



29th Annual **INCOSE**
international symposium

Orlando, FL, USA
July 20 - 25, 2019

Managing Systems Engineering Projects with Uncertain Requirements

Robert F. Bordley, Professor and Program Director

University of Michigan, Ann Arbor

734-647-4325

rbordley@umich.edu

Jeffrey M. Keisler, Professor

University of Massachusetts, Boston

Boston, MA 02125

617-287-7738

Jeff.keisler@umich.edu

Copyright 2019 by R. Bordley and J. Keisler

Permission granted to INCOSE to publish and use

Abstract

Prior to being alerted that a requirement has changed, project management typically recommends that an engineer treat that requirement as fixed. But proper risk management emphasizes that all uncertainties should be explicitly recognized prior to making any decisions. Hence the engineer should explicitly recognize the possibility that their requirements may change. As a result, systems engineering indexes each requirement by its stability, which reflects its probability of changing. Building on this insight, this paper presents a new technique for considering the uncertainty associated with any requirement in design decisions.

Key Words: Systems engineering, Unstable requirements, risk management, Uncertainty modelling

Introduction

Requirements management is critical in systems engineering. Since requirements drive considerable work in the design process, unexpected changes in requirements are costly. Requirements can change because of unforeseen changes in stakeholder needs (Ward and Chapman, 2003; Huemann, Turner and Keegan, 2007). These changes reflect uncertainty about the capabilities a stakeholder will require in using the deliverable in the intended environment.

In projects, including systems engineering projects, the project manager is required to treat requirements as fixed. Meanwhile a change control process reviews potential changes to requirements, evaluates their cost, and then determines whether to adjust the requirement (Calhoun, Deckro, Moore, Chris and Hove, 2002). Once the decision is made to change the requirement, the change is communicated to all affected managers as quickly as possible. At this point, the focus of the process is on minimizing the disruption caused by change.

Requirement change can lead to costly rework. Furthermore uncertain performance in some activities may affect the needed performance across other activities. But some requirement change

reflects the stakeholder not fully understanding and communicating what they need. To reduce this source of potential change, agile approaches (Maylor, Vidgen and Carver, 2008) restructure the work flow to allow for more frequent iterations with the customer. Since the stakeholder requirements may also change because of changes in their environment, this may not completely eliminate all requirement change. In this case, active monitoring and corrective action (Hu, Cui, Demeulemeester and Bie, 2016) can reduce the time required to detect and react to the need for change as well as speedy generation of options (Creemers, De Reyck and Leus, 2015).

Nonetheless these approaches do require costly and time-consuming changes in conventional project management (Schwaber, 2004). But since even agile approaches cannot eliminate all requirement change, the project manager will still have some residual uncertainty about whether a requirement may change. Recognizing the inevitability of requirement uncertainty, best practices in systems engineering project management classify requirements based on whether they are likely to change (volatility.)

Fortunately, there already exist well-established tools for making decisions optimally once uncertainty has been recognized (Ding and Zhu, 2015; Elmaghraby, 2003; Morgan and Henrion, 1992; Chapman and Ward, 1997; Clemen, 1996; Khodakarami, Fenton and Neil, 2007; Pich, Loch and DeMeyer, 2002). Many of these approaches, e.g., decision trees, influence diagrams, are already recognized in both the Systems Engineering Body of Knowledge (SEBOK, BKCASE 2019) and the Project Management Body of Knowledge (PMBOK, Project Management Institute, 2013). These tools have been especially successful in strategic decision making as well as in evaluating the uncertain outcomes of different design decisions.

To apply these well-established decision tools to requirement change, this paper shows how requirement uncertainty can be formalized as a kind of outcome uncertainty – which is something existing methods can handle. Hence the project manager, instead of treating a requirement as fixed

until it changes, will be able to make decisions which recognize the uncertainty in their requirements. The value of recognizing uncertainty in making decisions (versus ignoring it) has already been quantified as the expected value of including uncertainty (Morgan and Henrion, 1982). This paper will show that simply recognizing requirement uncertainty (versus ignoring it) is sometimes almost as valuable as recognizing and eliminating uncertainty using agile methods. Because agile methods are often costly and time-consuming, this can make the proposed approach a cost-effective alternative to agile.

The next section introduces set-based requirements for measuring the evolving degree of uncertainty associated with a requirement over time. The subsequent section then introduces the proposed technique for embedding this uncertainty within the existing structure of project management. Later sections then quantify the benefits of this approach using simulation.

Set-Based Requirements

This paper addresses requirement uncertainty by translating requirement uncertainty into an outcome uncertainty which can then be addressed by existing decision analysis tools. As simulations in this paper show, this approach can sometimes capture most of the benefits of agile approaches. To measure this uncertainty, we follow best practices in uncertainty management by identifying bounds on the amount of capability the stakeholder might require.

Thus at the start of the project, standard systems engineering practice, in addition to setting hard requirements on external interfaces and regulations, also sets ‘best-case’ bounds on other attributes. We define a bound as ‘best-case’ if the probability of the actual requirement being less lenient is negligible. For example, a commercial airplane must have at least enough range to fly from one airport to the nearest neighboring airport. An automobile must have sufficient range to drive a customer to a nearby gas station. These ‘best-case’ bounds can be formulated as hard

requirements, i.e., any performance below the bound is unacceptable. But as the project proceeds, information is often collected which enables this ‘best-case’ capability bound to be made more demanding.

We also define a bound as ‘worst-case’ if the probability of the actual requirement being more stringent than the bound is negligible. There are many cases in which the system engineer also often identifies ‘worst-case’ bounds on the capability a stakeholder needs. For example, there was a maximum weight requirement set for the Direct TV satellite based on how much weight the platform at Cape Canaveral could launch and still get the satellite into the desired geocentric orbit. But the systems engineer (Winter, 2015) realized that the amount of weight that could be launched from Cape Canaveral would change dramatically if there were a hurricane. To avoid the risk of failure due to weather conditions, the systems engineer gave his design team a more aggressive target with a more demanding weight limit than specified by the stated requirement. The wisdom of using such safety margins in adjusting the requirement was confirmed when there was a hurricane on the day that the Direct TV satellite was launched. So by setting a ‘worst-case’ bound on the capability that might be required, the systems engineer enabled the successful launch of Direct TV. New information might lead the systems engineer to update this ‘worst-case’ bound to be less demanding.

Together the best-case and worst-case bounds define an interval which bounds the range of uncertainty in the maximum capability the stakeholder will require in the intended application. Since there is little probability of the capability the stakeholder needs being outside this interval, future information about the uncertain capability needed will only shrink this interval (or set) by making the best-case requirement more aggressive or the ‘worst-case’ requirement less aggressive. In this case, the widely used stability and volatility indices measure the likelihood of the interval shrinking. Any design work which was rejected given the ‘best-case’ bound will still be rejected

when that bound changes. Any design work which was unnecessary given the ‘worst-case’ bound will still remain unnecessary if that bound changes. Thus new information (or new requirements) will not lead to a revisiting of previously rejected designs. It may however lead to a rejection of designs that used to be under consideration. These requirements could be considered semi-hard because they never become more lenient but may become more demanding.

The concept of describing requirements by a shrinking interval is consistent with the concept of set-based design. Set-based design defines a set of feasible designs and, as new information about requirements emerges, eliminates designs which are no longer feasible given this new information. For this reason, we refer to this concept of using bounds to describe requirements as set-based requirements.

Among the designs which satisfy the set-based requirement, the designs which perform closer to the best-case bound are less likely to fall below the lower bound when new information is added. Hence the more a design’s capability outperforms the ‘best-case’ bound, the more robust the design to the risk of a future tightening of that bound. As a result, the designer should favor designs which (other things being equal) score better on the interval because of their robustness even though they currently satisfy the set-based requirement.

Given complete information, the lower and upper bounds will not necessarily converge to a single point. Requirements are typically defined in terms of a threshold (which is the minimum capability acceptable) and an objective (beyond which further improvements in capability have negligible value.) Incremental improvements beyond the threshold but below the objective are valued. The lower bounds on the uncertain requirements will converge, given perfect information, to the threshold while the upper bound will converge to the objective. Hence the notion of set-based or semi-hard requirements extends the concept of threshold and objective by recognizing existing uncertainty about the capability needed.

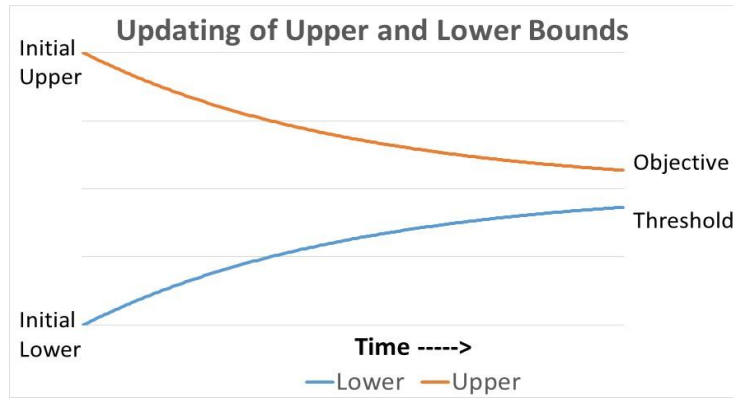


Figure 1: Convergence of Initial Bounds to Threshold and Objective

Before introducing the approach, it will be helpful to define a common ‘orientation’ for all the requirements. For example, a higher fuel efficiency score is always better. In this case, the optimistic requirement is the lower limit on the amount of fuel efficiency which the stakeholder might require (while the pessimistic requirement is the upper limit). But a smaller score is always better on the time required to accelerate a vehicle from zero to sixty miles. In this other case, the optimistic bound is an upper limit on time and the pessimistic bound is the lower limit. To avoid ambiguity, this paper will assume that all attributes are defined so that less is always better. Thus the optimistic requirement always specifies an upper limit on what level of capability on some dimension is acceptable while the pessimistic requirement specifies a lower limit beyond which further improvement is of negligible value. Thus for the project deadline (Kolish and Padman,2000), any project finishing later than the optimistic bound is unacceptably late while there is no benefit to the project finishing any earlier than the pessimistic bound.

When a project is at risk of not satisfying a requirement (i.e., of not meeting the threshold on that requirement), the manager must allocate additional resources to different activities so as to improve capability on that requirement. Note that a manager who recognizes requirement uncer-

tainty might still take action on improving the project's capability on some requirement dimension even if the average capability satisfies the requirement. Hence the manager who recognizes requirement uncertainty will make different decisions than the manager who is unaware of requirement uncertainty.

Proposed Solution

Motivation

The original Systems Vee description of systems engineering begins with the specification of stakeholder requirements and terminates with the delivery of a system satisfying those requirements to the stakeholder. But since this only reflects some of the steps in systems engineering (as specified in the SEBOK), more comprehensive design models like the winged Vee (and the Systems M variant on the winged Vee) have been developed. These models consider the 'meta-system' which precedes the specification of stakeholder requirements as well as the subsequent meta-system in which the stakeholder uses that deliverable to address that stakeholder's problem. Thus the systems engineer responsible for designing the Direct TV satellite also considers the transportation and launching of the satellite from a launch vehicle at Cape Canaveral. Likewise the systems engineer responsible for designing a vehicle also considers the additional vehicle-prep work done at the dealership. In some cases, this post-delivery work can be substantial.

Consistent with these approaches, suppose the stakeholder, upon receiving the deliverables from the manager's project, starts a 'complementary' activity designed to use those deliverables to finish some longer term project. When the manager's project and this complementary activity are finished, the stakeholder's objectives have been achieved. Suppose the longer-term project's required performance on different dimensions is described by the vector r_0 . Suppose this complementary activity 'uses up' some of the capability developed along all the requirement dimensions. Since

lower values on the requirement dimensions are preferred to higher values, this complementary activity will increase the score.

Then in order to meet the requirements, r_0 , the project needs to produce enough capability to offset the capability lost in the complementary activity. Suppose the stakeholder knows that the amount of capability lost by the complementary activity is some vector t . Then the stakeholder could give the manager's project the more aggressive requirement vector $r = r_0 - t$ which would then allow the stakeholder to meet their requirement vector r_0 .

But suppose the complementary activity's impact on each performance dimension is uncertain. If t is uncertain, then the requirements, $r = r_0 - t$, on the project will be uncertain. So the stakeholder cannot give the systems engineer requirements for the original project until they learn the outcome of the uncertainties in the complementary project. The systems engineer could address this requirement uncertainty by redefining their actual project so that it only meets its requirements when the longer term stakeholder project meets its requirements.

This enlarged project will have the same requirements r_0 as the stakeholder's long-term project. The project network describing this enlarged project consists of the network for the original project plus an additional activity, corresponding to the complementary activity. This complementary activity will only start after all the activities in the original project finish. But while the manager may be able to add resources to affect the work done on the activities in the original project, the manager will not be able to affect the work done by the complementary activity.

Since a project typically has multiple requirements, r_0 will be a vector of long-term requirements, r will be the vector of capability delivered by the project and t will represent the vector of capability lost in the complementary activity. The probability that the project satisfies all its long-term requirements will be the probability that $r + t \leq r_0$. To quantify the uncertainty in t and thus in $r_0 - t$, the stakeholder might assess an optimistic upper bound t_O on how lax t_O might be, a

pessimistic lower bound t_P and a most likely value t_L . (If the project manager treats requirement change as analogous to scope creep and believes that requirements will only get tighter — and never get more lax — than the project manager might estimate $t_O = 0$ and treat the initially stated requirement as the optimistic bound for the project’s eventual requirements.) Once t_O , t_L and t_P are specified, the manager can then use standard methods to model the probability of the redefined project satisfying the requirement vector r_0 (which is just the probability that $r+t \leq r_0$.) The uncertainty in the requirements for the original project is completely determined by the uncertainty in the complementary activity. Since the manager is now responsible for a redefined project including the complementary activity, the manager will make decisions that appropriately recognize the uncertainty in the requirements for the original project.

In this example, the stakeholders implemented one complementary project to achieve their objectives. In the absence of information on a complementary project, we can have the stakeholder express their uncertainty in the requirements by specifying estimates $r_0 - t_O$, $r_0 - t_L$ and $r_0 - t_P$ for the optimistic, most likely, and pessimistic level of requirements. Then t_O, t_L, t_P define optimistic, most likely and pessimistic estimates of how much the requirements will tighten. As before, making decisions to maximize the probability of satisfying the vector of eventual requirements is equivalent to maximizing the probability of the project’s characteristics, r , being no greater than $r_0 - t$. To model this with the classical project management structure, i.e., an activity network, the manager re-defines the original project to have requirements vector r_0 but adds an additional fictitious complementary activity with uncertain performance vector t which starts when all other activities finish.

Fictitious activities have been used, for other purposes, in other project management contexts (Vanhoucke (2013), pg. 244; Schwindt (2005), pg.8). The classical project management solution to the original project supplemented with this added activity will be the appropriate solution to

the original project where the stakeholder's completion time was uncertain. Figure 1 presents a screenshot both of the project network with uncertain requirements and a redefined project network with a fictitious complementary activity.

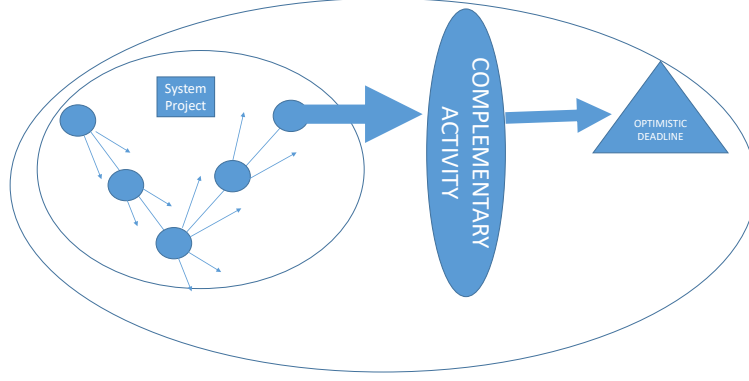


Figure 2: Project Network with a Fictitious Complementary Activity

Analytical Comparison

If r is the vector describing a project's performance on each requirement dimension, then the probability the project meets all its requirements is the probability that $r \leq r_0 - t$ or the probability that $r + t - r_0 \leq 0$. Define μ to be the mean of $r + t - r_0$. Define σ^2 to be the variance-covariance of r , the outcome of the project. Define v_0 to be the vector of variances of $t - r_0$ which reflect uncertainty in the requirement. If the requirement uncertainty is independent of the deliverable uncertainty, then $\sigma^2 + v_0$ is the variance of the difference between the deliverable and the requirement.

Suppose we focus on the case in which there is a single requirement and let $s = \sqrt{\sigma^2 + v_0}$. Then the probability this requirement is satisfied, if all uncertainties are Gaussian, is the probability that

$$r + t - r_0 = 0 \implies \frac{r + t - r_0 - \mu}{s} \leq -\frac{\mu}{s}$$

If $z = \frac{\mu}{s}$ and if Φ is the cumulative normal distribution, then the probability is $\Phi(z)$.

Suppose that costs are rescaled so that the cost of a unit improvement in the performance of

the project on this requirement is one unit. Suppose that the cost of reducing the variance in the project's deliverables from σ^2 to $\frac{\sigma^2}{n}$ is cn . Then if we improve the path's mean performance by C units and reduce the variance by dividing it by a factor of n , the z-score becomes $\frac{\mu+C}{\sqrt{v_0+\frac{\sigma^2}{n}}}$ and the overall cost is

$$\Phi\left(\frac{\mu+C}{\sqrt{v_0+\frac{\sigma^2}{n}}}\right) - C - cn$$

Differentiating by C gives

$$\frac{\phi(z)}{s} - 1 = 0 \implies \phi(z) = s \tag{1}$$

Differentiating by n gives

$$\frac{1}{2} \frac{\phi(z)}{s^3} \frac{\sigma^2}{n^2} - c = 0 \tag{2}$$

Substituting $\phi(z) = s$ into this equation gives

$$\frac{1}{2} \frac{\sigma^2}{n^2 s^2} = c \implies \frac{1}{2cn^2} = \frac{1}{n} + \frac{v_0}{\sigma^2}$$

This quadratic equation can be solved for n (and thus s). Note that as $\frac{\sigma^2}{v_0}$ decreases, the righthand-side of the equation increases and n decreases. Thus the more the deadline uncertainty exceeds the project uncertainty, the smaller the value of reducing project uncertainty. We can then solve for C using

$$\phi(z) = s \implies \mu + C = s\phi^{-1}(s)$$

Simulation Comparison of Alternate Approaches

The relationship between the project's performance on each requirement dimension and the contributions made by different activities is complex and varies from requirement to requirement. As a result, the impact of adding resources to different activities will vary from requirement to requirement. However this relationship is particularly simple when the requirement dimension is project completion time. Thus, considering how uncertainty may be incorporated in this case is both useful

for its own sake and as a demonstration of the potential practicality of the proposed approach. Here, the project completion time is the maximum completion time of each path through the project and the completion time of each path is the sum of the completion times of each activity along the path. The well-known PERT algorithm solves this problem by defining the project's completion time as the completion time of the path with the largest expected completion time (the critical path.) This approach involves an approximation which is known to consistently underestimate a projects actual completion time except in those cases when the critical path always finishes later (or at the same time) as every other path in the project, Our approach, applied to PERT, simply involves the addition of a fictitious activity along the critical path.

There is a trade-off between the ease of implementation and benefit of using PERT, with this adjustment for requirement uncertainty, compared to that for more modelling intensive approaches. To explore the usefulness of the proposed approach extended to other situations, we consider a slightly more complicated activity network and simulate in Microsoft Excel the performance of the PERT heuristic (modified to include requirement uncertainty) against an optimization approach under different assumptions about uncertainty.

Specifically, we consider the activity network shown in Figure 4, which is a tree with one level of branches of varying lengths beyond the root. This is still relatively simple, but different enough from a linear project to test whether the results are an artifact of the critical path assumption. The sequence of activities 1, 2 and 3 forms the first branch, activities 4 and 5 in sequence form the second, and activity 6 is the third. All three branches precede activity 7. We select initial parameters so that

- the mean deadline is greater than the mean project completion time;
- it is plausible that some activity not on the PERT critical path will delay completion;

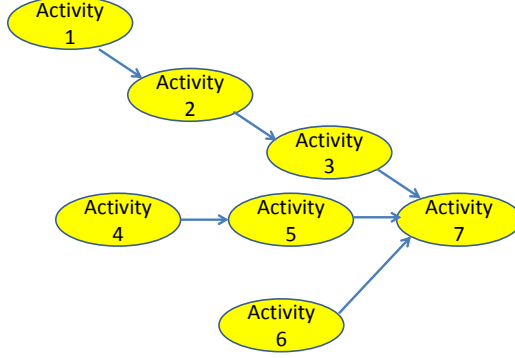


Figure 3: The Project Network

- there is a non-trivial chance that the project will not be on time.

We initialize random variables Y_i for the activity durations with means m'_i of 6, 5, 4, 8, 6, 13 and 8 respectively, each with the same initial standard deviation $S_i = S$ (which is varied across four scenarios). The project's deadline T is uncertain with mean 25 and standard deviation 3. The effect of crashing activity i is to reduce its mean to $m''_i \geq 0$, and to increase its standard deviation to $S_i = S + (m'_i - m''_i)$. A crashing strategy, which leads to reductions in the mean time for each activity, creates a new set of random variables Y'' and T which are Gaussian and independent. To calculate the probability of on-time completion, we first compute the cumulative density function for $\chi = \max(Y_1 + Y_2 + Y_3, Y_4 + Y_5, Y_6)$, which is defined by

$$\Pr.\{\chi \leq w\} = \Pr.\{Y_1 + Y_2 + Y_3 \leq w\} \Pr.\{Y_4 + Y_5 \leq w\} \Pr.\{Y_6 \leq w\},$$

noting that the sum of activity times on each branch is Gaussian with mean equal to sum of the activity means and variance equal to the sum of activity variances on that branch. The project is on time if $\chi + Y_7 \leq T$, i.e., if $\chi < T - Y_7$ where $T - Y_7$ is the difference between two Gaussian random variables and hence is Gaussian. (This is equivalent to adding the dummy activity for deadline uncertainty.) From the joint distribution of χ and $T - Y_7$, we obtain the density $\Phi(T - Y_7 - \chi)$

and thus $\Pr.\{T - Y_7 - \chi \geq 0\}$, i.e. the probability the project finishes by its deadline. We then use a generalized reduced gradient algorithm to solve for the crashing strategy which maximizes this probability for given parameters of Y'' and T , subject to the constraint that each $m_i'' \geq 0$. We do this once for the scenario in which initial activity times are known so that $S = 0$, once where activity times have low uncertainty ($S = 0.5$), once with medium uncertainty ($S = 1.0$), and once with high uncertainty ($S = 2.0$).

For each scenario, we identify the strategy which optimizes probability of success under four different assumptions about uncertainty:

1. PERT: The deadline is assumed to be fixed with standard deviation 0, and there is no chance that activities 4, 5 or 6 can enter the critical path (i.e., they are assumed to have means of 0 and standard deviations of 0);
2. Optimization: The deadline is fixed but all activities might be appear on the critical path and therefore all might be crashed, i.e., the correct parameters are entered;
3. Enhanced PERT: The deadline is assumed to have its correct standard deviation of 3, but as before activities off the initially identified critical path (1, 2, 3, 7) are ignored;
4. Enhanced optimization: all activities are crash-able and deadline uncertainty is included.

Note that strategies (1), (2) and (3) are optimal under conditions which somehow distort the project, while strategy (4) is a gold-standard which is optimal under the correct assumptions. Our measure of performance is the probability that each of the strategies thus identified leads to on-time completion under the assumptions of (4). For additional comparison, the table includes the probability of success if there no crashing. The results in Table 1, as expected, show that no crashing is worst and enhanced optimization is best in each scenario. For $S \leq 1.0$, the performance of enhanced PERT

	No Crashing	PERT	Optimization	Enhanced PERT	Enhanced Optimization
$S = 0.0$	74.21 %	74.21 %	74.21 %	91.24%	94.49%
$S = 0.5$	72.1 %	85.47%	77.82%	88.29%	91.07%
$S = 1.0$	66.18%	82.81%	81.29%	84.65%	87.91%
$S = 2.0$	53.83%	77.25%	81.49%	77.29%	81.50%

Table 1: Outcomes of Simulation Experiment

is close to the performance of enhanced optimization with both enhanced PERT and enhanced optimization substantially outperforming both PERT and optimization. When $S = 2.0$, the deadline uncertainty is small relative to the project duration uncertainty and optimization, with or without enhancement, outperforms both PERT and enhanced PERT.

Also note that for $S = 0.5$ and 1, PERT counter-intuitively performs better than optimization because

- the risk aversion of the project manager in a favourable situation is artificially strengthened when deadline uncertainty is ignored which leads to too little crashing on any of the three branches
- PERT ignores the possibility that any activities will be replaced on the critical path, and so overweights the impact of crashing those activities. This offsets the bias the risk-aversion created by ignoring deadline uncertainty

In practical terms, when the scale of the project is not so large or the structure of the project is not so complex as to justify large modelling expense, project managers can still obtain considerable benefit utilizing internal staff to conduct modified-PERT analysis in problems involving substantial deadline uncertainty.

Summary

Requirement uncertainty is a major problem in systems engineering. This paper proposed a general approach toward solving the problem by introducing tentative thresholds and objectives that are gradually tightened as more information surfaces about those requirements. This addresses the concern of engineers being forced to ignore the uncertainty in their requirements by treating them as fixed (and certain) until they change. At the same time, it recognizes that formalizing hard requirements is essential to the design process.

We consider the case where this requirement uncertainty is due to stakeholder uncertainty about the level of capability needed in meeting the stakeholder project for which the deliverables were created. We then consider the general case by defining the fixed requirements to be the optimistic or most lenient level of requirements. In projects where requirement change only lead to more stringent requirements, the optimistic requirements will be the initially stated requirements and the deviation between optimistic and pessimistic requirements reflects potential scope creep. A fictitious customer expectations management activity is defined whose success is measured by the degree to which requirement change is minimized. This activity is added after the existing activities in the project network. Since this is now part of the systems engineering project, managing the project using the standard methods of project management will now account for requirement uncertainty.

To demonstrate the benefits of this approach, we then considered its application to managing uncertainty in a single requirement, the project deadline. This paper then demonstrated the potential benefit of this innovation in making project crashing decisions. The technique of introducing an artificial activity was especially useful for address project deadlines. In the context of PERT (and related network descriptions of projects), this way of recognizing deadline uncertainty requires only minimal changes in existing methodologies. It is, in theory, possible to introduce a similar artificial expectations management activities for other requirements. In similarly simple contexts,

e.g., analogous to PERT, where weight (or cost) of the product is sum of weight (or cost) of its parts, or cost extending to other features, we might have a fictitious part with random weight (or cost), or even some combination of fictitious parts and activities. In more complex cases, sophisticated algorithms are needed even without requirement uncertainty. As seen in the multi-path activity network example, there may be substantial additional benefit to incorporating requirement uncertainty to those algorithms.

References

- [1] Brucker, P., A. Drexl, R. Mohring, K. Neumann, and E. Pesche (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*. 112. 3-41
- [2] Calhoun, K., R. Deckro, J. Moore, J. Chrissis and J. Van Hove. (2002). Planning and re-planning in project and production scheduling. *Omega*. 30, 155-170.
- [3] Chapman, C and S. Ward. (1997). *Project Risk Management: Processes, Techniques and Insights*. Wiley, New York
- [4] Clemen, R. (1996). *Making Hard Decisions*. 2nd ed. Duxbury Press, Belmont, MA.
- [5] Creemers, S., B. De Reyck and R. Leus (2015). Project planning with alternative technologies in uncertain environments. *European Journal of Operational Research*. 242(2), 465-476.
- [6] Ding, C. and Y. Zhu.(2015). Two empirical uncertain Models for project scheduling problem. *Journal of the Operational Research Society*. 66, 1471-1480.
- [7] Elmaghraby, S. (2005). On the fallacy of averages in project risk management. *European Journal of Operational Research*. 165, 307-313.

- [8] Estevez-Fernandez, A. (2012). A game theoretical approach to sharing penalties and rewards in projects. *European Journal of Operational Research*. 216(3), 647-657.
- [9] Golenko-Ginzburg, D. (1989). PERT assumptions revisited. *Omega* 17, 393-396.
- [10] Herroelen, W. and R. Leus (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*. 165. 289-306.
- [11] Herroelen, W. and R. Leus (2003). The construction of stable project baseline schedules. *European Journal of Operational Research*. 156. 550-565.
- [12] Hu, X., N. Cui, E. Demeulemeester, and L. Bie (2016). Incorporation of activity sensitivity measures into buffer management to manage project schedule risk. *European Journal of Operational Research*, 249(2), 717-727.
- [13] Huemann, M., R. Turner and A. Keegan (2007). Managing human resources in the project-oriented company. In Morris, P and J. Pinto. *The Wiley Guide to Project Organization and Project Management Competencies*. John Wiley & Sons: Hoboken, New Jersey
- [14] Kamburkowsi, J. (1996). New validations of PERT times. *Omega*, 25(3) 323-328.
- [15] Keefer, D. and W. Verdini (1993). Better estimation of PERT activity time parameters. *Management Science*. 39, 9, 1086-1091.
- [16] Kerzner, H. (2009) *Project Management: A Systems Approach to Planning, Scheduling and Controlling*. John Wiley & Sons: Hoboken, New Jersey
- [17] Khodakarami, V., N. Fenton and M. Neil. (2007). Project scheduling: Improved approach to incorporate uncertainty using Bayesian networks. *Project Management Journal*. 38(2), 39-49.

- [18] Klastorin, Ted (2003). *Project Management: Tools and Trade-offs* (3rd ed.). John Wiley & Sons: Hoboken, New Jersey.
- [19] Maylor, H., R. Vidgen and S. Carver (2008). Managerial complexity in project-based operations: A grounded model and its implications for practice. *Project Management Journal*. 39(S1), S15-S26.
- [20] Meredith, J., S. Mantel (1995). *Project Management: A Managerial Approach*. John Wiley and Sons, Hoboken: New Jersey
- [21] Miller, R. (1963). *Schedule, Cost and Profit Control with PERT*. McGraw-Hill, New York.
- [22] Morgan, M. and M. Henrion (1992). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge University Press:Cambridge, U.K.
- [23] Neumann, J. and J. Zimmerman (1999). Resource leveling for projects with schedule-dependent windows. *European Journal of Operational Research*. 117(3), 591-606.
- [24] Prez, J. G., M.Martn, C. Garca and M. Granero (2016). Project management under uncertainty beyond beta: The generalized bicubic distribution. *Operations Research Perspectives* 3, 67-76.
- [25] Pich, M. ,C.Loch and A. DeMeyer. (2002). On uncertainty, ambiguity, and complexity in project management. *Management Science*, 1008-1023, 48,8
- [26] Project Management Institute (2013). *A Guide to the Project Management Body of Knowledge* (5th ed.). Project Management Institute.
- [27] Schwaber, K. (2004). *Agile Project Management with SCRUM*. Microsoft Press, Redmond, Washington
- [28] Schwindt, C. (2005). *Resource Allocation in Project Management*. Springer: New York

- [29] Sculli, D. (1989). A historical note on PERT times. *Omega* 17(2), 195-196.
- [30] Vanhoucke, M. (2013). *Project Management with Dynamic Scheduling*. Springer, New York.
- [31] Ward, S. and C. Chapman.(2003). Transforming project risk management into project uncertainty management. *International Journal of Project Management*, 21, 97105
- [32] Winter, D. (2015). Personal Communcation.
- [33] Zafra-Cabeza, A., M. Ridao and E. Camacho. (2008). Using a risk-based approach to project scheduling: A case illustration from semiconductor manufacturing. *European Journal of Operational Research*. 190, 708-723.

Biography

Dr. Robert F. Bordley: Professor and Program Director, Systems Engineering and Design, University of Michigan, Ann Arbor, formerly with Booz-Allen-Hamilton, General Motors and the National Science Foundation, 90 academic publications.

Dr. Jeff M. Keisler : Professor of Management Information Systems, University of Massachusetts, Boston, formerly with Strategic Decisions Group, Argonne National Labs, General Motors, 69 academic publications.