Optimal Resource Allocation for Projects

Collin Carbno, 40 Dunning Crescent Regina, Saskatchewan S4S 3W1 Canada

■ Abstract

Out of the opportunities available management must decide which ones will be pursued and how many resources will be assigned to each. Usually, these opportunities (projects) are evaluated for economic feasibility, and then the favorable ones pursued. Several questions arise regarding the allocation of available resources to the approved projects: In which order should the projects be done, or should they be done concurrently? How many resources should be assigned to each project? Given the multitude of factors affecting such decisions, one would expect complex formulas to describe the optimal allocation of resources. Contrary to expectations, the formulas describing optimal allocation are intelligible and practical in use.

Keywords: optimal resourcing; Manfred's distribution; optimal staffing levels; project desirability

©1999 by the Project Management Institute -- 1999, Vol. 30, No. 2. 22 31 8756 9728/99/$5.00 per article + $0.50 per page

Project management is a central activity of information technology (IT) groups. The allocation of staff resources by management of an IT group to the various software projects under development is a complex undertaking. Management must weigh numerous factors, such as urgency of the project, required skills, competence and availability of staff, and the impact of resource levels on delivery schedules, in determining the allocation of resources. But are these allocations optimal? Could mathematical models provide insight and assistance into the allocation and scheduling of resources to projects? This paper describes such mathematical models.

The resourcing level assigned to projects sometimes happens more by default than by explicit decision. A particularly valuable and important project may "arrive" in a responsible, but small, group and be resourced at a relatively low level, while other resources in nearby groups that could, and probably should, be assigned to this project are defaulted to less important projects. Furthermore, resources that come from larger pools are often assigned to projects on the basis of "project size." While allocation schemes based on size are most often appropriate, sometimes they are not. The economic implications of inappropriate project resourcing can be significant. A project with $10K per month delay cost that takes 10 months instead of five months suffers an additional delay cost of $50K. In this case, if the development costs remained constant regardless of the development time frame, then inappropriate resourcing would add about 100% to true overall project costs.

Taking into account changing development costs with development time frame usually lowers the overall impact, but that impact will remain significant. A simple rule of thumb is that if resourcing is out $x\%$ and $x < 20\%$, then costs will roughly increase by $x\%$ also. Thus, guidelines for assisting management and project managers in setting appropriate resourcing level for projects by taking into account the various factors such as risk, project effort expected, timing issues, and availability and cost of resources have direct implications for bottom-line performance.

Optimal Resourcing for a Single Project

Consider a single information technology project. This project can be evaluated by the usual discounted cash flow method to yield a net present value (NPV) for the project. During the economic evaluation process of the project, an estimate of the development effort E (usually in person days) is done. Implicitly, this estimate also involves giving context for the estimate; namely, the development time frame and the resources assigned. For the moment, let us restrict consideration to projects for which the effort lies within one department.

Projects provide benefits. Delaying project benefits therefore constitutes an opportunity cost (i.e., delay cost). Thus, decreasing the development time lowers the delay costs. But, decreasing the development time at some point requires significantly more staff to be assigned to the project. As more staff are placed on a project, more effort is consumed in communication, learning, and coordination. Thus, while the delay costs of the project are dropping the development costs are rising. In the context of the standard microeconomic approach involving total cost, indirect cost, and direct cost, there must be a development time frame for which the combined sum of total development costs and total delay costs of the project will be a minimum. Projects that are developed quickly will be development cost driven, and those developed slowly will be delay cost driven.

The delay cost curve (total delay cost versus project time frame) can be estimated from business considerations. The development cost curve (cost of developing the project in a given time frame) can be based on project development details. A graphical analysis on the sum of delay cost curve and development cost curve could then be used to estimate the optimal development time frame.

The graphical analysis approach requires considerable effort and a detailed knowledge of business delay costs and development scenarios. An alternative approach is to approximate the delay cost curves and development cost curves with mathematical expressions that reflect usual behavior of projects. Algebraic, numerical, and standard methods from elementary calculus can then be used to determine the optimal development scenario. The advantage of this approach is that parameters of the curves are estimated easily from known project information. Furthermore, the parameters can be changed easily in order to perform a sensitivity analysis.

Development Cost Curve Modeling. The development estimate is not a fixed number independent of proposed development time scale. Mathematically, the delivery effort E of the proposed project can be written as a function of a number of variables (S = staff on project, T = development time, complexity of project, skill of staff, newness of technology ...).

Holding constant all the factors other than time frame and staff assigned to the project, it is expected that there will be a development time frame called the limit of compression T_C at which development effort will approach ∞. As the development time frame is increased, the total effort should drop from ∞ until the limit of stretch T_S is reached, at which development effort begins to rise again. (See Figure 1.)

It has been suggested that if all other production factors are kept constant, then for a particular project S and T will satisfy a formula of the form

$$S \times T^{\alpha} = K \qquad (1)$$

where K is a constant for the project in question. Provided that T is sufficiently inside the range between the T_C and T_S, this simple formula appears to give reasonable results. The constant α is usually chosen in the interval [1, 2]. A value of $\alpha = 1$ would indicate that we can compress the project delivery time indefinitely without a "cost penalty," a situation contrary to experience. Based on personal preference and experience, a value of $\alpha = 4/3$ will be used for the remainder of this paper. Given an estimate that the project can be delivered by S_1 staff over a time T_1 one can use equation (1), a type of Rayleigh's law, to calculate the project constant K. Then equation (1) can be used to calculate the required staff S_2 for a time T_2, respectively or vice versa (Boehm, 1981; Pressman, 1992).

This formula assumes that there is available staff of the same quality or "worth" to the project. Naturally, if the additional staff assigned to the project are less experienced or skilled, the additional effort added to the project may cause the project duration to grow instead of shrink. This reminds one of the observations that adding staff to a late project will only make it even less on time (Conte, Dunsmjore, & Shen, 1986).

The total estimated effort, E, for delivering the project is given by E = S × T from which equation (1) yields,

$$E = \frac{K}{T^{1/3}} .$$

Limit of Compression. The limit of compression can be estimated by using the rule of thumb known as "Square Root Person Month Formula." It basically says that limit of compression for a project is typically given by

$$T_c = \sqrt{E_s}$$

where E_S is the effort to deliver the project at T_C expressed in units of person months.

Limit of Stretch. The limit of stretch T_C can be approximated by the point at which staff assigned to the project is one person. When one person has to juggle several project efforts his or her overall productivity usually drops and the total required development effort begins to rise again. In the proposed model, the amount of increase in development effort is modeled by $1/(1 - S)$. Thus, the limit of stretch is not a hard limit and many projects are run at staffing levels below one person (Conte et al., 1986).

Generalized Rayleigh's Law. For most projects, equation (1) can be used to model a project's labor development cost curve, provided one stays away from the limits. If limits become important, then a more complex form can be used such as:

■ About the Author

Collin Carbno graduated with a double major in mathematics and physics from Brandon University, and has an M.S. in theoretical physics. He has 19 years experience in the IT industry, including experience in real-time assemble-level programming, VAX system management, C, and C++ development. For the last nine years, he has been an internal business consultant doing project consulting for various business reengineering initiatives, process initiatives, and providing assistance in ongoing prioritizing and evaluation of potential IT projects.

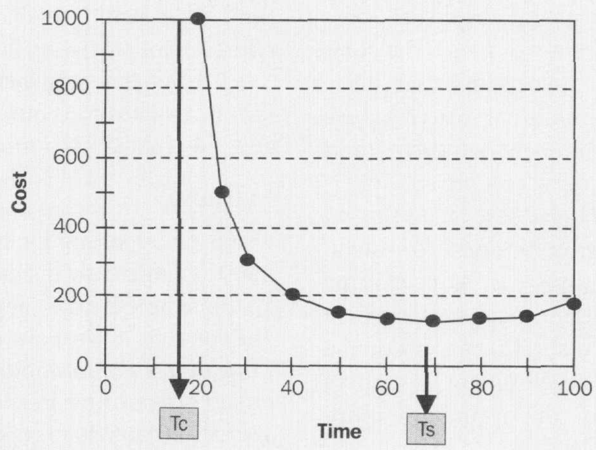

Figure 1. Typical Development Cost Curve

$$S = K/(T-T_C)^{4/3} \text{ for } S > 1, T > T_C$$
$$S = K/(T-T_C)^{4/3} \times 1/(1-S) \text{ for } S < 1.$$

My personal experience suggests that use of such complex forms is rarely justified. Projects in the early stages have, at best, rough estimates, and it is better to plan project development away from the limits.

Delay Cost Curve Modeling. It has been found that delay costs of projects can usually be modeled adequately by one of the following three mathematical curves (see Jardine, 1973).

- A cost curve in which the delay cost rises in linear fashion (linear delay curve, $C = D \times T$).
- A cost curve in which zero delay cost is encountered until a given target date (time), at which one suffers an immediate cost and then growing cost thereafter (hard date curve).
- A cost curve in which the delay cost rises in quadratic fashion. (The quadratic delay curve, $C = G \times T^2$, occurs when the project is affecting a customer base that is growing or shrinking so that each additional lost or gained customer adds to the impact.)

Estimation of Linear Delay Constant D. Given a linear delay cost curve, how should one estimate the linear delay constant D? To calculate the delay cost curve constant, one could calculate different NPVs for the different scenarios of project development. This approach has been found to be computationally messy. The difficulty is that not only does it require more than merely time shifting of the cash flows but also a detailed understanding of how various delays impact the various cash flows. A simpler and more productive method is to assume that over a period L the benefit B of the project vanishes. The linear delay cost coefficient D can then be approximated by $D = B/L$. A simple choice in many cases is to let B be NPV and to let L be the period used in the economic study. Intuitively, this assignment reflects the notion that benefit B would be reduced to zero in the period L.

For example, if a five-year NPV study indicates a NPV of $1,000K (B = NPV), then one can understand the project as delivering a cash flow of about $200K per year (neglecting time flow value of money). Including the time value of money, however, the annualized contribution of project at an 8% discount rate is about $250K per year. Not implementing the project then, in effect, results in an opportunity loss at rate of $250K per year. While using annualized values is a valid approach to delay cost, often the economic numbers at an early project date are uncertain, and the above approximation is a simple and conservative estimate of delay cost.

If a project must be done (sometime), and if we assume that a delay of period L results not only in loss of any NPV but also incurs the loss of NPC (net present cost) of development, then B = NPV + NPC more accurately reflects the true benefit. Again, many economic studies give present worth of annualized costs (PWAC) indicators, and these can be used for more precise estimates of a linear delay cost factor.

Optimal Staffing for Linear Delay Project. Given a linear delay curve, the total cost of project can now be written into the formula

$$C(T) = P\frac{K}{T^{1/3}} + D \times T$$

where P is the cost of a development effort per staff on the project. For example, a typical programmer has a loaded labor rate of about $100K per year, so P would have a value of $100K per year. Using elementary calculus, one finds that

$$C'(T) = -\frac{P}{3}\frac{K}{T^{4/3}} + D\,.$$

Equating $C'(T) = 0$ and noting $S = K/T^{4/3}$, the optimal staffing point S_0 is given by

$$S_O=3\left(\frac{D}{P}\right).$$

It is interesting to note that optimal staffing on the project depends only on the delay cost of the project and not on the size of the project effort. This result reflects the fact that the mathematical model selected here assumes that large and small projects are really the same except for a matter of scale in behavior. In the case of quadratic delay cost curve, a similar calculation yields that

$$S_O=\left(\frac{6G}{P}\right)^{4/3}\left(K^{3/7}\right)$$

and the optimal project resourcing does depend on the project size.

Capital Cost Considerations. Various capital (office space, computers, software, disk space, etc.) is involved in projects. The simplest way to handle these resources is to imagine that they are rented. In particular, a staff on a project will need phones, workstations, offices, and will have other costs that generally are directly proportional to the number of staff (S) on the project. Let R_S be the rental cost of such services per staff person. Also, there are usually other costs that are independent of the number of staff assigned to the project, such as a development server and so on. These costs can be reflected in a rental cost R_P.

The staff costs can be absorbed into a "loaded labor rate" $P_{new} = P + R_S$, and the capital overhead can be absorbed in the delay cost with $D_{new} = D + R_S$. Thus, in calculating optimal staff, the loaded labor rate should be used for the staff, and the delay cost should also include the direct project costs incurred in running the project.

Optimization in Multiple Project Situations

Typically, an IT department needs to optimize its resources over numerous projects. Usually, the specialized IT resources are insufficient to staff all the projects to the optimal levels. Several questions arise. First, how do we optimally allocate the scarce IT resources across the projects, given each project's delay costs (reflected by each project's D) and resource requirements (reflected by each project's K)? Second, when should projects wait? And third, over the long term how many staff members should the entire department have?

Let's tackle these questions in order. Assume that the projects must be run simultaneously. To determine how scarce IT resources should be allocated, one could develop a total cost equation including the resource constraint. Then, following the usual calculus procedure of setting all the partial derivatives with respect to project times Ti (development time for i-th project) to zero and solving the set of algebraic equations, the optimal times could be found. Given 50 projects, however, one will obtain a set of 50 nonlinear simultaneous algebraic equations in 50 variables Ti. Furthermore, because the projects will have different completion times, there are many ways of reallocating the resources; hence, the number of scenarios that would need to be checked grows exponentially with the number of projects. Obviously, a different approach is required.

Suppose, instead, that one develops an expression that gives the relative resource allocation that one should give to any two projects. By computing the ratios to a given key project, one could then rescale the result to determine the optimal resources to assign to each project. This rescaling approach is outlined below. Naturally, in a given IT department, projects are being completed and new ones added continuously. In this approximation approach, the optimal staffing requirements are recalculated as projects are added or removed from the list, and appropriate adjustments made to the project resourcing.

Ratio of Project Staffing in the Case of Plentiful Resources. Clearly, in the case of plentiful resources, the optimal ratio will be the ratio of optimal staffing for each project. For the linear delay projects, this yields

$$\frac{S_1}{S_2}=\frac{D_1}{D_2}.$$

For two quadratic delayed projects, one finds

$$\frac{S_1}{S_2}=\left(\frac{G_1}{G_2}\right)^{4/3}\left(\frac{K_1}{K_2}\right)^{3/7}.$$

Ratio of Project Staffing in the Case of Scarce Resources. In the case when the resources are scarce relative to the optimal resourcing both projects will be delay cost dominated and one can approximate the cost function by using just delay cost terms for the two projects (where each project is a linear delay project), namely,

$$C(T)\approx D_1T_1+D_2T_2$$

subject to the constraint that $S_1 + S_2 = Q$ and where Q is the total available staff in the department. Using the method of Lagrange multipliers in variables S_1, S_2, λ gives

$$C(S_1,S_2,\lambda)=D_1K_1^{3/4}S_1^{-3/4}+D_2K_2^{3/4}S_2^{-3/4}+\lambda(S_1+S_2-Q)$$

and setting all the partial derivatives to zero one obtains that

$$\frac{S_1}{S_2}=\left(\frac{D_1}{D_2}\right)^{4/7}\times\left(\frac{K_1}{K_2}\right)^{3/7}=V_{12}.$$

This approximate result suggests that in the case of scarce resources, larger projects should be resourced more than smaller projects. Computation of linear to quadratic, quadratic to quadratic, and so on, delay cost curve type is messy. In practice, it is simpler to approximate all curves with some linear delay value that will yield a similar delay cost over the expected time frame of the project.

Project	Delay Cost	D_1/D_n	S_o	SS
1	$100K/yr	1	3	12*1/6=2
2	$200K/yr	2	6	12*2/6=4
3	$300K/yr	3	9	12*3/6=6
Total		6	18	12

Table 1. The Staffing of Three Projects

Resourcing at a Staffing Level X. In most cases, the department will have neither plentiful nor scarce resources. In this case, an approximation to the optimal resource ratio can be found by simple linear approximation between the two extreme cases of plentiful and scarce resources. Given a set of projects for a given IT department, one can easily calculate the optimal staffing for the department as the sum of the optimal staffing on each project. Let Z represent this optimal staffing value. Given that the department only has resources Q, the staffing level is defined by X = Q / Z.

The ratio for S_1 / S_2 for a department with resources between scarce or plentiful can be given by the linear approximation between scarce and abundant resources ratios, namely by

$$\frac{S_1}{S_2} = \left(\frac{D_1}{D_2} - V_{12}\right)X + V_{12}\,. \qquad (2)$$

Example Project Level Resourcing Calculation. Suppose a department has almost enough resources to optimally staff all the projects. To determine the staffing level for a given project, calculate S_1 / S_2 (which in this case is given by D_1 / D_2 ratio) ratio for all the projects relative to some key project. Then, divide each ratio by the total of all the ratios, and multiple by Q to determine the optimal staffing for any given project. For the sake of illustration, imagine a department with three projects and a staff Q of 12. The optimal S_o is given for each project by 3 × D/P where staff cost P = $100K/yr. Table 1 indicates that to optimally staff all the projects would require a staff of 18. The suggested staff SS, for each project, is given by SS = Q × (D_1 / D_n) / R, where R is total of ratios, which in this case is 6.

If the department had only four resources, then in place of the D_1 / D_2 ratio, one could use S_1 / S_2 ratio of equation (2). Whether the added complexity of calculating using the scarce resource ratios is worthwhile, given the roughness of data at these early project stages, is debatable. In practice, it is probably sufficient to use just the optimal resource ratios. Using scarce resource ratios becomes particularly messy when one has a mixture of linear, quadratic, and hard date delay curves.

Simultaneous or Serial Development. When a department has adequate resources, all projects are staffed to the optimal level and no projects should wait. However, if resources are significantly below the optimal values, the development time of some projects may exceed the limit of stretch T_S. In this case, it may be beneficial for management to delay some of the projects and concentrate on the others.

One approach to the question of whether a new project should wait is simply to calculate the limits of stretch for the new project. A new project should not be added when the resourcing on that project is below the limit of stretch for that project. Another approach is to compare the project's delay cost against the average delay cost of all the other projects in the department. Adding a new project then involves a tradeoff between delay induced in all the other projects and the delay cost incurred by delaying the new project by the amount of time expected to wait until sufficient resources become available. Unfortunately, pursuing this reasoning with precise mathematics results in messy formulas. A crude rule of thumb, reflecting somewhat the above reasoning, is that the project should wait whenever

$$\frac{D_n}{D_a} < \frac{T_n}{T_a}(1-X)$$

and where the small subscript "n" denotes a new project's value and subscript "a" the average value of the current projects in development. The factor (1 – X) follows from the observation that when staffing level X is 1, no project should wait. The practical meaning of the above expression can be seen by the following reasoning. If a project has the same duration as the average active project but has a higher delay cost then it makes sense to have another similar project wait while this one is done in its place. Similarly, if a project has a shorter development period but a similar delay cost why not finish it sooner and start reaping the benefits from minimizing the delay cost. If both factors hold, the delay cost is higher than the average project delay cost, and the project duration is smaller than average project, then obviously it is beneficial to start the project.

Sometimes management is forced to choose to work either on project 1, and then on project 2, or vice versa. Which project should be done first? Naturally, priority should be reflective of delay costs; a huge delay cost makes it less likely that the project should be delayed. But suppose we have a project with larger delay costs but also a longer time frame than the other project. If we put the lower delay project second, it has to endure the long wait for project 1 to finish. There will come a point with increasing length of project 1 at which, although the second project has less delay cost per unit of time, it is better to do it first and then do the second project. Using the cost function one finds that project 1 should be done before project 2 whenever

$$\frac{D_1}{K_1^{3/4}} > \frac{D_2}{K_2^{3/4}}.$$

Thus, a set of projects can be ordered by factor $D/K^{3/4}$, with highest values having the highest priority. This wait priority factor reflects the idea that the smaller the project and higher its delay cost, then the more likely the project should go first. This result is consistent with known results in dynamic scheduling, such as the generalized C_u rule.

Information technology industry departments typically have staffs with highly specialized and unique skills. This situation creates the potential where these resources can easily become bottlenecks between projects. When a unique resource is needed simultaneously on several projects, which ones should wait? Or should the resource be spread across the various projects and, if so, in what ratio or portion of time should it be spread? The analysis given above can be applied with small modifications directly to these situations.

Project Resourcing and Risk

Project Failure. Consider a total cost curve for one project, but this time include a risk factor in the cost. If B is the benefit expected, then B(q) represents the additional expected cost to the project due to expectation of failure $C(T) = S(T) + D(T) + B(q)$. If the failure rate of the project q does not depend on the time of development T, the optimal staffing point remains as it would without considering risk. However, when the probability of failure depends upon the development time T it does impact the optimal resource level. Consider the simple case where the probability of failure depends linearly on the delivery time through a risk coefficient r (for example, $q = rT$), so that $C(T) = S(T) + D(T) + B(r)T$. Following the usual arguments yields that optimal staff is given by

$$S_0 = \frac{3D+Br}{P}.$$

The risk of failure can be absorbed into the delay constant D by adding an amount $Br/3$ to the delay costs. Again, benefit B in this equation should be thought of as a quantity that should reflect the expected impact, usually in case of project failure given as NPV + NPC. For example, a company may have a large but necessary project with small NPV. The delay cost constant D calculated on this NPV value would indicate a small resourcing effort. However, if the project has an increasing risk of failure with delay and is clearly needed, the resourcing may need to be substantially larger.

This behavior of increased risk with the passing of time is actually fairly common. Indeed, any project with a fixed completion date, such as the "year 2000 compliant" conversions that many companies are struggling with, can be thought of as having an increasing danger of failing as the year approaches.

IT Demand. It often seems that the demand for IT resources is infinite. However, all projects are not created equal, as some projects have better return NPV/NPC ratios and some projects have a higher probability of failure. Defining the desirability index $d = NPV/NPC$, it can be hypothesized that the number of potential projects N(d) with desirability d follows a formula of the form

$$N(d) = \frac{N_0}{d^2}.$$

Clearly, this formula suggests infinite IT demand for all projects with positive NPV. Looking at project data at Saskatchewan Telecommunications and counting the number of projects with say, a desirability of 50 compared to those of a desirability of 10, it has been found that the number of projects grows faster than $1/d$ but not as fast as $1/d^3$. Now consider many equivalent projects, with probability of success p and failure q. The expected gain per project is given by the expression

$$NPC \times d \times p - NPC \times q.$$

A success rate of p indicates that p fraction of the benefits will be realized on average. Clearly, an IT department should favor projects for which $d > q / p$. The effective desirability of a project including the expected failure rate is $d_e = d \times p - q$. Given that projects have a finite probability of failure, it follows that there is a cutoff desirability d below which projects should not be pursued. Thus, the demand for IT resources should not be considered infinite. Indeed, an IT shop that maintains a careful history of its projects can more effectively estimate the total demand it faces at any particular desirability level.

Resourcing of Project Steps

Projects often consist of a series of different steps such as analysis, design, development, and testing. Furthermore, different portions of the project are often done in parallel in different departments.

Serial Steps. The question arises as to the proper resourcing for the individual steps of the project. Should these steps be resourced the same as the whole project or differently? Under the reasonable assumption that each step can be considered to be a miniproject in itself, it follows for a two-step project that

$S_1 \times T_1^{4/3} = K_1$
and
$S_2 \times T_2^{4/3} = K_2$

where S_1, T_1, S_2, T_2 the staff and times for steps 1 and 2, respectively. Assuming a cost for staff of P_1 step 1, and P_2 step 2 and using the usual arguments, one finds that optimal staffing is given by $S_1 = 3 \times D/P_1$ and $S_2 = 3 \times D/P_2$.

The optimal staffing doesn't depend on step constants K_1, K_2 but does depend on the cost of the staff for the different steps. Note that, if a particular step requires resources at a higher cost it should be resourced at a lower rate. If the staff at every step has the same cost, then project resourcing should remain constant. Normally, a project is resourced with a bell-shaped curve, that is starting with a few resources and then peaking and gradually tapering off. Such a resource curve is probably driven by constraints of managing and supplying work to the project staff effectively. Thus, the optimal staffing levels given here should probably be considered more the average staffing levels of such resource curves.

Parallel Steps. The IT department's portion of the work is often only a small fraction of the entire project. One can think of the IT development work as being done in parallel to other efforts of the project. Therefore, we would expect that the two parallel steps would also be governed by equation (1)

$S_1 \times T_0^{4/3} = K_1$
and
$S_2 \times T_0^{4/3} = K_2$

where S_1 is staff on parallel step 1, S_2 is the staff on parallel step 2, and the total staff on both steps is $Q = S_1 + S_2$. The cost function for the two parallel steps is therefore, $C(T) = D(T) + S_1 (T)(P_1) + S_2(T)(P_2)$. Using the usual approach one finds that the optimal point comes at $3(D) = S_1 (P_1) + S_2(P_2)$. The total project length is $T_o = (K_1 + K_2) / Q^{3/4}$. Solving for Q, one finds that optimal staffing level is given by

$$Q_0 = \frac{3D(K_1+K_2)}{(K_1P_1+K_2P_2)} .$$

If $P_1 = P_2$, then $Q_0 = 3D/P$ as before. Thus, one again finds that the total optimal staffing applied at any time of the project does not depend on the way the project can be broken into parallel activities if the staffing costs of those parallel steps are the same.

Business Reengineering and Project Springs. Reengineering methodology typically involves determining the steps of a manufacturing or service process and the cost per step. One then attempts to determine changes in the process or a redesign of the process to lower the overall cost (or time) of the process while increasing or improving output. One approach to identify potential steps for improvement is to assign a value to each step of the process, and determine thereby the benefit per cost for that step. A valid process requires all the steps, so it is somewhat arbitrary as to the value that might be assigned a given step. A similar problem arises in a project: which steps in the project provide the value of the project? The totality of the project is what delivers the benefit, and no one step by itself provides the project value. Similarly, in a factory, only the entire output of the system determines the value. This output is likely constrained by a number of factors. In a continuous factory setting, one constraint will dominate, and finding this constraint (a bottleneck) and removing it dramatically improves the factory's throughput.

In a project, however, most processes are not repeated. Thus, it would seem that the rule of one dominating constraint does not apply. Yet, in a manner, it does apply. The throughput of a project can be thought of as the benefits of the project minus the development costs. In a delay dominated project, development time is in fact the project constraint, and reduction of overall project time lowers overall cost; and in a development dominated project reduction of resource usage lowers overall cost. Furthermore, while no one step determines (but all tasks on critical path contribute to) the project delivery time, the steps do often resist attempts to shorten their duration differently. Indeed, the joint action of the factors of step sizes (reflected in the task Ks) and task compressibility (α values) will result in the steps requiring different efforts (costs) to yield a similar schedule compression.

Consider the example in Table 2 of a three-step project, with delay constant = $1K per day, staff cost of P = $.5K per day, and the following estimates.

The parallel and serial analysis has shown that if all three steps have the same compressibility $\alpha = 4/3$, ($K_1 = 3^4$, $K_2 = 2 \times 4^4$, $K_3 = 4 \times 2^4$) that the optimal resourcing is given by constant expenditure of 3D per unit of time (6 resources per step) over the entire project. As the plan exists, the project will take 99 days. But with optimal resourcing, the project length is estimated to be roughly 7.04 + 28.08 + 5.90 days = 41.02 days, or 41 days. In this case, the first step has for an increase in total work of 15 workdays (7 × 6 – 27 × 1) delivered a benefit of an approximately 20 days (27 – 7) shorter project; for step 2, an increase in total work of 40 workdays delivered a benefit of a 38-day shorter project; for step 3, an increase in total work of 4 workdays gives a shorter duration of 2 days. Clearly step 1 delivers the best bang for increased cost while step 2, because of its overall size, is also important for project manager to focus upon.

But what if the compressibilities are different? Say ($\alpha_1 = 1$, $\alpha_2 = 4/3$, $\alpha_3 = 2$). In the dramatic case of the first step where $\alpha_1 = 1$, where there is no compression penalty, and the optimal process is to make this task short in duration (near the limit of compression).

Step	Days	Resources
1	27	1
2	64	2
3	8	4

Table 2. Three-Step Project

Indeed, the optimal rate of resource usage for different compressiblities, is given by

$$Q_i = \frac{D}{P_i(\alpha_i - 1)} .$$

For the third step, when $\alpha_3 = 2$, then the optimal resource level is two resources.

Is there something practical in this for the average project manager? Yes. The project manager should consider all the tasks on the critical path of his or her planned project ... and ask the question, "What would it cost to speed each task up on this critical path by 1%, 10%, 20%, 50%?" If any of these compressions yield delay cost savings to the project that are greater than costs to achieve them, then they are possibilities to be pursued. Similarly decompressions may in some cases be appropriate. In any given project plan, one task will usually stand out as having the greatest net benefit from a marginal compression. This task can be thought of as the current "project bottleneck." This task will be the task with the greatest marginal return from using more resources.

It is worth noting that, in theory, the project manager is not limited to using the staff-time relation as given by equation (1). Indeed, with careful estimates of the effort required for various time frames on given steps, an S(T) approximation curve could be built and the optimal resource for the particular task can be determined by brute computation. It has been the author's experience, however, that most teams have a hard time estimating the impact of additional staff on the steps duration, so that the equation (1) approximation is a significant improvement over the naive approximation that work is compressible without penalty. It is conceivable that for particular types of tasks a table of appropriate αs could be built up over time. Until such time that considerable data and experience has evolved in this sort of analysis, use of different αs is, in the author's opinion, a deluded attempt to obtain precision far beyond what equations really give, which is at best only a ballpark estimate of the impacts of varying project staffing levels.

Since, generally, tasks will have similar compression ratios, the rate at which resource dollars are being consumed in each step is an approximate measure of the optimality of the assigned rate of task effort. Tasks with low costs/unit time will stand out as being suboptimal. For example, waiting a significant time for a small inexpensive task to be completed by a vendor or supplier often reveals that effective expenditure of cost per unit of time is very low. The analysis here suggests that the project manager should be willing to offer a bonus up to 3(D) for each day that a vendor or supplier can deliver earlier. Similarly, some tasks may have a cost above 3(D) per day and it may be more cost effective to negotiate a lower price if possible or accepting a slower delivery.

The various tasks of the project can be thought of as devices that resist the time compression. "Springs" are physical devices that resist spacial compression, so by analogy we can think of the project critical path as being composed of springs of different gages and different amounts of stretch. As we attempt to compress the overall project, some of the tasks are easier to compress because they are currently planned at a lower level of resources (task is currently less compressed) and others are easier to compress because the nature of tasks is such that the tasks resist less (costs less) for a given amount of compression.

To Meet or Not to Meet a Date

There are often obligations or expectations to complete a project by a given date. Typically, this type of project has hard date delay cost curves. Meeting these dates may require significant resources and sometimes it is wise to plan to miss the delivery date. Let T_d be the time to target date. Consider a delay cost curve specified by two parameters, D_m, a fixed penalty cost for not making the date, and a linear growing cost for each time unit over the date given by $D(T - T_d)$. It is assumed that there is no delay cost provided that one makes the specified date. In this situation, the total project cost function looks like

$$C(t) = S(T)(P) \text{ for } T < T_d$$
$$C(t) = S(T)(P) + D_m + D(T - T_d) \text{ for } T > T_d .$$

where T_d is the time to hard date. The cost to make the date C_m is given by

$$C_m = P\frac{K}{T^{1/3}} .$$

If one fails to the make the date then there exists another optimal delivery cost C_n given by:

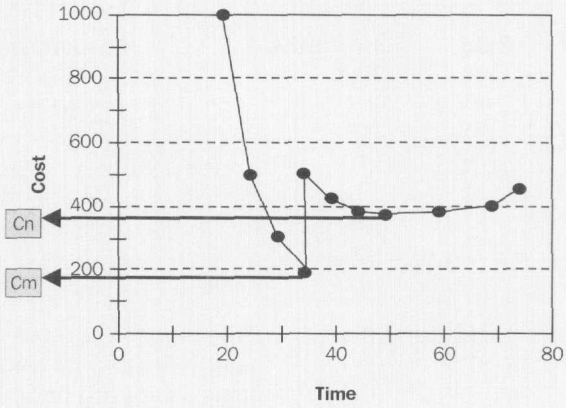

Figure 2. Total Cost Curve With Hard Date Cm

$$C_n = P\left(\frac{K}{T_o^{1/3}}\right) + D_m + D\,(T - T_d) \qquad (3)$$

where

$$T_o = T_d + \left[\frac{K(P)}{3(D)}\right]^{-3/4} \times \left(1 - \frac{T_d}{T_e}\right). \qquad (4)$$

T_o can be solved by iteration in equation (4) and then C_n can then be calculated from equation (3). The difference in cost between not making and making the date, Y, is given by $Y = C_n - C_m$. (See Figure 2.)

The project decision is simple in the case when $Y < 0$ as not making the date is cheaper than making the date. However, when $Y > 0$ the tradeoff is trickier. If you pull off resources from other projects to work on this project, you incur a "hidden cost" in the small delay impact on all the projects from which you have stolen resources. Furthermore, a rushed project typically has high risks and may fail. So, unless the delay impact is dramatic, management needs to be cautious in reallocation of resources.

Mathematics can provide some helpful information to management in this difficult decision. Defining $W = S_m - S_n$ where S_m is the staffing required to make the project date, and S_n is the staffing corresponding to C_n. Let D_a = average delay cost for all active projects in the department. Then the cost of delay of all the department projects, C_a, can be approximated by

$$C_a = T_d \times \frac{W}{Q} \times D_a$$

If $C_a > Y$, then the department should still plan to miss the required date. If $C_a < Y$, then there is still a significant risk that the project may overrun the initial development schedule (especially if making the date is aggressive) and end up missing the date anyway. So, if the date is to be met, then $C_a < Y$.

Manfred's distribution (Schroeder, 1991), outlined in the next section, gives the expectation that a project will overrun by a given amount and can allow us to calculate a new target date T_t that needs to be met so that, at a given confidence level, the date will be met. With the new target date T_t the above calculation can be repeated to see if the meeting the date is still economically wise.

Manfred's Distribution

IT projects have a history of overrunning significantly the original estimates. Is there a point at which a project should be killed? Manfred Schroeder (1991) suggests on theoretical grounds that the probability of a project completing follows a law of the form

$$p(T) = b\,\frac{T_i}{\beta+2}\left(\frac{T}{T_i}\right)^{\beta+1} \textit{for } T < T_i$$

$$p(T) = \left(a\,\frac{\beta}{T_i}\right)\left(\frac{T_i}{T}\right)^{\beta+1} \textit{for } T > T_i$$

where $\beta > 1$, a is the fraction of projects that end up higher than initial estimate, b is the fraction of projects that end up lower than initial estimate T_i of the project's length. Now at a time $T > T_i$ the expected project time T_e is given by

$$T_e = \frac{\int_T^{\infty} p(T)TdT}{\int_T^{\infty} p(T)dt} = \frac{\beta}{(\beta-1)}\,T.$$

Defining Manfred's constant $M = \beta/(\beta - 1)$ then expectation cost to complete a project at time T is, according to Manfred's distribution, $(M-1)\,T\,(P)(S)$. Manfred suggests that the longer a project has failed to complete, the longer its actual expectation time is likely to be. Two special times stand out. At first, the economic picture of the project improves because of the increasing sunk cost in the project. However, following Manfred's thinking, the expected remaining project length continues to grow while the delay in delivery continues to reduce the project's expected benefit. T_R, time of review, is given when expected net benefit of the project drops below the original expected benefit. T_R is thus given by expected remaining benefits minus expected remaining development costs equal the original benefit. The time at which the project should be killed is given when remaining benefits minus expected remaining cost is zero.

Summary

Optimal project resourcing depends upon minimizing the sum of the project development costs and the project delay costs. The key variable in this approach is the project time frame T. The delay cost curve can in most cases be approximated by a linear curve C = D(T). The delay constant D given by D = B / L, where B is the benefit of the project (usually net present value of project) over the period L during which if the project is not implemented the benefits vanish.

The staffing curve of required staff for a given time frame can be approximated by the relation that $S(T^{4/3}) = K$. Together with a loaded labor rate per staff, P, these two laws can be used to obtain the total cost curve. The minimal point can be found at staffing level given by S = 3(D)/ P. While resource allocation is commonly done on a project size basis, a key result of this model is that optimal resource allocation is driven more by delay costs. Thus, a project with twice the delay cost of another project warrants twice the resourcing level even if it is only 25% as large in terms of the overall project effort.

Projects are often subject to overruns on estimates. Manfred's distribution has been found to be useful in analyzing and predicting impacts of project overruns in situations where critical dates are important. The use of such a distribution is outlined for a question of whether a project with a critical date should be met.

It is suggested that a stream of projects can be optimally resourced approximately by resourcing proportionally to the optimal staffing for each project. The total size of potential project stream can be estimated by considering the desirability of the incoming projects (desirability is net present value/net present cost) and the average risk of failure. It is suggested that projects as a function of desirability follow a law something like $N(d) = N / d^2$. Since all projects have a risk of failure, there is an effective cutoff below which projects should not be done. Thus, the total size of the viable incoming project stream can be determined. With this information, one can then compute an estimate of the optimal size in terms of project numbers for a department.

Conclusion

The application of operations research methods to project resourcing provides a basis to answer the question: "How much staff is enough?" This question is a vexing one for most IT departments. An IT department can always add another project to the list with the main visible effect being only a slight delay in the delivery of the many projects currently in development. Thus, an IT shop will always appear to be working at maximum capacity but still be able to take on more work. Furthermore, experience has shown that increasing the workforce of an IT department has little effect on the "development" backlog. Indeed, the development backlog is a poor measure of the need for IT resources for it fails to evaluate the financial desirability and urgency of backlog projects.

With the widespread availability of desktop symbolic and numerical mathematical packages, the calculations suggested here, and generalizations thereof, are easily within reach of all project managers. The readers may be wondering which numerical/symbolic software packages they should purchase for doing the types of calculations suggested in this paper. Personally, the author uses mainly Maple and Mathcad but other packages such as Mathematica, Matlab, Quass, Derive, Scientific Workplace, Excel, and Quattro Pro could also be used successfully. Given the widespread availability of spreadsheets, many readers would want to start with them. Packages such as Mathcad, which allow one to work in standard mathematical notation (via GUI interface) rather than worrying about naming cells in a spreadsheet or entering complex mathematical expressions in the text form, are excellent intermediate investments. The high-powered and higher-priced packages such as Maple, Mathematica, and Matlab are ideal for those that would want to explore the more complex numerical and symbolic formulations. These high-end packages also have superior symbolic/numerical manipulation capabilities, graphic capabilities and output printing capabilities, and come highly recommended for those willing to spend the time to learn them. Particularly useful is the graphics capability of the high-end packages.

The methods outlined here are by no means exhaustive, complete, or thoroughly field verified. Considerable opportunity for research exists. The author hopes that this paper will spawn such research studies.

References

Boehm, B.W. (1981). *Software engineering economics.* Toronto: Prentice Hall.

Conte, S., Dunsmjore, H., & Shen, V. (1986). *Software engineering metrics and models. Effect of team size on productivity* (pp. 258–267). Don Mills: The Bejamin/Cumming.

Jardine, A. (1973). *Topics in operational research: Maintenance, replacement and reliability.* Toronto: Pitman Publishing.

Obdel-Hamid. (1993). *Software management—The dynamics of software project staffing: A system dynamics based simulation approach* (4th ed., pp. 290–300). Washington, DC: IEEE.

Pressman, R.S. (1992). *Software engineering: A practitioner's approach* (3rd ed.). New York: McGraw Hill Inc.

Schroeder, M. (1991). *Fractals, chaos, power laws.* New York: Freeman.