic benefits from information quality within the enterprise and across multiple enterprises as well as methodologies for resolving semantic data quality issues and applying automated reasoning technology resolving semantic differences in the interchange of information among autonomous, heterogeneous data sources and receivers. He has previously been involved in research on centralization and decentralization of information systems and on applications of information technology to substantive foreign policy analysis. He is co-author of the book Managing International Conflict: from Theory to Policy, which includes the CASCON computer software for analyzing international conflict.

Acknowledgments

The authors would like to thank Mr. Richard Herrick, Mr. Kelly Hughes, Ms. Sarah Kingensmith, and Mr. Ian Macurdy at the Defense Intelligence Agency and Mr. Douglas Marquis at MIT Lincoln Labs, for their valuable insights into the issues addressed in this paper. The work reported herein was supported, in part, by the MIT Lincoln Laboratories and the Defense Intelligence Agency (DIA) under the "Understanding the Challenges to Net-Centric Systems and Mitigating Approaches" project, MIT Lincoln Laboratory contract 16-11-TCO-0013. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not reflect the official policy or position of MIT Lincoln Laboratory, the Defense Intelligence Agency or the Department of Defense.