December 8, 2008

The Project Manager Learning Sim

Jim Hines and Martha Miller

Summary. This paper describes the Project Manager Learning Sim, its purpose, and its impacts on a sample of ten users. The Sim contains a simple project-dynamics model, and gives users access to the model via a game-like user interface. The Sim was developed in order to enable users to discover for themselves several fundamental insights into managing real world projects, the most fundamental of which is that progress data is critically important to managing a project successfully. Interviews with the ten users indicate the Sim can accomplish this fundamental purpose among those who haven't already gained this insight through real-life experience.

I. The Sim: Purpose, Design and Insights

Background.

A continuing issue for those seeking to help program managers within NASA has been obtaining numerical time series data. In one case, managers were willing to wait six months until time series (other than budget expenditures) became available via a new online system. The delivery date of the system slipped at a pace that approximately maintained the six-month time-to-delivery for more than two years. Many NASA managers, though not all, remained surprisingly patient.

Among the NASA managers who did NOT remain patient, was a group who had been strongly influenced by system dynamics project modeling. A system dynamics project model allows one to use the time series data that is typically available on a project (i.e. data on cost, progress, overtime, etc) to estimate aspects of a project – such as yet-to-be-discovered rework – which by definition are not directly observable.

A system dynamics project model allows indirect estimation because it represents the physical flows and causal connections fundamental to the model. For example, every system dynamics project model contains a representation of the rework cycle[*] (see Figure 1), wherein work being

[*] This structure is often called the "work accomplishment structure", and was invented at the system dynamics consulting firm Pugh Roberts in the early 1970's to understand the causes of cost-overruns in a large DOD ship-building project. The creators of that first "project model" were Henry Weil, Ken Cooper, and David Peterson, who later founded Ventana Systems, Inc. The structure today is widely known within the field of system dynamics and has been used in thousands of project models in a huge variety of settings

done is split in two: Part flows into *Correctly Done Work* and part flows into *Undiscovered Rework*, a hidden backlog. After discovery, rework can commence, but, of course, not all of it will be done perfectly; a portion of the rework will itself have to be reworked. Work can cycle many times between the visible "backlog" of *Work to Do* and the hidden backlog of *Undiscovered Rework*.

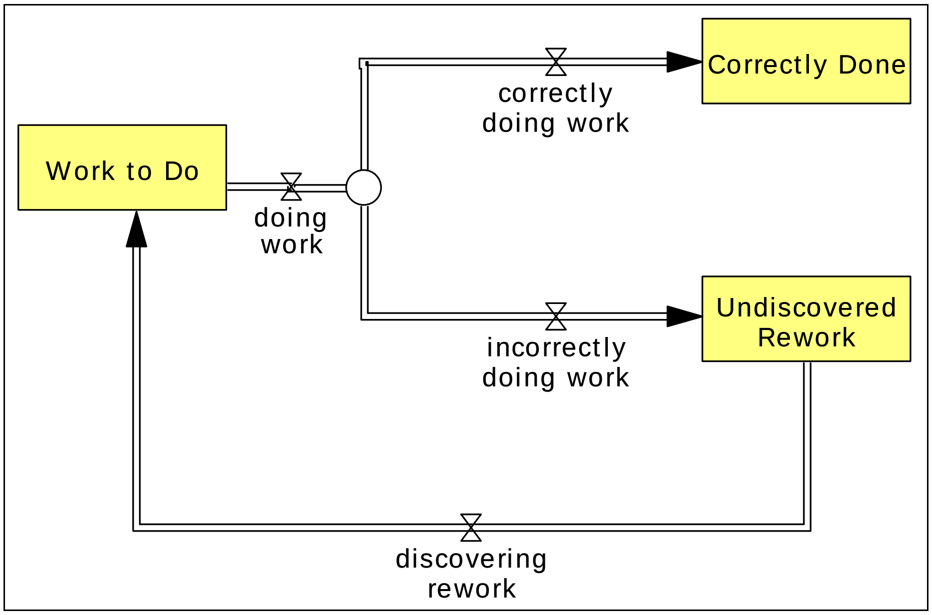


Figure 1: The Rework Cycle. This diagram uses a "fluid dynamics" analogy. Boxes represent pools or accumulations (mathematically, boxes are integrals). Double-line arrows represent pipes, and the hour-glass shapes represent valves (mathematically, the pipes are derivatives).

For example: $WorkToDo_t = \int_0^t discoveringRework_t$.

The critical point is that Undiscovered Rework can never be directly observed, after all it's *undiscovered.* But, if data are available on the "surrounding" structure – e.g. the people working, the work that *appears* to be getting done, the rework that that is getting discovered – it's possible to statistically deduce or estimate the amount of *Undiscovered Rework* and to predict how much, when, and where rework will be discovered in the future.

The value of time series data from directly observable aspects is significantly enhanced when those data also can be used to illuminate aspects of the project that are NOT directly observable. Hence, the NASA managers who were most aware of system dynamics project modeling were among those who were least patient with the ever-receding completion date of the promised data-delivery system.

---

including large software projects, construction projects, and aerospace projects to name just three application areas.

Understanding why some managers were more impatient than others is one thing. But understanding why many managers were actually *patient* for data is another. Perhaps the added benefit of illuminating previously unknowable aspects of a project is required in order to develop a preference for numerical data retained as time series. Perhaps numerical time series data per se do not provide significantly more information than simply talking to people about how a project seems to be progressing. It's quite possible that some managers have this viewpoint. However, the viewpoint does not characterize all of NASA nor all of its history. The Saturn program, for example, carefully tracked and summarized for its management team progress on part design, fabrication, and testing. In fact, at one point in that program, the managers had 800 major entities being tracked by 90,000 key events taking place all around the country.
A very different explanation for "data patience" is that managers whose access to data is limited for long enough get used to that limit and ultimately come to believe that their limited data is sufficient. This explanation involves a gradual habituation and so might require a long project – a project like Constellation.

Accepting a limited data set as sufficient may involve imbuing it with more information than it in fact contains. In at least one instance, a manager believed that an available time series, expenditures, contained approximately the same information as an unavailable series on progress would have contained. Clearly, in a situation involving rework and untested assumptions about productivity and work-quality, this is not the case. And yet, the belief appeared to be firmly held.

Firmly held beliefs can be changed through experience. In the context of a small scale or unimportant project, the value of the learning may exceed the cost of failure. But, this is not the case with Constellation, and the NASA managers concerned by the paucity of data looked for ways to provide a "learning experience" without the cost of failure. Not surprisingly, they considered the possibility that the project model itself could help their colleagues gain (or regain) their impatience for absent data. Ultimately, they teamed with Ventana Systems, Inc. to design and implement a game-like Learning Simulation based on (a highly simplified version of) the project model.[†]

Basic Design.

The Need for Progress Data. The Sim represents a project to build the first prototype of a launch vehicle (LV). The Sim's main window (see Figure 2) shows information on actual and planned expenditures (on the right side of the main window) as well as on design-adequacy (left side).

[†] The team that designed, created, and tested the Sim included people from NASA and Ventana Systems, Inc. NASA participants included Jason Derleth, Sharon Welch, Melissa McDowell, and Ave Kludze. Ventana participants included Jim Hines, Martha Miller, and Ron Suiter.

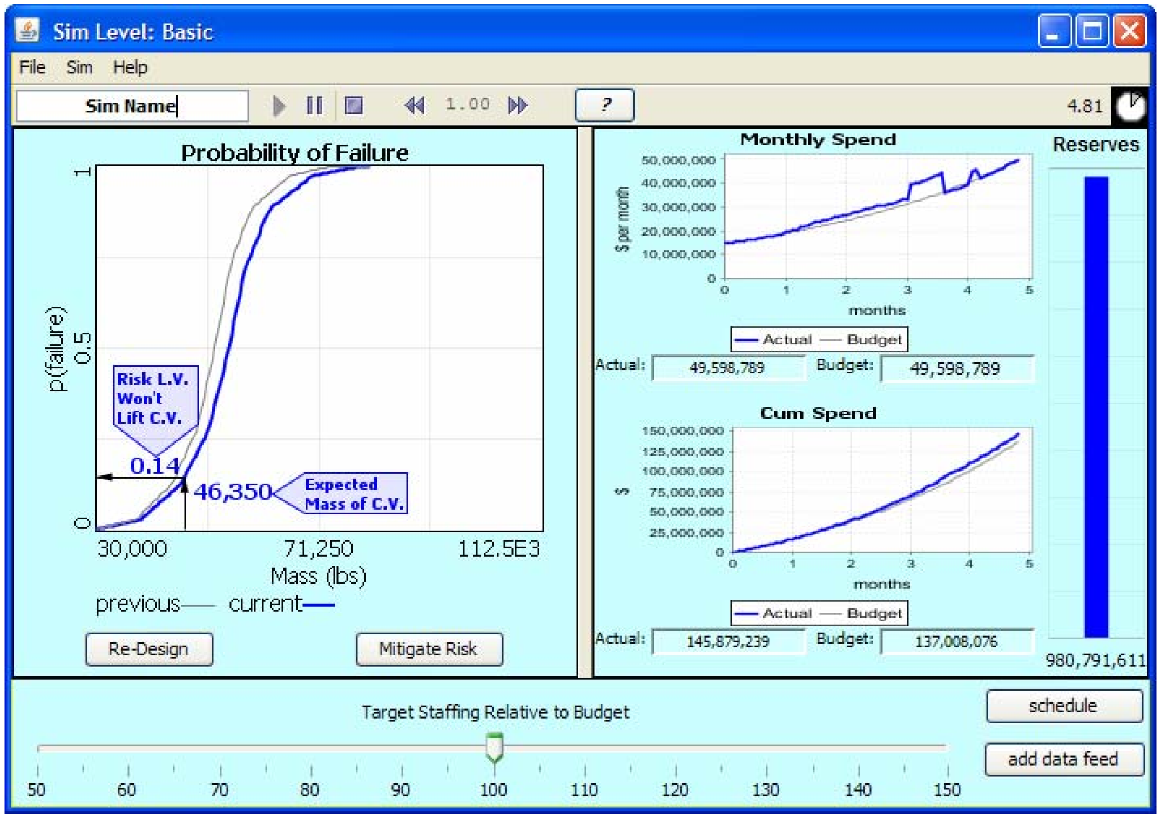


Figure 2: The Sim's main window shows information on design adequacy (left) and expenditures (right). The "add data feed" button in the lower right provides access to additional information.

The fundamental design "trick" is that *additional* data feeds are available, but are not provided on the Sim's main window. Instead, the user must actively seek out the data by clicking the "add data feed" button in the lower right and then choosing from a list of available feeds. The hope is that experiencing the need for data, and being able to successfully act on it, in the Sim will help the player to feel a need for action in the real world.

The rules of the Sim require the player to finish the project on time. The first problem a user must recognize and solve is that productivity and quality (i.e. the fraction of work done being done correctly) are less than explicitly planned. Consequently, if the originally planned staffing levels are maintained, the project will be unfinished at the originally scheduled completion date and the Sim player won't succeed.

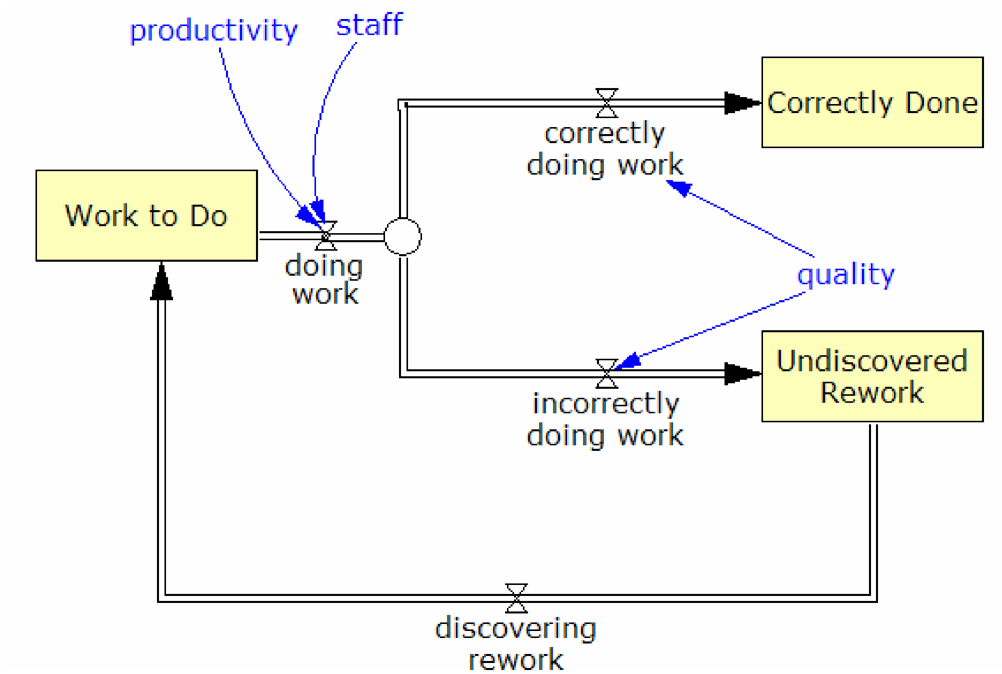


Figure 3: Productivity, Staff, and Quality. Blue "skinny" arrows represent causal connections (as opposed to physical flows, represented by double arrows). Mathematically,

$doingWork$. *And,*

$incorrectlyDoingWork_t = doingWork$. *If staffing is maintained as planned while productivity and quality are lower than planned, less work will get done and more of it will need to be redone than planned. At any point in time, less of the project will be complete than planned.*

In order to complete the simulated project within the allotted time, the player must use his reserves to boost staffing above the planned levels. Doing so isn't difficult: the Sim player simply must move the Staffing Lever (at the bottom of the main window) above 100% of plan. But, how does the player know how much to increase staffing, indeed how does he know that he must increase staffing at all?

"Expenditures", the information available on the Sim's main window, does not reveal the problem." Expenditures data relates to how many people are working, not how much they are accomplishing (see Figure 4). To recognize the need for, and to decide *how much,* additional staffing is necessary, the player needs a deeper view into the project than is provided by the main window's information on expenditure.

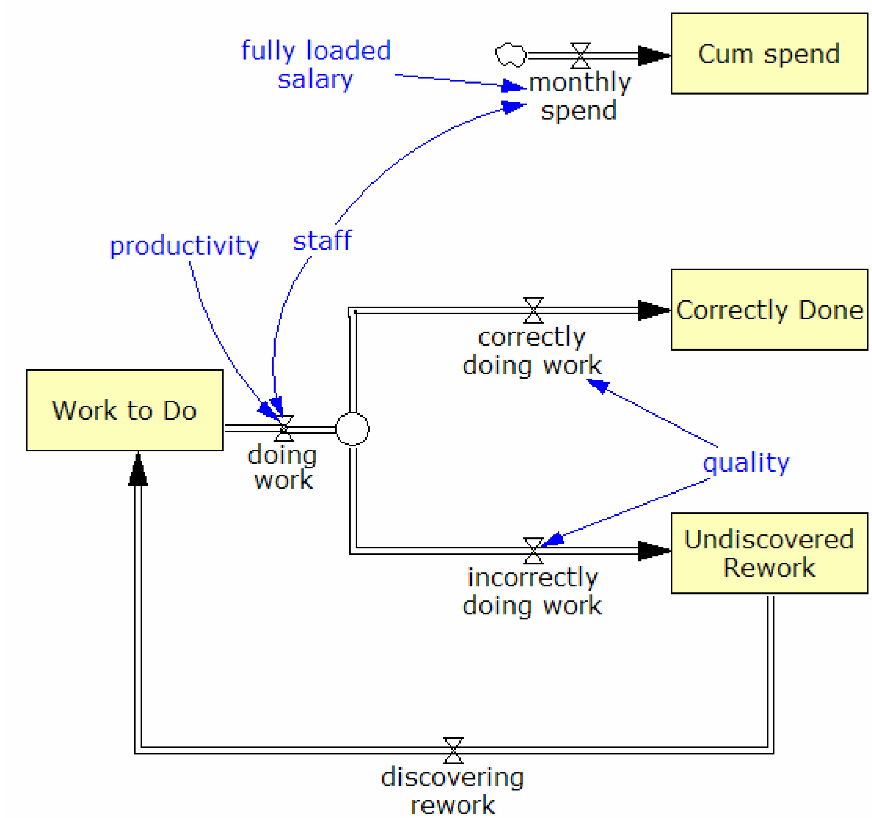


Figure 4: Expenditures. Expenditures (spend) is connected to (or caused by) staff, not by how much work has actually been done. Mathematically,

$monthlySpend_t = staff_t *$ *Staff levels can be deduced from data on spend. But, if productivity and quality are unknown, spend data does not provide enough information to deduce whether the project is on track or not.*

The Progress Data Feed[‡] provides the deeper view that the player needs. The data feed is not perfect. It represents what could be available to a project manager relatively easily. Because *Undiscovered Rework* is not directly measurable, the data feed does what most people do – simply counts the work that has been "done" and doesn't worry about the portion that will need to be redone. That is, the progress data feed shows *perceived cumulative work done,* which is the sum of *Correctly Done Work* and *Undiscovered Rework*.

[‡] The term "progress data" is abstract. The concrete reality of progress information varies depending on the setting. In one situation, progress data might be a time series on design drawing completions and revisions. In another situation, a series that tracks TBD's and TBR's might embody "progress".

Imperfect though it is, the progress data feed (and a bit of hand-eye coordination) allows the player to use the staffing lever to keep the project's cumulative (perceived) progress on track with planned progress[§] (see Figure 5).

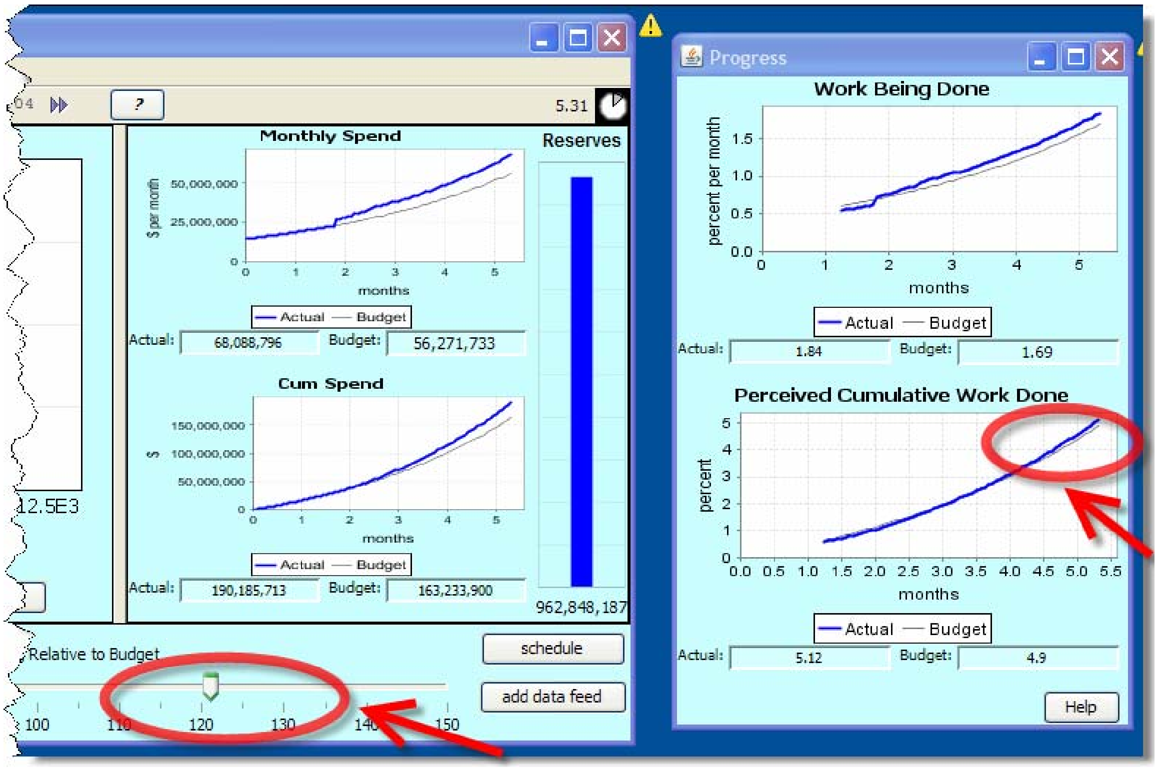


Figure 5: The player moves the slider, to make perceived cumulative work done track the plan (i.e. the "budget").

The progress data feed allows the player to create -- indeed to be part of -- a critical control loop in the simulated project: When perceived *cumulative work done* drops below plan, the player increases the staffing slider and that action brings *cumulative work done* back up.

---

§ In the real world, data on *perceived cumulative work done* is required for basic project management – except in very simple or highly routine projects. The real-world Constellation project lacked this data feed.

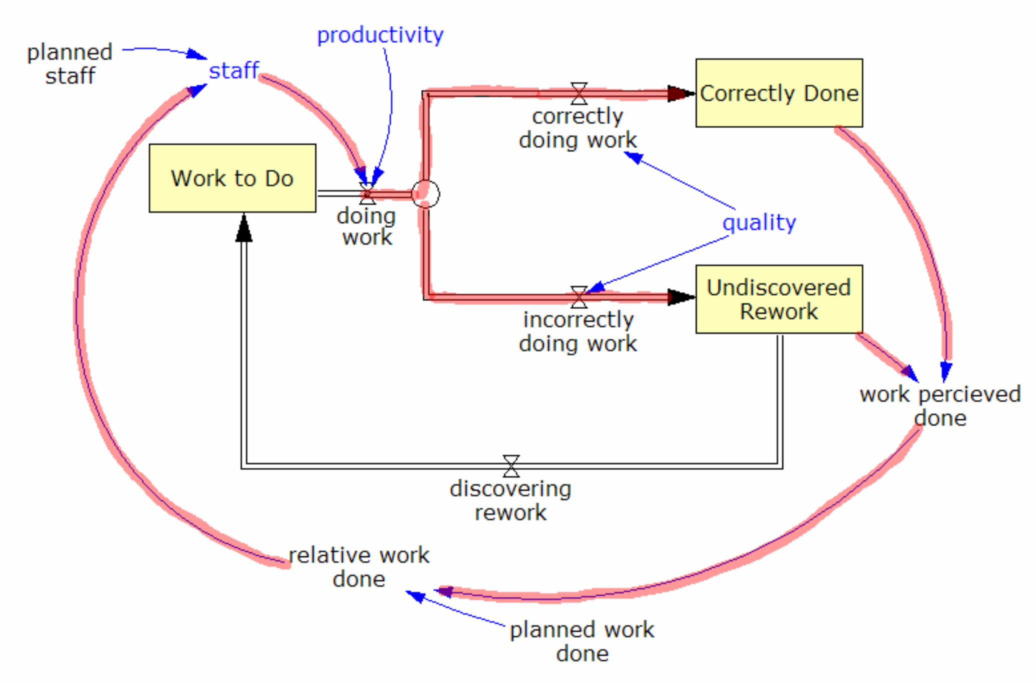


Figure 6: A critical control loop. The Progress Data Feed shows work perceived done and planned work done. The player mentally considers the amount of work done relative to planned and adjusts staff accordingly. Mathematically,

$$workPerceivedDone_t = CorrectlyDone_t + Undiscove.$$

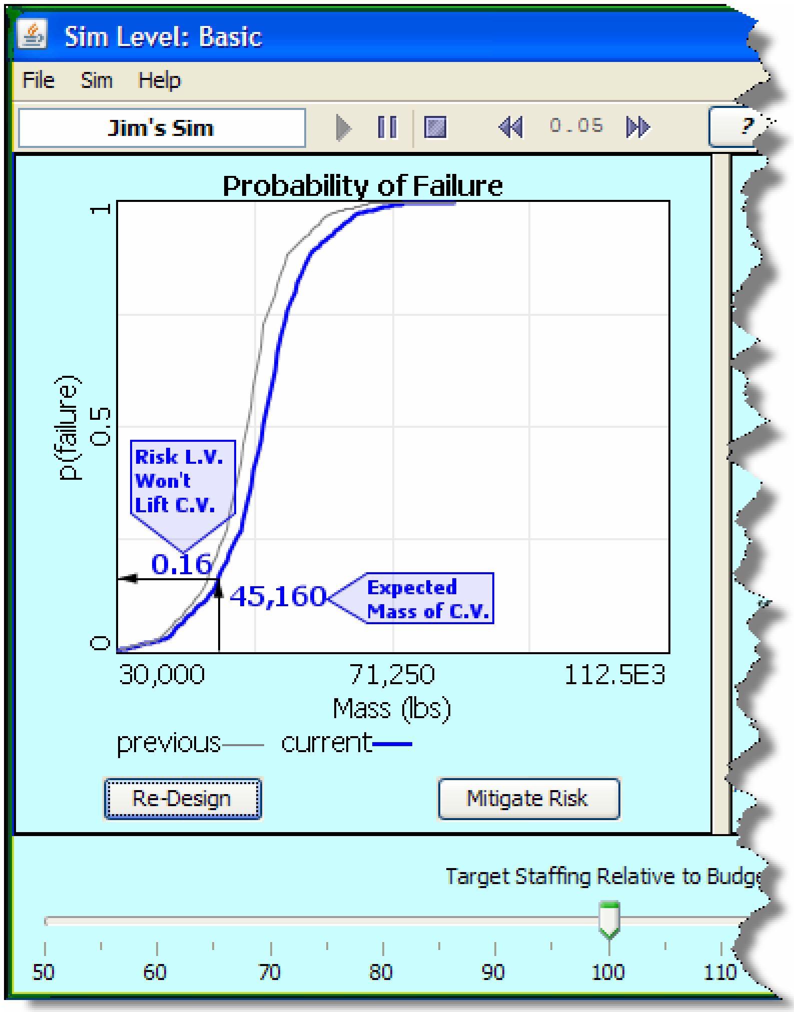
$relativeWo$. *One possible equation that might mimic the player's*

actions is .

The Need for Redesign. The game designers wanted the player to make a conscious effort to open a data feed. The game designers wanted to avoid a situation where players would open data feeds simply because they had nothing else to do. This led to three additional design decisions. First, opening a data feed costs "money", which is subtracted from reserves. Second, opening a data feed is a bit time consuming (the user sees three different windows, and must click four times in the process of opening a data feed). Third, and most importantly, the game player has a "distraction" clamoring for his attention.

The "distraction" in the Sim involves a realistic complication: In order to succeed, a Sim player must not only complete the LV project within a specified time, the completed LV must be able to lift the expected mass of the crew vehicle. The problem is that the expected mass of the crew vehicle continues to increase as the LV project continues. In response to these increases, the Sim player must initiate design changes to compensate.

Virtually the entire left side of the Sim's main window is devoted to the issue of lift and changes in the LV design. The intent was to make this aspect very obvious and, thus, very distracting.



The left side of the main window plots the probability that the LV's current design will *fail* to lift a given mass. The horizontal axis measures mass, the vertical axis measures *p(failure).* Small arrows, labeled with the currently expected CV mass and the current probability of failure, show the adequacy of the current design.

As simulated-time passes, the expected crew vehicle's expected mass increases and so the probability of failure rises. The Sim player sees the "up arrow" moving to the right and the "over arrow" rising. Whenever *p(failure)* rises above 20%, the background of the plot turns red, indicating an alarm condition.

To bring the probability of failure down, the Sim player must initiate design changes via the "Re-Design" or the "Mitigate Risk" buttons. "Risk Mitigation" represents such changes as substituting a lighter material for a heavier one and in general involves finding correctives to the most likely and easily correctable sources of failure in the current design. In contrast, "Re-Design" represents a fundamental change to the design.

Pressing either button is immediately reflected in a change to the shape of the probability-of-failure function. "Mitigation" shifts the bottom half of the curve lower, while "Redesign" causes the entire curve to shift rightward – either button will decrease *p(failure).*

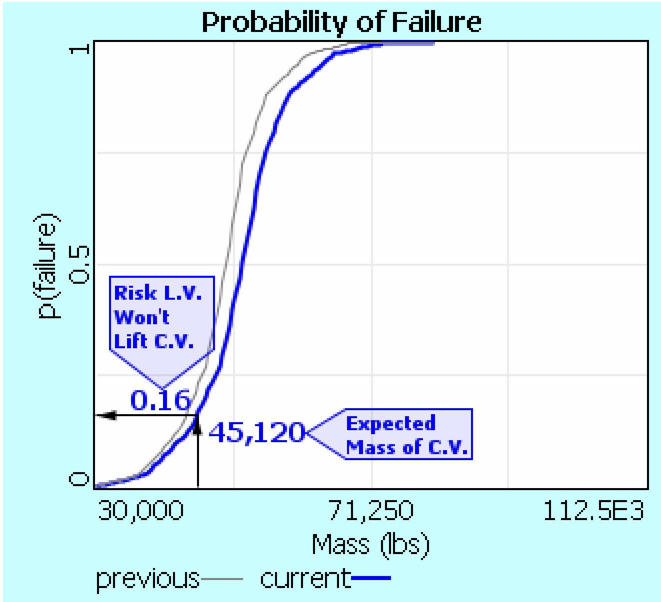


Figure 7: Redesign shifts the curve to the right.

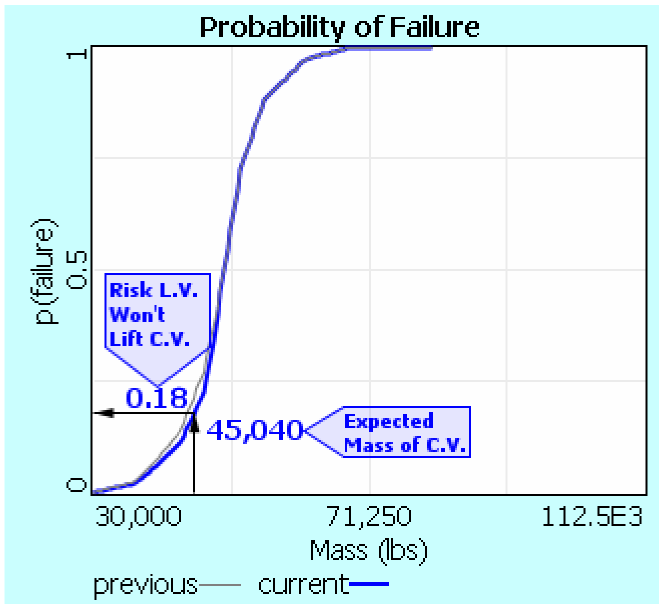


Figure 8: Risk Mitigation shifts the left part of the curve lower.

Pressing either button costs "money" and so shows up as a (small) reduction in reserves. Mitigation is less expensive but is also subject to diminishing returns – for example there is a limit to how light the material can ultimately be. "Re-Design" is a bit more expensive, but does not suffer from decreasing returns. Of course, because mitigation is specific to a design, clicking the "Re-Design" button obsoletes whatever mitigation has been performed previously.

Levels of Difficulty = Levels of Insight

The Sim contains three levels of difficulty, and insights build with successive levels.

Figure 9: The Sim contains three levels of difficulty. The player changes the level of difficulty via a menu item.

Basic Level. The fundamental insight of the entire Sim is contained in the first (or "basic") level: Progress data is critically important to managing a project successfully. In addition the first level contains two further insights.

The first additional insight of Level 1 is that managing a project involves managing both apparent and hidden backlogs of work (see Figure 1 above). A number of consequences follow. For example, most estimates of "work remaining" are directly based on the *apparent* stock of *Work to Do* and so tend to be biased low. Part of becoming a successful project manager involves gaining a sense of the size and future dynamics of the *hidden* backlog.

A further insight concerns how and when errors are discovered. In the real world, explicit testing of a component will often uncover errors. However, errors that make it through testing will likely remain hidden until the component is used with other components. This imparts a tendency for errors to be discovered later in a project, rather than earlier. And, so there's a tendency for more labor to be required late than seemed to be required early.

In the Sim, the bias toward more labor being required later is subtle, but creates a management challenge due to the final insight of the first level: Catching up late in the project can be difficult. The staff slider allows the player to increase staffing up to 150% of planned. Because planned labor in the Sim declines toward the end[**], the maximum absolute amount of additional labor declines as well. Players experience this as a difficulty in catching up late in the project.

Real world managers experience a similar difficulty – and for the same reason that real-world staffing curves also tend to decline toward the end of a project: The number of places where additional resources can be productively employed declines as a project is completed. First, as

---

[**] The Sim's staffing curve is "bell shaped": Low at the beginning as well as at the end.

components are integrated, there is less “room” to employ additional people, because physical space becomes cramped, as in an almost-completely-assembled launch vehicle. Second, late in a project work often becomes necessarily sequential as is the case when tracking down a bug in a piece of software. Hence, the difficulty Sim players have in adding staff late in the project corresponds tightly to a similar difficulty in the real world.

Intermediate Level: In the second (“intermediate”) level, design changes (i.e. Re-Design and Risk Mitigation) cause previously-completed work to become obsolete. Previously (i.e. in the first level) Redesign and “Risk Mitigation” only cost money – as if the only impact of a design change was the salaries of the designers. In the second level, the Sim player will find he must bring on additional people to accomplish design-caused work obsolescence.

The most important insight contained in the second level is that design changes that occur late are effectively more expensive, because late in the project there is so much more completed work that can be made obsolete.

The Sim player may realize a more subtle insight as well: Design changes late in a project are more expensive for a second reason. Toward the end of a project, undiscovered errors decline. Undiscovered errors need to be redone anyway, so obsoleting them is low-cost. But, late in a project, relatively little of the obsoleted work will be the low-cost undiscovered errors, simply because there aren't many. Consequently, the cost of obsolescence will rise.

Advanced Level. In the third (“advanced”) level, the Sim player can extend or contract the schedule (i.e. the staffing curve)[††]. Previously (i.e. in the first two levels) the player could view the planned staffing curve (via the schedule button), and could exceed the plan up to the amount in reserves, but the player could not change the plan. Now, the player’s control extends to include the plan itself.

[††] Note that extending the schedule automatically increases the budget by a commensurate amount. The dollar budget is proportional to the area beneath the staffing curve.

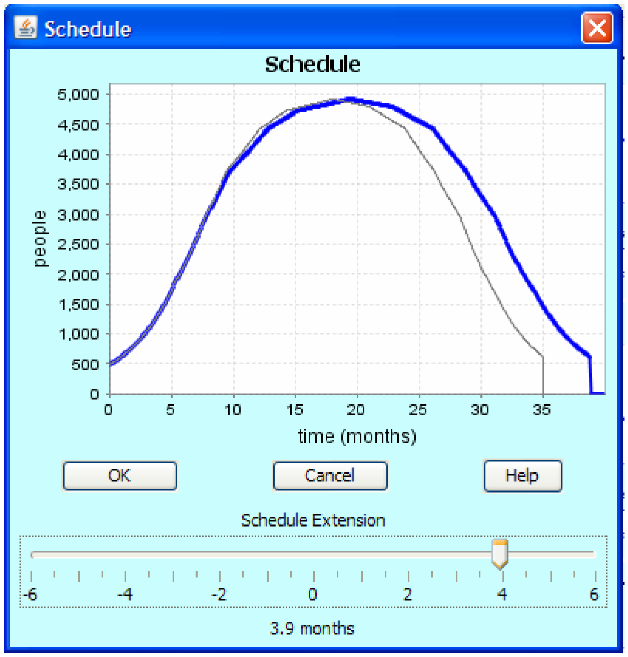


Figure 10: In the Advanced Level, the player can extend (or contract) the schedule.

Furthermore, "people inside the model" now **react** to the schedule. When the schedule/budget is tight, staff works overtime. Unfortunately, if maintained, overtime results in fatigue which reduces work speed and increases errors.

The advanced level's first insight is that overtime is a less powerful and more dangerous lever than it sometimes appears. Because of fatigue, people working overtime get less work done than they hope and create more future rework. In turn, this may create subsequent pressure for even more overtime, and hence more fatigue and even lower work speed and even more error generation. The situation can become dire enough that less is accomplished than would have been accomplished with NO overtime.

Because of the counter-impact of fatigue, overtime is always a less powerful lever than it appears on the surface, and in some cases overtime can actually threaten the viability of a project.

Some players may generalize this insight, beyond the specific example of overtime and fatigue in the Sim. In the real world "counter-impacts"[‡‡] afflict many of the other "fixes" that might be used to correct a project that is behind schedule. For example, if people decide to skip the "proof reading step" in order to get more work done, they will generate more undiscovered errors. If the "fix" is to increase the workforce, the project will have more hands, but skill will be

[‡‡] In system dynamics, this counter-impacts syndrome is called the "fixes that fail" archetype. The structure is composed of two fundamental feedback loops: An intended control loop and an unintended self reinforcing (or positive) feedback loop that undercuts the control loop and that in extreme cases can actually dominate the dynamics.

diluted, and so, again, average work speed will fall and errors will tend to rise. If the "fix" is to move ahead with work before pre-requisite (but slower) work is completed, information embodied in pre-requisite work will not be available to guide what should have been subsequent work, and the apparent increase in "production" will be followed by a surprising increase in error discovery down the line. Often, the solutions that seem the most drastic and least desirable – extending the schedule and increasing the budget – turn out to be the ones that actually have the fewest negative consequences for a project.

The ultimate insight in the Advanced Level is that schedule is not simply a plan or a prediction of what *will* happen. Schedule and budget are important management levers that determine the productivity of the workforce and the error generation rate.

To make this point, a Sim player will discover that simply choosing the correct schedule once at the beginning and maintaining it is quite sub-optimal. Extending the schedule early in the project to reflect the actual predicted completion date will create a sense of "plenty of time", because the (future) backlog of undiscovered rework is not visible to the Sim's "workforce". The "plenty of time" feeling in turn results in "under-time" (experienced in the real world as inefficiency or "gold plating"[§§]).

Consequently, early in the project the Sim player must maintain a schedule that seems sufficiently immediate to the staff. Later in the project, when rework begins to be discovered in earnest, the Sim player must maintain a schedule that does not cause excessive overtime and fatigue.

With the insight that schedule is a management lever comes a reinforcement of the Sim's fundamental goal, creating an impatience for data. In order to use the "schedule lever", the Sim player will need another data feed in addition to the one on Progress. Here, the Sim comes full circle in the final level, reinforcing the point that the world of project management is a world where data is important but not a world where any additional data is simply additionally useful. In the third level, as in the first, it's not the case that any additional feed will do. Specifically, the Sim player must ask for the Overtime Data Feed, because that's the one that's closely linked to -- and so reveals – the particular issue to which the level 3 manager must respond...

II. DATA ON SIM EFFECTIVENESS

Purpose:

Testing focused on the Sim's fundamental purpose to "increase the appetite for data". It was hoped that the Sim would increase users' perception of the usefulness of project progress data in guiding a project to successful completion and hence make them more willing and eager to gather it in the future.

[§§] "Gold plating" refers to adding on non-essential features to a project – perhaps a 3-D user interface for a piece of software when a 2-D interface would have been just as effective.

To test whether the Sim achieved this objective, we generated an online survey to be completed by each person who uses the Sim. The online survey includes before and after questions designed to gauge the impact of the Sim on its users.

It was thought that the impact of the Sim might be mediated by three demographic variables, however. Consequently, demographic information was gathered to test these hypotheses. The supposed mediating variables were:

a. Degree of comfort/familiarity of users with sim or computer gaming environments
b. Age affects, though perhaps mediated largely by the familiarity effects mentioned above
c. Highly experienced project managers might be differentially impacted by the Sim

The online questionnaire may remain a part of the Sim into the future, in which case testing could continue as the Sim gains more users. An initial analysis of the Sim's effectiveness was conducted as part of the current project.

Data for the analysis was gained through each participant's completion of the demographic and impact questions contained in the online questionnaire. In addition, we felt it important to conduct live interviews with each user. Live interviews allow plumbing behind multiple choice answers to gauge the clarity of the questions, more fully delineate participants' reactions, clarify areas of ambiguity, and ensure that the online questionnaire would solicit accurate information regarding the impact of the Sim in its future use. It was also hoped that live interviews would allow discovery of any improvements needed to the Sim itself.

Process:

An online questionnaire was created to be completed by each Sim user. The live interview guide included the same questions. Both the online and live versions have portions completed before and after participants work with the simulation. (See Appendix A.) Participants were sent the demographic questions in advance to facilitate responses to them during the live interview. The substantive questions were not sent in advance to avoid signaling to participants that the study focused on progress metrics in particular.

Participants were recruited through several channels within NASA. Some recruits were members of NASA's FIRST program for project managers in training. Others were senior people who already had considerable experience as project or program managers. Interviewees volunteered to participate in the research.

Ten participants elected to participate. They encompassed wide variability in project experience, age, seniority, familiarity with computer games and simulations, and the NASA centers in which they are employed.

Five interviewees gave their current primary job function as engineering. Their specialties included Computer Systems, Mechanical, Aerospace, Design and Mechanical engineering. One was a Scientist, trained in Physical Chemistry. Two were analysts—focusing on both Program and Strategic analysis. The final two were Program Managers.

The age distribution was wide, spanning from 20-year-olds to 50-year-olds. Four participants were in their late 20's, one each were in their early and late 30's, one was in the late 40's, two were in their early 50's and one was in the late 50's.

Center distribution tilted toward Langley. Five interviewees work at Langley, two each at Marshall and Goddard, and one at Johnson.

Experience with computer games and simulations varied broadly across the sample as measured both by the number of games/simulations individuals had tried, how frequently they used them in the past, and how long ago their peak use occurred. Thus the sample had a wide range of familiarity with computer simulations represented within it.

Finally, self-ascribed experience with running projects also diverged dramatically across the sample. Some said they had no or very limited experience running projects; others said they were highly experienced. Their grade level and supervisory status echoed this range.

Findings:

The interview responses allowed us to accomplish each of our original goals. We were able to gauge the reported impact of the Sim, explore moderating demographic variables and garner suggestions for improving the Sim. Each of these outcomes is described below.

Impact of the Sim.

Interviewees were evenly divided on whether they felt they had learned something from the Sim and said the learning would affect them the next time they ran a project. Five interviewees indicated they learned a lot and that the learning would definitely influence them in the future; five indicated they did not learn and would not do anything differently in the future. While this outcome might make it seem the Sim failed in its objective of increasing learning and an appetite for data since it created only a 50-50 effect, exploration of the demographic data reveals a more interesting pattern.

Exploration of demographic variables.

As stated earlier, we hypothesized a priori that several demographic variables might mediate the Sim's effectiveness. Below, we examine each variable in turn.

a. Degree of comfort/familiarity of users with sim or computer gaming environments

Interviewees were asked both about the number of computer games/sims they had used in the past five years and the frequency with which they used them. Both factors were NOT correlated with the amount of learning reported from this Sim. The number of games played was slightly negatively correlated with learning (-0.1248); the frequency of use was also slightly negatively correlated (-0.06667). Neither correlation explained a significant amount of variance in how much was learned.

Consequently, we can conclude that it is not true that only serious gamers or sim users will be able to learn from this Sim. Some individuals who had a great deal of gaming experience said they did not learn anything from the Sim; some who had no gaming or sim experience did. Apparently, then, game/sim experience was not a major mediator of the Sim's impact.

b. Age affects, though perhaps mediated largely by the familiarity effects mentioned above

The second demographic variable, age, had a higher effect on reported learning. In general, younger people said they learned more from the Sim than did older people (0.39). The second part of the hypothesis, however, did not hold since, as noted above, this effect was independent of gaming experience. Some older people were experienced gamers and some younger ones were not. Age had a greater effect than gaming expertise. Consequently, age does seem to be a factor, but not one mediated by gaming familiarity. Age is, however, confounded with the amount of project experience in this sample. Consequently, we have to test the third hypothesis to gauge the relative contributions of age and experience.

c. Highly experienced project managers might be differentially impacted by the Sim

Experience levels turn out to be highly predictive of interviewees' reported learning and gain of an increased "appetite for data". The correlation between experience and learning is 0.543915 which is considerably higher than for any other variable. What is causing this significant effect?

Fortunately, the interviews provide the answer. Specifically, highly experienced project managers knew of the importance of progress data before they ran the Sim. They reported its importance as high in their responses to the interview questions prior to running the Sim. They were, consequently, not surprised that it turned out to be important in successfully completing the Sim.

For inexperienced interviewees, however, project progress data was not cited as being particularly crucial to the success of the project prior to the Sim. Communication and personality factors within the team were more often cited as the key drivers of success. Following the Sim, however, participants reported being struck by how important progress metrics were to understanding how the project was doing. Interviewees said the Sim made them much more likely to actively seek data of this kind in the future if they were running a project.

In sum, individuals who already have a great deal of project experience have already gained an appreciation of the importance of project progress metrics. Consequently, the Sim did not create this knowledge in them. As many of them said, "The Sim reinforced what I already knew". They did not, therefore, report an increased appetite for data since they said "they already had a large appetite for it".

For more junior people, however, the Sim generated insight into project dynamics which were not salient at the outset. They reported that they would indeed be eager to seek progress metrics to allow them to manage those dynamics.

It is important to note that the Sim might have untapped value for both inexperienced and experienced project managers, however. Most participants remained at the beginning level of the Sim. Only a few reached the intermediate level and no one advanced to the third and highest level of the Sim. New dynamics and potential learnings emerge at each level. Senior people in particular were limited on the amount of time they could devote to the Sim given the many other pressing demands on their time. If the Sim is used in classrooms or other settings in which more time could be devoted to it, it is distinctly possible that both inexperienced and highly experienced project managers would report new learning and a greater desire for data.

Other learnings:

The interviews also garnered suggestions for useful modifications to the Sim. We are most grateful to all the participants for their thoughtful comments. Chief among them was a request for a "help" feature which would allow Sim users to get hints and tips if they were having difficulty progressing.

Interviewees also said they would like to have a summary of key learnings at the end of each level. They wanted to make sure they had indeed mastered the dynamics inherent in the Sim and understood how the forces were interacting to affect project success.

They also suggested relabeling elements in some charts to increase clarity and improving access to and salience of the initial screens which orient the user to the situation and objectives being portrayed in the Sim.

Each of these suggestions was adopted and incorporated into the final version of the Sim created during this project. That version of the Sim is now up, running and ready for deployment.

In sum, the Project Management Sim seems to engage the attention of those who use it (whether they had experience with prior sims and computer games or not) and to convey important and useful information to developing project managers regarding the role and importance of project status data—information which has clearly been gained by more experienced leaders and deemed important by them. Presumably, possessing this knowledge was either an important criteria for promoting individuals into project management or it was information gained by them in those roles. It might be useful to convey this information at the front end rather than having it be learned through trial and error. It seems that the Sim is helpful in carrying that information and hence might prove valuable in project management education programs.